

RiverCure Portal: Collaborative GeoPortal for Curatorship of Digital Resources in the Water Management Domain

Marta Gonçalves Barcia Gonzalez

Thesis to obtain the Master of Science Degree in

Engineering and Industrial Management

Supervisors: Prof. Alberto Manuel Rodrigues da Silva
Prof. Bruno Emanuel da Graça Martins

Examination Committee

Chairperson: Prof. Miguel Simões Torres Preto
Supervisor: Prof. Alberto Manuel Rodrigues da Silva
Members of the committee: Prof. Nuno Alexandre Correia Martins Cavaco
Prof. Rui Miguel Lage Ferreira

October 2020

Declaration

I declare that this document is an original work of my own authorship and that it fulfills all the requirements of the Code of Conduct and Good Practices of the Universidade de Lisboa.

Abstract

Nowadays, there is a huge quantity of information available regarding water resources, although with different structures, formats, languages, vocabulary, which lead to a lack of consistency, accuracy and increased the time spent modifying such information. On the other hand, flood events are relatively frequent. The previously stated problems underline the importance of a well-designed water resources information system. Without solving them, the knowledge that several sources have, cannot be linked or shared properly. This inhibits the development of accurate monitoring and alert systems.

To explore these problems, the Design Science Research Methodology was applied. The first two steps focused on the identification of the problem, motivation and objectives of this work. In the third step, the system was designed and developed. Afterwards, the system was demonstrated using a case study, the flood event in Águeda in 2016.

The RiverCure portal is a web-based system, that will allow its users to simulate flood events and to access information regarding water resources captured by sensors from distinct sources. This system intends to improve flood forecasting and the management of water resources and supporting decision-makers to make better decisions.

This research intends to preliminary discuss and define the requirements specification of the RiverCure portal using a rigorous language, namely the Requirements Specification Language. To achieve this goal, it was necessary to specify the systems' requirements of the main inputs of the portal, namely the Portuguese monitoring and surveillance system and the simulation software since these systems are important features of the portal.

Keywords: RiverCure, Water Resources, Information Management, Requirements Engineering

Resumo

Atualmente, existe uma grande quantidade de informações disponíveis sobre recursos hídricos, muito embora cada uma possua uma estrutura, formato, e vocabulário diferente, o que leva a uma falta de consistência, e a um aumento no tempo gasto na modificação dos dados. Além disso, as cheias são relativamente frequentes. Os problemas anteriores ressaltam a importância de um sistema de informações sobre recursos hídricos bem definido. Sem resolver estes problemas, o conhecimento das várias fontes, não pode ser interligado ou compartilhado adequadamente. Impedindo o desenvolvimento de sistemas precisos de monitorização e alerta.

Para explorar estes problemas utilizou-se o *Design Science Research Methodology*. Nas duas primeiras fases foi identificado o problema, a motivação e os objetivos deste trabalho. Na terceira etapa, o sistema foi projetado e desenvolvido. Depois, este foi demonstrado utilizando o caso de estudo da cheia de Águeda em 2016.

O portal RiverCure é uma plataforma *online* que permite aos usuários simular inundações e aceder a informações sobre recursos hídricos adquiridas por sensores de fontes distintas. Com esta ferramenta, é possível melhorar os sistemas de previsão de inundações e a gestão dos recursos hídricos, e ajudar os decisores a tomar decisões fundamentadas.

Esta dissertação pretende definir preliminarmente a especificação dos requisitos do portal RiverCure utilizando uma linguagem rigorosa, o *Requirements Specification Language*. Para atingir este objetivo foi necessário especificar os requisitos dos principais sistemas insumos do portal, nomeadamente do sistema de monitorização e vigilância de recursos hídricos portugueses e do software de simulação, uma vez que são peças importantes no portal.

Palavras-Chave: RiverCure, Recursos Hídricos, Gestão de Informação, Engenharia de Requisitos

Acknowledgements

Through this journey, I had the opportunity to learn a lot about the topic, but also about myself and the people that surround me.

Thank you so much, Professor Alberto Silva and Professor Bruno Martins, for all the guidance, feedback, patience, and availability during this project.

RiverCure team, thank you for the opportunity to work in this team, for the work put into it and for all the dedication.

Thank you, Dra. Manuela Saramago, for all the help provided, for all the mail responses and all the meetings.

Without the motivation, support and help of my dad, mom, grandmother and boyfriend this project would have been much more difficult. For all the conversations, the patient and support a huge thank you.

I would also like to thank my friends, Ana Sofia Angélico, Morgana, Joana Assunção, Joana Oliveira, Catarina, Madalena and Inês for all the moments, for always being there to pull me up and for showing me how lucky I am to have all of you in my life.

I also want to thank all the members of BEST Almada for all the talks, all the help and the countless moments that helped me relax from all the stress.

Finally, thank you to the Portuguese FCT for co-funding the project RiverCure: Curating and assimilating crowdsourced and authoritative data to reduce uncertainty in river flow modelling (GRANT_NUMBER: PTDC/CTA-OHR/29360/2017).

To all a big thank you.

Table of Contents

- Abstract** iii
- Resumo**..... v
- Acknowledgements**..... vi
- Table of Contents** vii
- List of Figures**..... ix
- List of Tables**..... xiii
- List of Acronyms** xv
- 1. Introduction** 1
 - 1.1. Context 1
 - 1.2. Motivation and Problem Identification 2
 - 1.3. Dissertation Objectives..... 3
 - 1.4. Research Methodology..... 4
 - 1.5. Contributions 6
 - 1.6. Document Structure..... 6
- 2. Background**..... 7
 - 2.1. Water Resources Management..... 7
 - 2.1.1. Basic Concepts..... 7
 - 2.1.2. Sistema Nacional de Informação de Recursos Hídricos (SNIRH) 8
 - 2.1.3. Sistema de Vigilância e Alerta de Recursos Hídricos (SVARH) 10
 - 2.2. WaterML 12
 - 2.3. Geographic Information System (GIS) 13
 - 2.4. HiSTAV 15
 - 2.5. Requirements Engineering 18
 - 2.5.1. Basic Concepts..... 19
 - 2.5.2. UML 19
 - 2.5.3. ITLingo and RSL..... 21
- 3. Bibliographic Review** 24
 - 3.1. Water Resources 24
 - 3.2. Requirements Engineering 28
- 4. Related Systems** 31
 - 4.1. SNIRH and SVARH 31

4.2.	HiSTAV	44
5.	RiverCure Portal	50
5.1.	Glossary.....	51
5.2.	Stakeholders.....	52
5.3.	Actors.....	52
5.4.	Data Entities	54
5.5.	Requirements	62
6.	Conclusions and Future Work.....	79
6.1.	Conclusions	79
6.2.	Future Work	80
Annexes		86
	SNIRH	86
	SVARH	88
	HiSTAV	91
	RiverCure.....	94
	Meetings	106

List of Figures

Figure 1: Data's transformation scheme.	3
Figure 2: Methodology's scheme of this work.	4
Figure 3: Location of Águeda's bridge and photo of a flood in Águeda [19].	5
Figure 4: a) Hydrometric station [28]; b) Meteorological station [24].	8
Figure 5: Schema of SNIRH's processes (inspired from [24]).	9
Figure 6: Schema of SVARH's processes (adapted from [24]).	10
Figure 7: a) Rios desktop [24]; b) Rios editor [11].	11
Figure 8: Alarm levels example (extracted from [26]).	12
Figure 9: Example of layers in QGIS.	14
Figure 10: Example of a mesh.	15
Figure 11: HiSTAV data flow schema.	16
Figure 12: a) Example of an inlet definition in QGIS; b) Boundary conditions definition.	17
Figure 13: a) Pre-processor running; b) Physics, numerics and time files.	17
Figure 14: a) HiSTAV running; b) Example of a simulation output file.	18
Figure 15: Example of a simulation result (bed elevation) in ParaView.	18
Figure 16: Overview of the RSLingo's process (adapted from [52]).	22
Figure 17: UML diagram of SVARH DataEntities: Basin package.	35
Figure 18: UML diagram of SVARH DataEntities: Station package.	36
Figure 19: Example of a station in a dam and a Hydrometric station in Rios [24].	37
Figure 20: UML diagram of SVARH DataEntities: Alarm package.	37
Figure 21: Station definition in Rios editor.	38
Figure 22: UML diagram of SNIRH's DataEntities: Location package.	38
Figure 23: UML diagram of SNIRH's DataEntities: User package.	39
Figure 24: User preferences in Rios.	40
Figure 25: UML diagram of SNIRH UseCases: Data package.	42
Figure 26: Example of a graphic in Rios [69].	43
Figure 27: UML diagram of the SVARH UseCases: Manage package.	44
Figure 28: UML diagram of the pre-processor outputs DataEntities.	47
Figure 29: UML diagram of the HiSTAV UseCases.	49

Figure 30: Data transformation schema.	50
Figure 31: RiverCure portal user roles.	53
Figure 32: UML diagram of the RCP DataEntities: Feature package.	55
Figure 33: UML diagram of the RCP DataEntities: Sensor simple package.	55
Figure 34: UML diagram of the RCP DataEntities: Fixed in-situ package.	56
Figure 35: UML diagram of the RCP DataEntities: Alarm package.	56
Figure 36: UML diagram of the RCP DataEntities: Photo package.	57
Figure 37: UML diagram of the RCP DataEntities: Event package.	58
Figure 38: UML diagram of the RCP DataEntities: Simulation main package.	58
Figure 39: UML diagram of the RCP DataEntities: Map package.	59
Figure 40: UML diagram of the RCP DataEntities: User permissions package.	61
Figure 41: Home page mock-up of the RCP.	63
Figure 42: a) Data menu mock-up of the RCP; b) Interactive map mock-up of the RCP.	63
Figure 43: UML diagram of the RCP UseCases: Map package.	64
Figure 44: UML diagram of the RCP UseCases: Monitoring points package.	64
Figure 45: a) Monitoring points by basin (User) mock-up; b) By monitoring points (User) mock-up. ...	64
Figure 46: UML diagram of the RCP UseCases: Hydrometric and Udometric data package.	65
Figure 47: a) Hydrometric monitoring points (User) mock-up; b) Hydrometric monitoring points (Hydrometric technician) mock-up.	65
Figure 48: Radar observations mock-up of the RCP.	66
Figure 49: a) Photos menu mock-up of the RCP; b) Upload photos mock-up of the RCP.	66
Figure 50: a) Photo gallery mock-up of the RCP; b) Photos gallery (Crowdsourcing technician) mock-up of the RCP.	67
Figure 51: a) Simulations menu mock-up; b) Simulations menu (Numeric technician) mock-up of the RCP.	68
Figure 52: a) New simulation scenario (Input definition) mock-up; b) New simulator scenario (Progress indicator) mock-up of the RCP.	68
Figure 53: UML diagram of the RCP UseCases: Simulations package.	69
Figure 54: Simulations records mock-up of the RCP.	69
Figure 55: Simulations input files (Manage files) mock-up of the RCP.	70
Figure 56: UML diagram of the RCP UseCases: Technician simulation package.	71
Figure 57: a) New simulation (Pre-processor) mock-up; b) New simulation (model configuration) mock-up of the RCP.	72

Figure 58: a) Events records mock-up of the RCP; b) Events menu (Udometric technician) mock-up.	72
Figure 59: Settings (Profile information) mock-up of the RCP.	73
Figure 60: Settings (Account) mock-up of the RCP.	75
Figure 61: Settings (Alarms) mock-up of the RCP.	75
Figure 62: UML diagram of the RCP UseCases: Manager package.	76
Figure 63: Settings (Manage information) mock-up of the RCP.	77
Figure 64: UML diagram of all SNIRH DataEntities.	86
Figure 65: UML diagram of SNIRH UseCases: Manage package.	87
Figure 66: UML diagram of the SNIRH UseCases: Restricted user and Public viewer package.	88
Figure 67: UML diagram of SVARH DataEntities.	90
Figure 68: UML diagram of SVARH UseCases: Data package.	91
Figure 69: UML diagram of the pre-processor inputs.	92
Figure 70: UML diagram of the pre-processor UseCases.	93
Figure 71: UML diagram of the RCP DataEntities: Spatial package.	95
Figure 72: UML diagram of the RCP DataEntities: Sensor package.	95
Figure 73: UML diagram of the RCP DataEntities: Simulation input package.	95
Figure 74: UML diagram of the RCP DataEntities: About package.	96
Figure 75: UML diagram of the RCP DataEntities: User main package.	96
Figure 76: UML diagram of the RCP DataEntities: User settings package.	96
Figure 77: UML diagram of the RCP UseCases: User photos package.	100
Figure 78: UML diagram of the RCP UseCases: Hydrometric and Udometric settings package.	100
Figure 79: UML diagram of the RCP UseCases: Photos package.	101
Figure 80: UML diagram of the RCP UseCases: Alarm package.	101
Figure 81: UML diagram of the RCP UseCases: Settings package.	101
Figure 82: UML diagram of the RCP UseCases: Technician settings package.	101
Figure 83: UML diagram of the RCP UseCases: Events package.	102
Figure 84: UML diagram of the RCP UseCases: SuperAdmin package.	102
Figure 85: Interactive map (photos) mock-up of the RCP.	104
Figure 86: Monitoring points (By district) mock-up of the RCP.	104
Figure 87: Monitoring points (Graphic) mock-up of the RCP.	104
Figure 88: Photos gallery (Crowdsourcing technician) mock-up of the RCP.	104

Figure 89: Photos gallery mock-up of the RCP. 104

Figure 90: Simulations input files (Upload files) mock-up of the RCP. 104

Figure 91: New simulation (Results) mock-up of the RCP. 105

Figure 92: Settings (Profile information) mock-up of the RCP..... 105

Figure 93: Settings (Account) mock-up of RCP..... 105

Figure 94: Settings (Account) mock-up of the RCP..... 105

Figure 95: Settings (Manage users) mock-up of the RCP..... 105

List of Tables

Table 1: Examples of UML elements.	20
Table 2: RSL constructs' classification [5].	22
Table 3: Conversion of some system's terms [7].	51
Table 4: Permissions of RCP user roles.	94

List of Acronyms

APA	Agência Portuguesa do Ambiente
CFL	Courant-Friedrichs-Lewy
CHT	Confederação Hidrológica do Tejo
CRS	Coordinated Reference System
CPPE	Companhia Portuguesa de Produção de Eletricidade
CSIRO	Commonwealth Scientific and Industrial Research Organization's
CUAHSI	Consortium of Universities for the Advancement of Hydrologic Science, Inc
DSRM	Design Science Research Methodology
DSM	Digital Surface Model
DTM	Digital Terrain Model
EDP	Energias de Portugal
FIS	Flood Information System
FCT	Fundação Ciência e Tecnologia
GIS	Geographic Information System
GML	Geography Markup Language
HEC-DSS	Hydrologic Engineering Center Data Storage System
HEC-HMS	Hydrologic Engineering Center - Hydrologic Modeling System
HIS	Hydrologic Information System
HiSTAV	High performance version of Strong Transients in Alluvial Valleys
IFIS	Iowa Flood Information System
INAG	Instituto da Água
INSPIRE	Infrastructure for Spatial Information in the European Community
IPMA	Instituto Português do Mar e da Atmosfera
INESC-ID	Instituto de Engenharia de Sistemas e Computadores - Investigação e Desenvolvimento
ISO	International Organization for Standardization
ISRBC	International Sava River Basin Commission
IST	Instituto Superior Técnico

NWIS	National Water Information System
O&M	Observation & Measurements
ODM	Observations Data Model
OGC	Open Geospatial Consortium
OMG	Object Management Group
PGRH	Planos de Gestão de Região Hidrográfica
PSL	Project Specification Language
RE	Requirements Engineering
RCP	RiverCure portal
RML	Requirement Modeling Language
RSL	Requirements Specification Language
RSL-IL	Requirements Specification Language - Intermediate Language
RSL-PL	Requirements Specification Language - Pattern Language
SNIRH	Sistema Nacional de Informação de Recursos Hídricos
SRS	Systems Requirements Specifications
SVARH	Sistema de Vigilância e Alerta de Recursos Hídricos
SysML	Systems Modeling Language
TSL	Test Specification Language
UML	Unified Modelling Language
USGS	United States Geological Survey
WAMDaM	Water Management Data Model
WOF	WaterOneFlow
WEAP	Water Evaluation and Planning System
WFS	Web Feature Services
XML	Extensible Markup Language

1. Introduction

1.1. Context

Flood events have been increasing in Portugal. For instance, between 2011 and 2018 the Agência Portuguesa do Ambiente (APA) reported 306 flood events and the number one underlying cause identified was strong precipitation [1]. The fact that several basins, namely those from the Tagus and Douro rivers, cross Spain's border hinders the assessment of the flood risk and the characterization of these risks. Another factor that must be taken into consideration is climate change, which again according to APA can have a significant impact since the forecast for Portugal is that, in general, the annual mean precipitation will decrease, but the number of short-period precipitation events is expected to increase and with more intensity compared with the present situation [1].

Nowadays, with the evolution of technology, the size of information collected, discovered and available has been increasing rapidly [2]. New problems have thus emerged with it, such as data not properly documented, difficult access to data, due to ownership or cost, and inconsistent data between different sources, making the use of it more difficult [3]. To overcome these problems, data should be rigorously defined. In the current research, different languages were analysed, so that terms and concepts could be accurately defined and connected. Taking that into account, the Unified Modelling Language (UML) [4], the Requirements Specification Language (RSL) language [5], and consequently the ITLingo project were researched, so that the requirements specification of the systems linked to the project could be written in the best possible way. Overall, RSL includes a process to transform requirements written in a natural language into formal structured requirements [5, 6].

This research should also analyse the Water Markup Language standard (WaterML), which is a standard for publishing time series of hydrological observation data, which uses the Extensible Markup Language (XML) format to publish data over the internet via web services, with the intent of sharing and documenting this type of information [8, 9].

To further understand this topic, a case study was used to demonstrate the main results: the flood event on the 12th of February 2016 in Águeda, Aveiro, Portugal. This event it reached historical water levels according to the data obtained from the City Hall and the Sistema Nacional de Informação de Recursos Hídricos (SNIRH), which is owned by APA.

The RiverCure project has the objective improving the forecasting systems focusing on the impact of floods, to manage water resources and habitat protection. In order to achieve that it is necessary to assimilate data from official agencies (e.g. APA) and acquire new data that is shared on the internet, such as images of such events shared in Instagram or Flickr, among other social media platforms, to help tune the forecasting capabilities of the mathematical models. High performance STAV (HiSTAV) is the simulation tool applied, to check the consequences inland of a flood [4]. The three main outputs expected from the RiverCure project are (1) an IT platform that articulates data and model forecasts;

(2) a new-generation river model; and (3) a curated database that combines authoritative and crowdsourced data. The project is funded through Fundação para a Ciência e Tecnologia (FCT). There are three main partners involved in the project, Information and Decision Support Systems' laboratory of the Instituto de Engenharia de Sistemas e Computadores - Investigação e Desenvolvimento (INESC-ID), Association of Instituto Superior Técnico for Research and Development (IST-ID) and APA.

This research project was developed within the RiverCure project, more specifically in the development of the IT platform and it counts with the support and guidance of Professor Alberto Rodrigues da Silva and Professor Bruno Emanuel da Graça Martins, which are the supervisors of this Master Thesis in Engineering and Industrial Management at Instituto Superior Técnico, Universidade de Lisboa.

1.2. Motivation and Problem Identification

Information systems have a major impact in organizations. They can reduce costs, help support innovation strategies and improve communication between the people involved [9].

There are several data sets regarding water resources available online, such as the Hydrologic Information System¹ (HIS) from the Consortium of Universities for the Advancement of Hydrologic Science, Inc (CUAHSI) and the SNIRH² from APA. However, since the files have different structures, term definitions, file formats, units, and file types, the usage of that data becomes quite difficult [3]. The most probable reason for this sudden quantity of data growth is the increasing awareness regarding climate change, leading to an increase in the number of monitoring programs (e.g. SVARH) and sensors so that more data can be collected to be used as inputs in models [3]. Several international guidelines are trying to standardize this information, in terms of geographic properties (e.g. International Organization for Standardization (ISO) 19109) and observation data (e.g. Observations Data Model (ODM), WaterML, Open Geospatial Consortium (OGC) and ISO Observations and Measurements (O&M)).

Currently, a higher effort has been seen in writing more carefully system requirements. Nevertheless, the amount of errors is still significant, probably due to the flexibility of the natural language, which can lead to ambiguity, lack of accuracy, and shortage of relevant information [10].

By using rigorous specification techniques, it is possible to facilitate the communication among the stakeholders, develop better products, minimize risks, produce and visualize relevant models, keep a record of all decisions, and control and guide the direction of the project.

There are different sources of information relevant to the project. This means that there are also many stakeholders involved, namely APA, IST-ID, INESC-ID, IPMA, and FCT. Consequently, it can make the

¹ <https://data.cuahsi.org/>

² <https://snirh.apambiente.pt/>

process of data collection harder, due to miscommunications and coordination problems between the stakeholders, and also ownership problems [11].

The RiverCure project intends to define and develop a digital platform with data, both from authorities (e.g. APA and Instituto Português do Mar e da Atmosfera (IPMA)) and crowdsourcing. In the end, the project will have produced a more accurate river model, in terms of hydrodynamic and morphodynamic characteristics, and a platform that links with the models, which will contain all the data from different sources [12].

To summarize, the main problems and challenges regarding the subject addressed in this thesis are the difficulty in managing the water resources data, the existence of several stakeholders, and the presence of many guidelines to standardize water-related information. All these issues are some of the reasons why forecasting natural events, such as floods, has been a challenge. The development of the RiverCure portal (RCP) is one possible solution to these problems.

1.3. Dissertation Objectives

The main objective of this thesis is to develop a requirements specification for the RiverCure Portal system, in order to allow an easier system implementation in the future.

Figure 1 shows all the processes that the different types of data go through until being released in the RiverCure portal. Firstly, the data from distinct sources, such as APA, IPMA, and the photos uploaded in social media platforms (e.g. Facebook, Instagram) is collected and analyzed. Once this step is concluded the data is presented in the portal and the simulation process can start, since it has the needed input data. The simulation tool, HiSTAV [4], is executed, producing a simulation that indicates the possibility of a flood event occurring. Lastly, the result from this simulation is published on the RiverCure portal. This means that the system must be able to collect and analyze information, to perform a simulation and to present data in the portal. These processes will be further explained in Chapter 5.

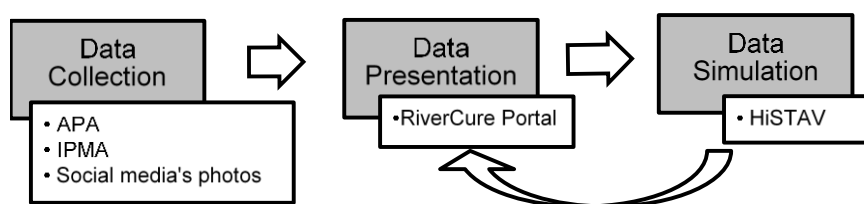


Figure 1: Data's transformation scheme.

Upon understanding all the previous information, it was concluded that answering the following questions would be important:

- Which are the requirements necessary to define the RiverCure Portal?
- How should the specifications be represented?
- Is the RSL language an adequate approach to define structured and rigorous requirements in the water domain?

To sum up, the objectives for this project are:

- Define the as-is model of the SNIRH, SVARH (Sistema de Vigilância e Alerta de Recursos Hídricos) and HiSTAV (High performance version of Strong Transients in Alluvial Valleys) systems;
- Define a requirements specification for the RiverCure portal.

1.4. Research Methodology

Due to the existence of several guidelines, stakeholders, and information systems, it was decided that an appropriate methodology to apply in this research is the Design Science Research Methodology (DSRM). The DSRM is used in cases where new artefacts are created in order to solve real problems and it can be applied to a variety of areas, such as engineering, and information systems [13]. The next figure (Figure 2), presents an overview of the methodology used in this project [13].

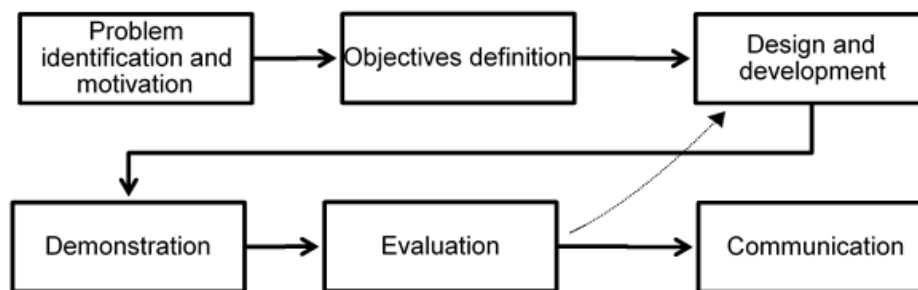


Figure 2: Methodology's scheme of this work.

1.4.1. Problem Identification, Motivation and Objectives Definition

The first step has significant importance since, by defining the problem, the complexity of it can be captured and better outcomes can be achieved. However, it is necessary to research the state of the art and acquire specific knowledge related to the topic, in order to properly define the problem and the objectives. By explaining the motivation of the work, the importance of it, and the impact that it can have, the readers will be more engaged. The definition of the objectives, the second step, should state the goals that the artefact must fulfil.

1.4.2. Design and Development

The third step is divided into two parts. The first is focused on the determination of the structure and requirements of the model, while the second part is the construction itself. To achieve it, data must be collected, so research methods, like interviews and meetings, are employed to acquire the information needed. The use of structured interviews, one-to-one or group, allows collecting data, but also some opinions and experiences related to the problem. I specifically organized, meetings with the RiverCure team, which includes researchers from different areas of expertise (e.g. Information Systems, Geographic Systems, Civil Engineering, Hydraulics and Computer Engineering) and meetings with the responsible for APA's water resources department, Dra. Manuela Saramago. The list of all the project

meetings is presented in the annexes. To define the system requirements specification, I used the UML (with the Sparx Systems Enterprise Architect tool) and RSL (supported by the ITLingo-Studio tool) languages.

1.4.3. Demonstration

The fourth step is the application of the system to solve at least part of the problem. This step involves a case study. The case study allows one to understand the relationship between parts, which may demonstrate possible complex situations. The case study makes it possible to understand better the outcomes. It also allows the use of a variety of sources and types of data. Case studies also were applied in the papers [15, 16] presented in Chapter 3.

The flood event in Águeda, at 12th of February 2016, is the case study that is going to be used since it is well documented and the event reached a historical water level (Figure 3), namely 5,88 m in the Águeda's bridge station [16] (location shown in Figure 3). Águeda is located in the Aveiro district, in Portugal, and it has the main tributary of the Vouga river crossing the city, namely the Águeda river. With its spring in the Caramulo Mountains, situated in the Vouga's river basin and with approximately 35 kilometres of extension, it covers an area of 971,8 km² [17]. This area was flagged by APA as one of the 23 critical areas in Portugal with a flood risk [18]. This case study was applied in all the mock-ups of the RiverCure portal.

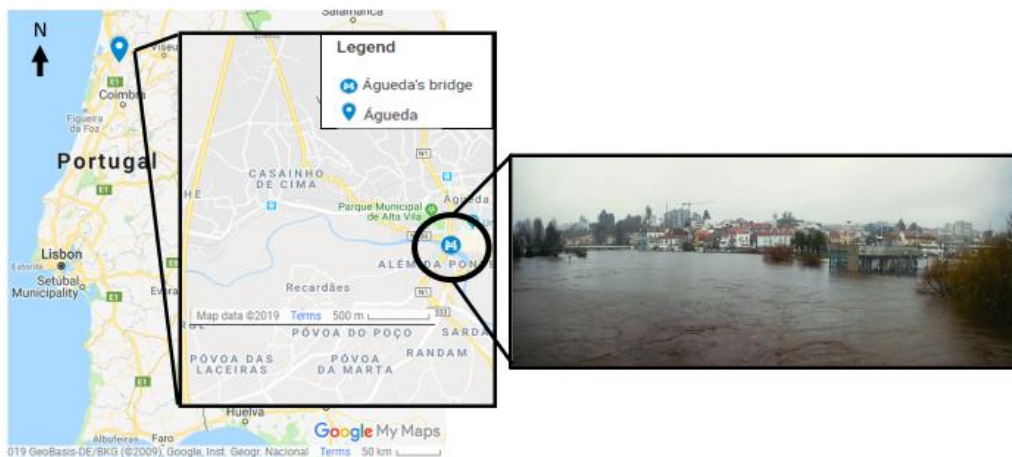


Figure 3: Location of Águeda's bridge³ and photo of a flood in Águeda [19].

1.4.4. Evaluation

After applying the case study, the artefact must be checked to see if it works properly and fulfils the objectives defined. However, if the model does not satisfy the conditions or is not working correctly, the project should return to step three, so that it can be improved and tested again. To evaluate the requirements specification defined in the previous steps, informal feedback from some members of the RiverCure team was collected. Those comments were acquired during team meetings (see list of meetings in the annexes).

³ <https://www.google.com/mymaps>

1.4.5. Communication

At last, a research paper should be written in order to present to the community all the work done regarding the subject. This step is highly important for the development of the subject, by publishing an article stating the results and findings of this project, it will allow to other professionals to take advantage of the work previously done and develop it even more. For instance, by using data that was collected beforehand for other projects, it is possible to save time and reduce costs, since there is no need to use specialized equipment or to treat data. Also, by divulging a paper, awareness regarding the topic is raised, an open science environment is instigated, and communication between the people involved in project is improved [20].

1.5. Contributions

To achieve the objectives defined in this thesis and to answer to all the research questions, the following contributions were made:

- The definition of the models of the SNIRH, SVARH and HiSTAV in both RSL and UML languages, in order to understand how these systems function;
- The specification of the RiverCure portal system was made using UML diagrams and RSL, so that in a first stage the specification would be easily understandable and in a second it would assure that the requirements were written in a rigorous and structured way. Mock-ups of the RiverCure portal were also developed, to demonstrate the specifications of the portal and to receive more feedback, especially from those team members that are domain expert.

1.6. Document Structure

This dissertation is divided into 6 Chapters. Chapter 1 presents the context of the problem, the objectives of this work, and the actual problem. Chapter 2 defines the core concepts, topics and systems for the RiverCure portal. Chapter 3 gives a bibliographic review regarding water resources and requirements engineering. Chapter 4 shows the as-is model of the systems related to the RiverCure portal (SNIRH, SVARH and HiSTAV). Chapter 5 contains the business and the application model of the RiverCure portal system. Chapter 6 presents conclusions regarding the problem and some developments that can be made in the future. At the end of this document, one can find two other sections, the annexes, displaying several additional information regarding the systems and the RiverCure project, and the references used in this document.

2. Background

This section introduces the main terms, definitions and information regarding the related systems and languages used in this work. This Chapter is divided into three sections: Water Resources, Simulation Tools (HiSTAV) and Requirements Engineering (RE). The first section presents all the terms and standards related to hydrology. The second section provides an overview of contents related to simulation and geographic information systems. Lastly, the RE section provides basic concepts of requirements and modelling languages.

2.1. Water Resources Management

This section introduces, concepts related to water management. Afterwards, the Portuguese information system regarding water resources, SNIRH, and the Portuguese flood alert information system, SVARH, are explained.

2.1.1. Basic Concepts

Throughout this work many hydrological terms are used. Each one has a definition which can vary by author and sources. For instance, the definition of a hydrographic network according to the Agência Portuguesa do Ambiente (APA) corresponds to a “set composed by the main river and its connected affluent, this includes lakes. Creating a geographical space that receives all the superficial flow from occurred precipitation” [21]. By comparing it to the WaterML’s definition it can be stated that they are quite similar, for example both include the lakes. However, APA’s definition does not state if it includes or not reservoirs.

In order to be clear and easier for the reader to follow this work, some hydrological terms are defined here:

Hydrology – “Science that deals with the waters above and below the land surfaces of the Earth, their occurrence, circulation and distribution, both in time and space, their biological, chemical and physical properties, their reaction with their environment, including their relation to living beings” [22].

Hydrographic network – “Aggregate of rivers and other permanent or temporary watercourses, and also lakes and reservoirs” [22].

Hydrometric or hydrological network – “Hydrological stations and observing posts situated within a catchment in such a way as to provide the means of studying its hydrological regime” [22].

Stream – “Water, generally flowing in a natural surface channel, or in an open or closed conduit, a jet of water issuing from an orifice, or a body of flowing groundwater” [22].

Watercourse – “Natural or man-made channel through or along which water may flow, including large interstices in the ground, such as cave, cavern or a group of these in karst terrain” [22].

Flood - A temporary area of land covered by water, that normally is not covered. In this case, flood events will only be considered if they had negative effects, such as environmental, cultural or economic damage, being hurtful to the community and injurious to health. In this work, the focus will be mostly on fluvial floods, caused by a huge quantity of rain throughout several days or weeks, which spread to the surrounding areas [1].

Most of the definitions of the terms presented previously are from the WaterML standard because it is the standard that is being studied and that by using it errors caused by inconsistent definitions are avoided.

2.1.2. Sistema Nacional de Informação de Recursos Hídricos (SNIRH)

SNIRH (Sistema Nacional de Informação de Recursos Hídricos) was divulgated on October the 1st, 1995, by Instituto da Água (INAG) as a water resources monitorization system, that acquires hydro-meteorological data from a set of stations, saves it in APA's Structured Query Language (SQL) Server database and releases the information in a portal⁴. The portal gets 600 visits per day, with users including teachers, students, researchers, journalists and public administration personal. The system is divided into three sub-systems: SNIR-Lit, SNIRH-Júnior and Sistema de Vigilância e Alerta de Recursos Hídricos (SVARH) [23].

There are two main types of automatic stations in Portugal: Hydrometric or Meteorological [23]. The Hydrometric stations (Figure 4 a)) record data from rivers, lakes or reservoirs. These data includes the water level, flow, transport and deposit of sediments, temperature and other physical, chemical and biological properties of the water. The Meteorological stations (Figure 4 b)) can obtain meteorological data, such as precipitation, temperature, air humidity, wind velocity and others. They can be divided into two sub-types, Udometric and Climatological. If the station only measures precipitation it is called Udometric or Udographic. If it also measures air and wind properties it is called Climatological.

a)



b)



Figure 4: a) Hydrometric station [28]; b) Meteorological station [24].

In Portugal, there are a total of approximately 931 stations, 311 hydrometric and 620 meteorological [25]. For them to work, the stations must have sensors, a datalogger and a power supply, which

⁴ <https://snirh.apambiente.pt/>

includes a battery and a solar panel. The datalogger can have up to 30 channels each one with a different storage rate and alarm levels. Some stations are also equipped with teletransmission, meaning that they can communicate with SVARH using Global System for Mobile Communications (GSM) or General Packet Radio Service (GPRS) [11].

If the stations have teletransmission, the data is sent directly from the station to APA's central system. Otherwise, the data must be collected during the maintenance of the station by the maintenance company, and uploaded in the maintenance company server afterwards.

Most of the errors that appear in the data collected are the result of poor phone coverage, temporary loss of network connection, fallen scales, blockages in sensors, sensors not gauged, dried sensor bowl, or influences from an external factor (e.g. a water pump). To eliminate the sources of mistakes and to preserve the machines, a company is hired to perform a maintenance service, where all the sensors are tested, the data is compiled, an assessment sheet is filled with observations, and photos might also be taken. The hydrological stations require a visit every three months and the meteorological stations every three months or month to month, depending on the sensors assembled.

Figure 5 illustrates the main steps that must occur for the data from stations to become available to the regional authorities and on the web. The data goes through three stages, namely data acquisition, data processing and data release. The processes required to acquire data from the stations vary depending on the station communication capabilities and their ownership. If the data is provided by the Spanish hydrographic authorities (e.g. Confederação Hidrológica do Tejo (CHT)) or by hydroelectric power plants' companies, known as Companhia Portuguesa de Produção de Eletricidade (CPPE) (e.g. EDP), the data is transferred to the SNIRH's server using a script. For an APA's station without teletransmission, the data will be recorded, and the parameters defined using a software called Geolog. Then, it will be uploaded to the maintenance company server, where APA's technicians will download it from. Upon these steps the data is run in the application INAG SIF, that will generate a file with all the parameters that will be uploaded to the SNIRH's server and published on the web.

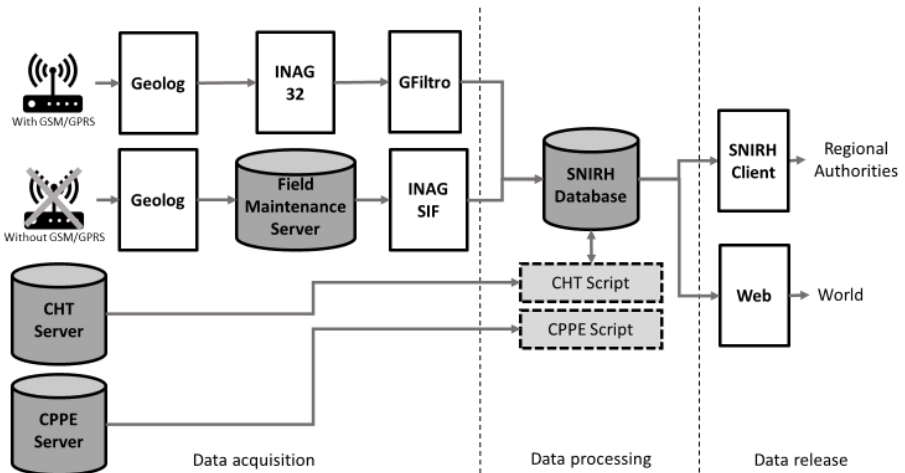


Figure 5: Schema of SNIRH's processes (inspired from [24]).

The process that the data provided by APAs' stations with teletransmission goes through is approached in the next section since it is the main source of data of SVARH.

2.1.3. Sistema de Vigilância e Alerta de Recursos Hídricos (SVARH)

SVARH is a system that allows analysing in real-time the hydric state of the rivers and dams all over Portugal. Meteorological information is also collected to allow a more precise prediction, and for better understanding better the data collected. For the system to work, a network of stations with teletransmission was installed in critical points, where the risk of floods exists.

The only users are APA's technicians, Regional Environmental Departments, Regional Water Authorities, National Fire Department and Civil Protection Service [11].

The SVARH has three core modules, as can be seen in Figure 6 [24]:

- First module: Data acquisition (Acquisition and conversion of data from the stations with teletransmission and from partners);
- Second module: Data processing (Upload of the data to the databases);
- Third module: Data release (Dissemination of the data in real-time using the Rios application).

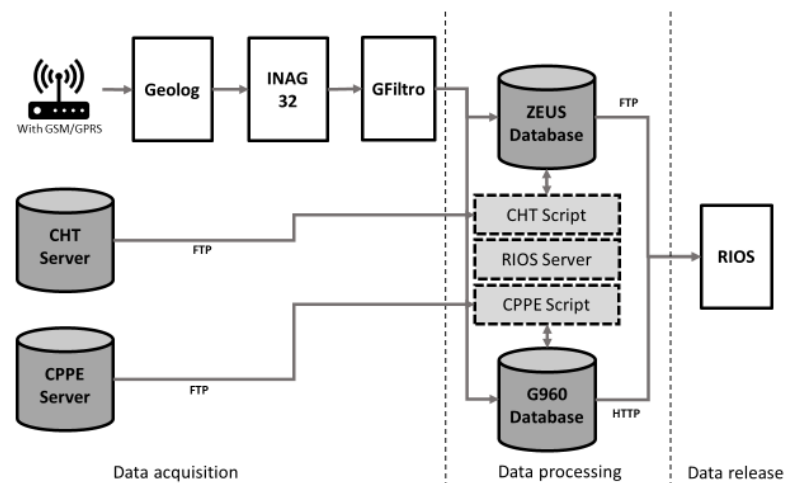


Figure 6: Schema of SVARH's processes (adapted from [24]).

The management of the alarms sent from the automatic stations (e.g. change of interrogation rhythm) and the alteration of the parameters remotely are some of the functions that the Geolog software is responsible for.

The INAG32 software is responsible for converting the data from the Geolog format into the American Standard Code for Information Interchange (ASCII), creating one or more files. This application works in collaboration with the Geolog, allowing the definition of the number of days that the output will contain excluding older data [24].

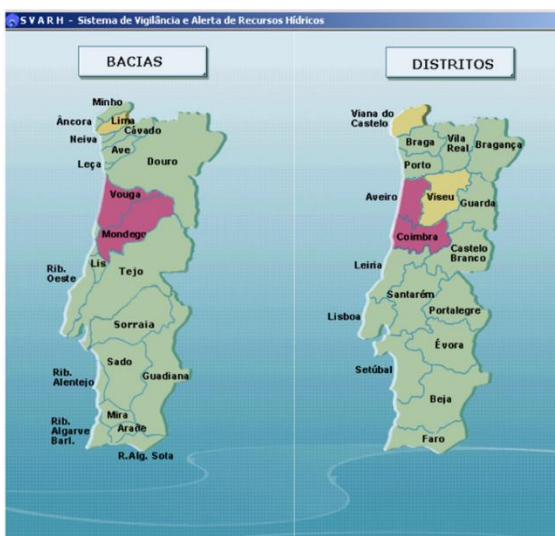
The GFiltro software is used to convert the ASCII files obtained from the INAG32 to files into the SVARH format, and it also uploads them into the database [24].

This system has two servers, namely Zeus and G960. Both can store the data recorded by the automatic stations. The Zeus server uses the File Transfer Protocol (FTP) to transfer the data, while the G960 server uses Hypertext Transfer Protocol (HTTP) (Figure 6). After being transferred, the data can be consulted using the Rios application. Another functionality of the G960 server is the ability to send the alarms from the stations via text message or email [24].

Besides the information recovered from APA's stations, the CPPE (e.g. EDP), and the CHT also send their data in exchange for APA's information (Figure 6) [11]. The data is transferred every hour from both servers using specific scripts that follow the FTP transference protocol [24].

Rios is the application used to present the data in a user-friendly manner, considering that the access to the data is based on web services and SQL servers. Figure 7 a) presents the Rios desktop platform, which can create charts and is able to export to MS-Excel. Besides the Rios desktop, there is also the Rios editor (Figure 7 b)), used to manage users and to edit the screen layouts, which includes designing and defining rivers, basins and stations. When defining a station, the parameters, alarm levels and past events are specified. The Rios SMS application is also an important service that allows the reception of text message (SMS) alarms on the mobile phone for selected incidents [11].

a)



b)

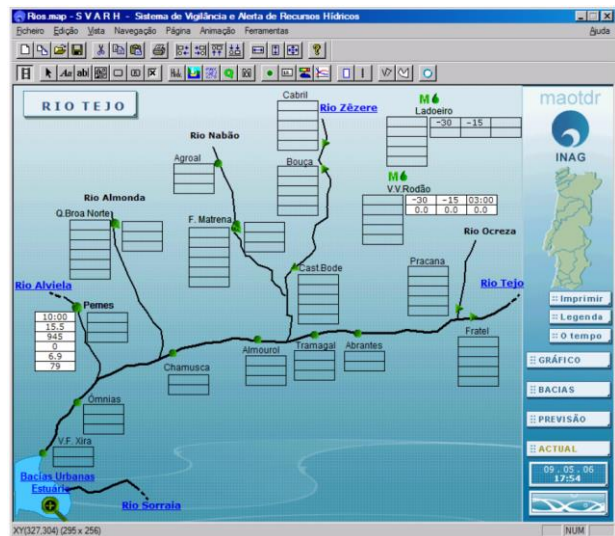


Figure 7: a) Rios desktop [24]; b) Rios editor [11].

When an alarm is triggered, a notification is sent to the control centre automatically, and it also changes the frequency of the data storage and transfer. For the warning to be sent, a specific parameter that is being measured must have reached a previously defined limit. From Figure 8 it is possible to understand that Rios has two levels of alarm (yellow and red), and that the colour of the station symbol, basin and district matches the alarm level. Note that the limits of each level change between the stations, since they are specifically defined for each site. All the records from the area are considered by an APA technician when deciding the alarm limits.

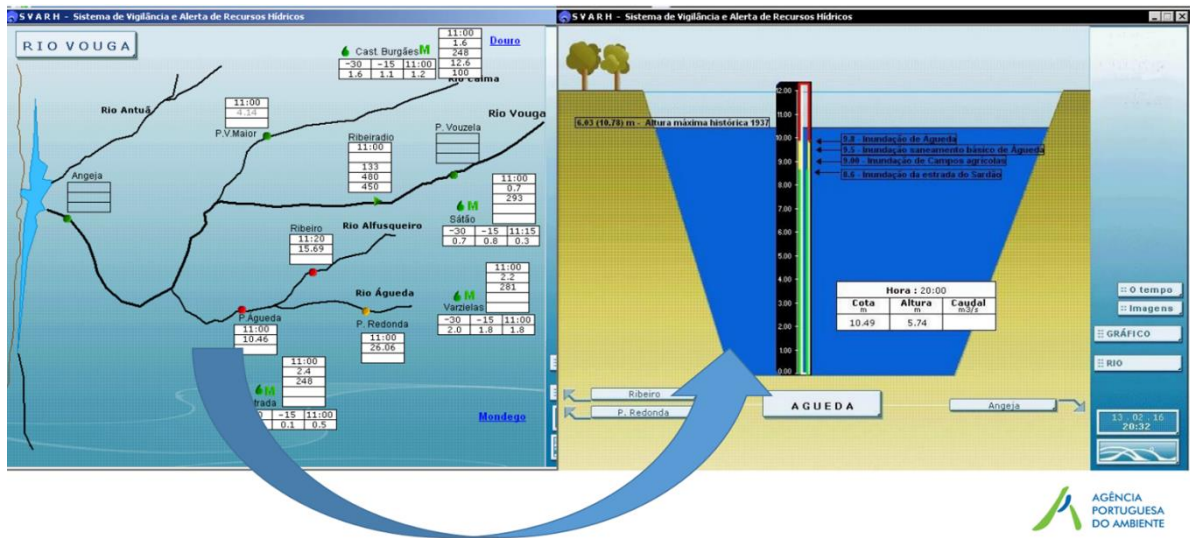


Figure 8: Alarm levels example (extracted from [26]).

An important feature in this system (SVARH) is the ability to run simulations but, unfortunately, only some of the main river basins have hydrologic forecast models incorporated, not allowing the prediction of floods in other locations [12, 26].

2.2. WaterML

WaterML 2.0 is one of the specifications that allows the data from hydrological observations to be coded, implementing a XML schema [27]. The standard is composed of two parts, namely the conceptual UML model regarding the observational data (O&M) and the implementation of the model in XML schema, more specifically in a GML schema. This specification represents not only timeseries data but also spatial data. WaterML is an open specification, interoperable for example with the Geoscience Markup Language (GeoSciML), and it has a diverse glossary, which allows it to be applied on several scenarios. However, the WaterML schema is complex, which makes it hard to understand. A lot of tests have to be made in order to ensure that the standard is properly implemented, and there are no generally-accessible tools [8].

GML is a language based on XML that can be applied to share, model and store geographical information. Some of the benefits that this language presents are the possibility to model different data from distinct areas, supporting the description of entities and attributes [27].

The WaterML project leveraged the results from another project, namely the Hydrologic Information System (HIS), that started in 2005 with a Consortium of Universities for the Advancement of Hydrologic Science, Inc (CUAHSI), aiming to give access to several hydrologic data repositories owned by a different number of entities, such as United States Geological Survey (USGS), the National Water Information System (NWIS) and the US Environmental Protection Agency's STORET (Storage and Retrieval), through web services. Given the increasing demand, it was necessary to develop a system to check the information from different points of observation, so the WaterML 1.0 standard was created [28].

However, there was still one major flaw in WaterML 1.0, namely the incapability of interoperability with other systems. In order to correct that, the Observation & Measurements (O&M) specifications by the OGC were considered, creating the WaterML 2.0 standard. The updated version is a candidate specification to codify data from hydrologic observations [14]. It is important to state that the O&M - ISO 19156 is an information model used to describe observations.

According to the WaterML 2.0 discussion paper [7], the best definition of observation is “...an act associated with a discrete time instant or period through which a number, term or other symbol is assigned to a phenomenon. It involves the application of a specified procedure, such as a sensor, instrument, algorithm or process chain. The procedure may be applied in-situ, remotely, or ex-situ with respect to the sampling location. The result of an observation is an estimate of the value of a property of some feature”.

To describe water observations, the WaterML 2.0 standard defines five components: (i) Time series, (ii) Observation specializations, (iii) Procedures used in measurement/analysis/processing, (iv) Observation metadata, (v) Location description and collections [7].

The WaterML 2.0 proposal is presented in a four-parts discussion paper: the first part regards timeseries, while the second is about ratings, gaugings and sections. The third part concerns surface hydrology features and the last one, GroundWaterML 2., concerns with groundwater data.

Other standards replace the WaterML 2.0 in terms of logical models, by following the norm ISO 19109, which sets the rules for the schema application regarding geographic information. Examples include, the GeoSciML from the Geological Surveys, the EuroRoadS, the CityGML, the Norwegian SOSI, the German AFIS-ALKIS-ATKIS, the Dutch NEM 3610 and the Swiss INTERLIS [27].

2.3. Geographic Information System (GIS)

According to ISO 19101-1, a GIS is an information processing system, that provides and distributes information concerning phenomena associated with a location relative to the Earth [29]. It integrates several types of data (mesh, vector, matrix), supports different spatial analyses operations, and organizes layers of information applying maps and 3D scenes.

Figure 9 illustrates the mechanism used to visualize the geographic datasets, within GISs, namely the layers. Each layer is saved as a file or as a record in a database [30].

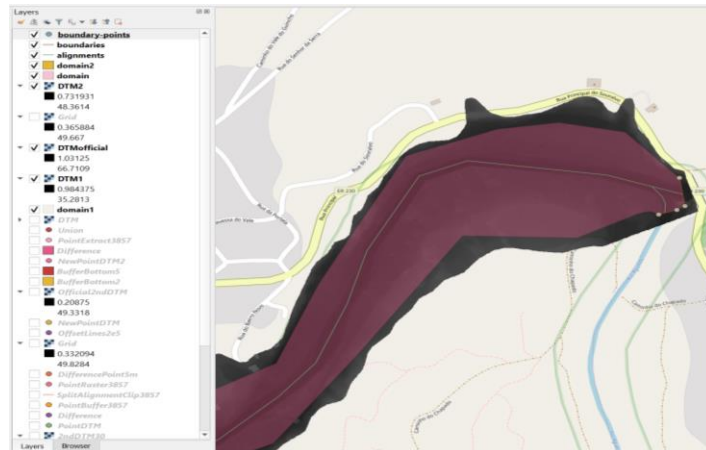


Figure 9: Example of layers in QGIS.

Raster data is a very common type of data in the GIS world. It is composed by a matrix of cells (also known as pixels), that have in each cell a value that represents the existing conditions in the area covered by the pixel. The size of each element is the same in all parts of the layer. A raster layer with a high resolution has smaller cell sizes and higher feature spatial accuracy. However compared with a lower resolution layer it takes longer to process the data and it has a larger file size, which consequently implicates that a larger amount of computer memory is required [31].

Another type of data used in GIS is the **vector** data, which “*is stored in the computer memory as a series of x, y coordinate pairs*” [32]. This type of data is used to represent points (e.g. stations), lines (e.g. rivers) and polygons (e.g. buildings). To represent a real-world object, a feature is defined in terms of geometry (point, polyline or polygon) and attributes (description of the feature). The attributes of a feature are stored in a table, in which a line corresponds to a single record. ESRI shapefile is the most popular format used to save vector data. This file is divided into three parts, namely the main file, an index file and a dBASE table. The main file contains multiple records, each one describes a shape with a list of its vertices. The index file stores the index of the feature geometry. The attribute information of each feature is kept in the dBASE table. For each record, there is only one geometry and one set of attributes [33].

A **mesh** is an unstructured grid that contains spatial components such as vertices, edges and faces [34]. Every vertex, edge and face are identified with a specific string. One benefit of a body-fitting unstructured triangular mesh is that it optimizes the computational workload [35]. A vertex (or a node) is an XY(Z) point, that is capable to store datasets values, with or without a temporal dimension [34]. An edge (or an arrest) is responsible for connecting pairs of vertices and it can be used to calculate fluxes. To do it is necessary to know the volumes of its direct neighbours and to ensure that the information from a specific element only propagates to its immediate neighbours. The conventional Courant-Friedrichs-Lewy (CFL) condition ($CFL_{max} \leq 1$) on each edge grants that last requirement [36]. A face (or cell) is set of edges that form a closed shape, normally a triangle or a quadrilateral. This element can have different sizes in the same mesh, rendering this way different detail levels according to the area of interest, as can be seen in Figure 10. For instance, if the area that is being studied is a section of a river, this site will most likely have a finer cell size [35].

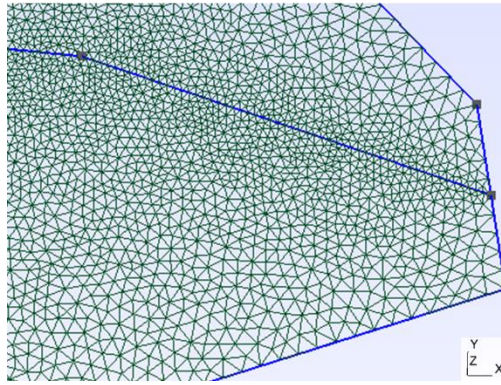


Figure 10: Example of a mesh.

2.4. HiSTAV

This section presents the basic concepts and some background information regarding the simulation tool, named HiSTAV.

HiSTAV, a high performance version of the Strong Transients in Alluvial Valleys model, is a high-resolution simulation tool that uses a 2DH (two-dimensional horizontal) mathematical model based on shallow-water equations featuring dynamic bed geometries and sediment transport to simulate fluvial and estuarine flows. The possibility of forecasting an overland tsunami and the development of a system that could assist in the decision-making process regarding water-related hazards were some of the motivations for the creation of this hydrodynamic model [36]. One important feature of this tool is that it can track uncoupled materials between consecutive timesteps, namely large debris, such as vehicles and containers. This property allows detecting areas where the debris are more likely to accumulate, thus creating a new obstacle [35]. It is important to state that this model is faster than real-time tsunami forecasts [36].

The model uses the finite-volume discretization scheme, which divides the domain into many control volumes (or cells) and approximates the integral conservation law on each of the control volumes [36]. This eulerian mesh-based approach was chosen to allow a proper representation of the conservation laws in cases of discontinuities in the domain [37]. By applying this method, the numerical discretization can interact properly with GIS tools (e.g. QGIS) [38].

HiSTAV has some additional programs (add-ons), that have the function of supporting the user when defining the geometric entities needed for the creation of the mesh and refining them with the assist of a GIS tool [37]. There are two types of utilities, namely for pre-processing, which includes a GIS software (e.g. ArcMap) and the mesh refinement, and for post-processing, which consists of the resampler and raster converter.

The ArcMap add-on is of great importance since it allows analysing, viewing, editing and creating vector or matrix data with a more user-friendly interface. One of the features of this extension is that it is capable of converting boundary polygons defined in ArcMap into a format compatible with Gmsh. This work used the QGIS software, which has similar functions to the commercial tool named ArcMap

but, it is open-sourced. Another relevant pre-processing sub-program is the mesh refinement add-on because it allows the user to adjust visually the refinement levels in certain locations without having to do it for the entire domain. The mesh generator that is incorporated in the pre-processing utilities is the open-source code Gmsh [37].

The post-processing toolkit was defined so that the integration with a GIS is functional [37]. One of the tools is the VTK resampler which is used to interpolate flow variables (e.g. water velocity) between two meshes. One of the advantages is that the obtained results from it can be refined and applied in the simulation process as an initial condition. The other mechanism is the raster converter which is responsible for the importation of HiSTAV outputs in GIS environments.

Figure 11 gives an overview of the simulation process, which normally starts with the preparation of all the data that is going to be applied in the model through a pre-processing tool. Then, after the control files are created, the data processing stage can start, which is the execution of the HiSTAV. Lastly, upon obtaining the simulation output, it can be viewed in the post-processor software.

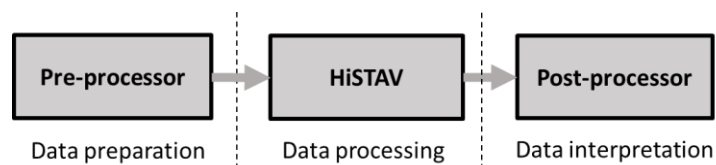
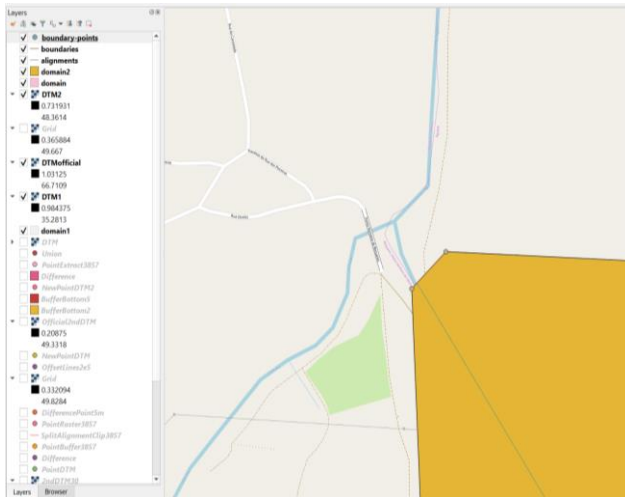


Figure 11: HiSTAV data flow schema.

To begin the simulation, it is necessary to collect and define all the data inputs that are essential for the correct execution of the model, so that the best simulation output can be obtained. It is important to have files stating the initial conditions, boundary conditions, physical constants, numeric and time information. Besides the indispensable information, one can also consider other data inputs, such as distributed conditions (e.g. radar timeseries).

A mesh is used to define the initial conditions and boundaries. To generate a mesh, it is necessary to first define the domain and the boundary conditions using QGIS [39]. To delineate the boundary conditions (Figure 12 a)), inlet or outlet, the boundary lines and points must be identified. Figure 12 b) shows that each point is assigned a specific timeseries, and that each line is associated to the parameter (e.g. flow, height) present in the timeseries. Next, the pre-processor (Figure 13 a)), corresponding to a mesh generator tool is executed, leading to the importation of shapefiles and raster files, such as, Digital Terrain Model (DTM), domain, obstacles and boundaries. When the process is concluded, the outputs including the mesh file and the files with the initial conditions are saved and become ready for the simulator tool.

a)



b)

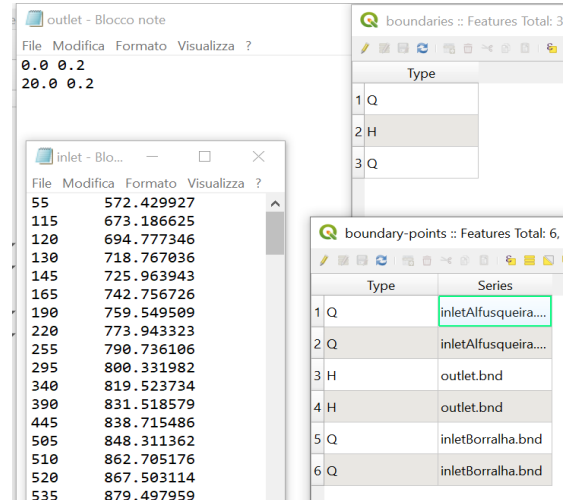
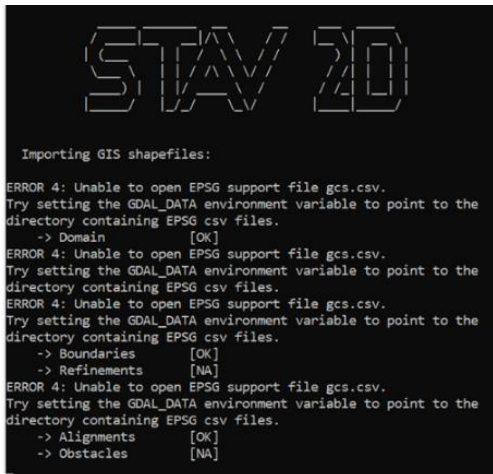


Figure 12: a) Example of an inlet definition in QGIS; b) Boundary conditions definition.

The initial conditions are defined in two ASCII files, one contains information regarding the altimetry and the other about the resistance.

Figure 13 b) presents the control files, which includes the physical constants, numeric and time information, that must be prepared before the simulation. To track the diverse materials the model uses several formulas with numerous physical constants. Some of these are general (e.g. gravitational force) and some concern a specific material (e.g. sediment diameter). Figure 13 b) shows that there is a “physics” ASCII file, that contains physical constants, namely the gravitational force (9.81).

a)



b)

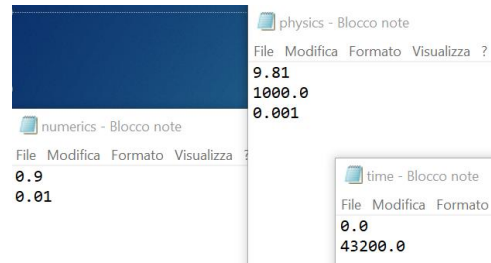


Figure 13: a) Pre-processor running; b) Physics, numerics and time files.

Upon the definition of all the parameters, and the preparation of the input data, the HISTAV tool can be run (Figure 14 a)), and it will generate an output folder with several files, just like the one in Figure 14 b).

a)

```

STAV 2D

-> Importing control files ...
-> Reading ./control/time.cnt
-> Reading ./control/physics.cnt
-> Reading ./control/numerics.cnt
-> Reading ./forcing/forcing.cnt
-> Reading ./initial/initial.cnt
-> Reading ./output/output.cnt

-> Importing boundaries ...
-> Reading ./boundary/meshData/boundaryGim.bnd: 3 boundaries
-> Reading ./boundary/boundary.cnt

-> Importing mesh ...
-> Reading mesh.info
-> Reading nodes [=====] 100 %
-> Reading elements [=====] 100 %

-> Mesh dimensions: 328478 elements and 165952 nodes.
-> Allocated memory: 181 MB of RAM.

-> Reading ./bed/bed.cnt

-> Setting initial conditions
-> Setting IC's [=====] 100 %

-> Setting boundary buffers

dt (s): 11.021754, CPU (s): 0.094, Ratio: 117.565, Time (s): 11.022, Comp. (%): 0.026, OMP: 8
Writing ./output/hydrodynamics/hydro-1.vtk ... Done.
dt (s): 11.969583, CPU (s): 22.250, Ratio: 0.514, Time (s): 22.382, Comp. (%): 0.052, OMP: 8
dt (s): 11.350155, CPU (s): 0.063, Ratio: 181.602, Time (s): 33.732, Comp. (%): 0.078, OMP: 8

```

b)

Nome	Stato	Ultima modifica	Tipo
hydro-1			
hydro-15782			
hydro-25335			
hydro-27701			
hydro-30041			
hydro-32343			
hydro-34614			
hydro-36854			
hydro-39076			
hydro-41295			
hydro-43501			
hydro-45696			
hydro-47871			
hydro-50050			
hydro-52256			
hydro-54493			
hydro-56757			
hydro-59047			
hydro-61348			
hydro-63665			
hydro-65995			
hydro-68324			
hydro-75350			
hydro-77697			
hydro-80041			
hydro-82365			


```

File Modifica Formato Visualizza ?
# vtk DataFile Version 2.0
STAV-2D solver at 0.214285 secs.
ASCII
DATASET UNSTRUCTURED_GRID

POINTS 165989 FLOAT
-945735.937500 4950006.000000 0.000000
-945719.187500 4949995.500000 0.000000
-945727.562500 4950001.000000 0.000000
-945716.812500 4950006.500000 0.000000
-945421.000000 4949735.500000 0.000000
-945427.937500 4949743.000000 0.000000
-945417.687500 4949743.500000 0.000000
-945455.687500 4949771.500000 0.000000
-945462.625000 4949778.500000 0.000000
-945434.875000 4949758.000000 0.000000
-945448.750000 4949764.000000 0.000000
-945441.812500 4949757.000000 0.000000
-945428.437500 4949753.000000 0.000000
-945421.125000 4949751.500000 0.000000
-945432.437500 4949759.000000 0.000000
-945424.250000 4949759.500000 0.000000
-945419.375000 4949759.000000 0.000000
-945419.375000 4949766.000000 0.000000
-945446.437500 4949773.000000 0.000000
-945438.000000 4949767.000000 0.000000
-945438.750000 4949775.500000 0.000000
-945444.875000 4949780.500000 0.000000
-945452.562500 4949778.000000 0.000000
-945434.562500 4949783.000000 0.000000
-945430.312500 4949775.000000 0.000000

```

Figure 14: a) HiSTAV running; b) Example of a simulation output file.

ParaView⁵ is one of the possible software tools that can be employed for results post-processing, and it was chosen because it is an open-sourced tool that allows an interactive scientific visualization of the outputs obtained from the simulator (Figure 15). With this tool it is possible to see the water level of a river over a period, for instance, allowing this way to evaluate the possibility of a flood occurring.

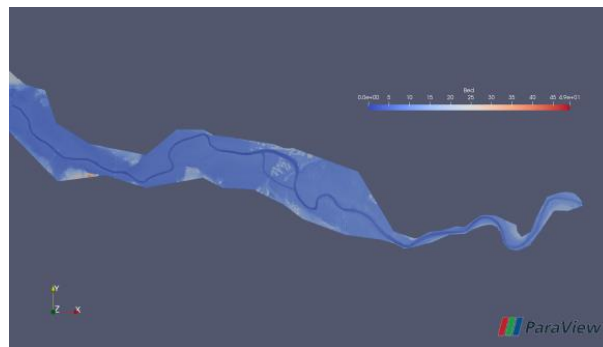


Figure 15: Example of a simulation result (bed elevation) in ParaView.

The strong points of HiSTAV include following features [36]: i) availability for engineering and scientific research communities; iii) easiness of integration with open-sourced GIS platforms; iv) compatibility with scientific visualization toolkits for data analysis.

2.5. Requirements Engineering

Requirements models, such as those defined with UML, are commonly used to produce conceptual models, that contain the requirements of a system [40]. These models allow a much better understanding of the information compared with descriptions written with just natural language.

⁵ <https://www.paraview.org/>

The document that describes the multiple technical concerns of a software system is called a System Requirements Specification (SRS) this document is used during the whole life of the project so that the focus of the project is not lost and to avoid misunderstandings among the stakeholders [6].

2.5.1. Basic Concepts

According to C.Pohl [40], Requirements Engineering (RE) is “*a systematic and disciplined approach to the specification and management of requirements, which should follow two goals, knowing and documenting the relevant requirements, using the standards imposed; and achieving a consensus among the stakeholders while ensuring that their desires and needs are satisfied*”.

Another important concept is the process of software development which, according to A.Silva and C. Videira [9], is a vague concept that describes a sequence of activities, grouped in tasks, that are executed in a systematic way and from a set of entries produce a set of outcomes.

A model is defined as an interpretation of a certain problem domain according to a specific concept structure, as stated by A. Silva and C. Videira [9].

There are three types of data models, namely conceptual, logical and physical. Conceptual models identify the general concepts of a system, with the definition of the important entities, without detailing their attributes or primary key. This type of model allows easier communication with the stakeholders. A logical model is a more refined conceptual model that defines all the entities and its attributes, primary key and foreign keys. The physical model type is associated with the implementation of a database management system, it presents all table stereotypes, attributes, primary key, foreign keys, constraints and relations between tables [9].

A. Silva and C. Videira [9] define requirement as a statement regarding a functionality or a condition, which the system must attend. There are two types of requirements, namely functional, that describe an action that should be made by the system, and not functional, that concern transversal aspects that are normally applied to the whole system.

2.5.2. UML

The Unified Modeling Language (UML), is a language that allows specification, construction, visualization, and documentation of objects in an information system. It was created in 1996 after an attempt to unify the most significant object-oriented modelling languages at the time, namely the Boosh, the OMT (Object Modelling Language), and the OOSE (Object Oriented Software Engineering). In the context of the Object Management Group (OMG) in 1997, the UML version 1.0 acquired the status of standard. One of the advantages of this language is that it has extension mechanisms which allow future modifications. Other benefits include the fact that it can be applied to many different domains, merging many distinct diagrams from different languages, and is independent in terms of process and of modeling tools [9]. The most recent version is the UML 2.5.1⁶and it was




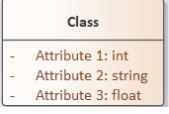
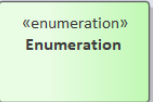
⁶ <https://www.omg.org/spec/UML/2.5.1/>

launched in December 2017. The model's elements can be categorized in three groups: classifiers, which describe a set of objects, event, that describes a set of situations which will cause consequences to the system, and behaviour, which describes a set of possible actions that can lead to the occurrence of events [41].

In the UML 2 there are diagrams for behavioural modelling (e.g. use cases, activity diagrams) and for structural modelling (e.g. class and packages diagrams). In this thesis, only use cases and class diagrams were used.

Some of the basic elements of UML representation are defined in Table 1.

Table 1: Examples of UML elements.

Diagram type	Name	Representation	Definition
Use Cases	Actor		Usually, <i>“is a representation of a person or a system that interacts with the system”</i> [9].
	Use Case		A sequence of actions that one or more actors make in the system to obtain a certain result [9].
	System Boundary		<i>“Defines conceptual boundaries and helps to group logically related elements”</i> (e.g. use cases) [42].
Class	Class		A description of a group of objects, which consists of structural features, also known as attributes, and behavioural features, or operations, which define the interaction of the objects [43].
	Enumeration		<i>“Enumerations contain sets of named identifiers that represent the values of the enumeration”</i> [44].

A class can be related to others, through a relationship of inheritance, association, aggregation, composition, or dependency [43]. The aggregation association is used to indicate that an element contains or is composed of other elements, while the composition relation is a stronger type of aggregation, *“where the child cannot exist independent of the parent”* [45].

A use case diagram can have two types of relationship between use cases, namely include and extend. The *include* relationship is used to incorporate the behaviour of other use case related. The *extent* relation is commonly known as the can relationship since it is applied when a user sees the case as optional or when a certain condition is met [9].

The UML element named package refers to *“A collection of UML elements”* (e.g. group of classes) and it is used for organizing large systems [28]. In this work, it was applied in the class and use cases diagrams, in order to simplify them and to assure a better understanding of the subject matter.

2.5.3. ITLingo and RSL

ITLingo is a long-term initiative aiming to research, develop and apply rigorous specification languages in the Information Technology area, namely in technical documentation [5]. It focusses on three subjects: Requirements Engineering, with a RSL (Requirements Specification Language); Testing Engineering, with a TSL (Testing Specification Language); and Project Management, with a PSL (Projects Specification Language [5]. This project will only use RSL language.

ITLingo RSL is a requirements and tests specification language. RSL is based in several languages, such as ProjectIT-RSL [50, 51], XIS*, i*, Pohl, SilabREQ, RSL-IL (Intermediate Language) [48] and RSL-PL (Pattern Language) [5]. The RSL-IL was applied to formally represent requirements, and RSL-PL used to define linguistic patterns [49].

RSL is a language with the goal to improve the definition, production and documentation of requirements specifications, making them more consistent, clear and rigorous [5]. An system requirements specification with RSL can have distinct writing styles and requirements types, such as business goals, systems goals, functional requirements, user stories and use cases.

RSL is supported by ITLingo-Studio, i.e. an Eclipse-based tool [50]. ITLingo-Studio allows to import Microsoft Excel files, PlantUML diagrams and can export to Microsoft Excel and Word. Also, it can create UML Use Case Diagrams, UML State Machine Diagrams, UML Class Diagrams, UML Behaviour Diagrams, UML Sequence Diagrams, and SysML Requirements Diagrams. These diagrams will be stored in a special folder, containing a text file with the corresponding PlantUML specification [51].

Most of the errors that occur in the process of SRS's creation are due to the use of natural language, although this is still the best option to specify requirements [10]. Thankfully, one of the features of the ITLingo-Studio is that it can import an Excel file to validate the requirements. Although the quantity of errors lowers with this tool compared with others, there are still mistakes, so human validation is needed.

Figure 16 shows that this approach has two different levels: a process level, that consists in the conversion of information from the RSL-PL format to the RSL-IL format, and a project level, which is dedicated to writing the requirements specification in a natural language.

Throughout the whole process, the Requirements Engineer and the Business Stakeholder are the two major participants. They have distinct roles: the first one acts as a facilitator with the business stakeholder, so that the best requirements can be found, and the process takes less effort. The second player, in turn, is essential to develop a proper project, since it is a major source of information and an excellent evaluator of requirements in order to validate them. Together, they are responsible for the textual inputs (Ad-hoc NL, Requirements Specs and Glossary) [52]. RSL was implemented with the Xtext framework, with the intention of being able to be modified into several formats and representations for the specifications, but also to validate them automatically [5].

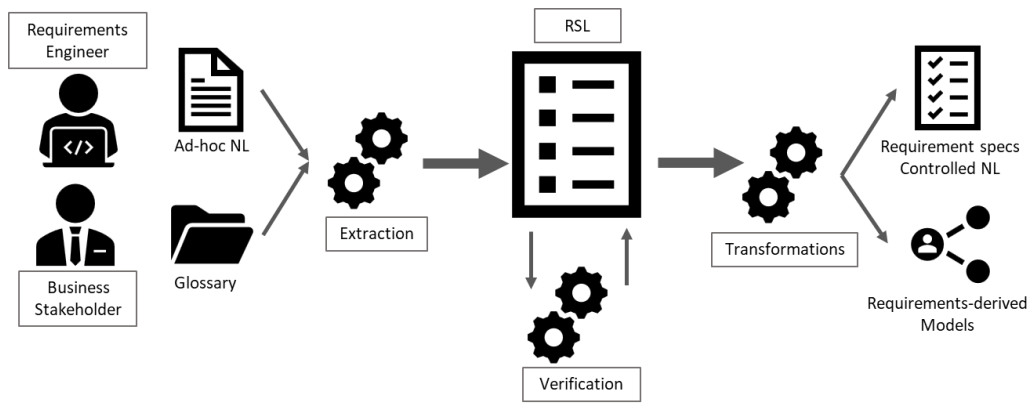


Figure 16: Overview of the RSLingo's process (adapted from [52]).

The RSL constructs are classified according to two standpoints [5]: Abstraction Level and Specific Concerns. The first one includes Business, Application, Software and Hardware. Consequently, the second perspective is composed by the active structure, passive structure, requirements, tests, relations & sets and other concerns, as it can be seen in Table 2.

Table 2: RSL constructs' classification [5].

System	Concerns	Active Structure (Subjects)	Behaviour (Actions)	Passive Structure (Objects)	Requirements	Tests	Other	Relations & Sets		
	Abstract Levels									
Business	Stakeholder	ActiveElement (Task, Event)	DataEntity	Goal QR Constraint	Acceptance CriteriaTest UseCaseTest	Glossary Term	SystemsRelation RequirementsRelation TestRelation			
Application	Actor	StateMachine	DataEntityCluster	DataEnumeration	FR UseCase UserStory	Risk Vulnerability	SystemElementsRelation ActiveFlow			
Software								DataEntityTest	Stereotype	SystemView
Hardware								StateMachine Test		SystemTheme
Other						IncludeAll IncludeElement				

A GlossaryTerm can be defined as a noun, verb, adjective or adverb in the RSL. It also allows setting possible synonyms and acronyms. Besides that, it states the possible applications of the terms, which can be to Stakeholder, System, Actor, DataEntity or Other. A description and the relation between terms can also be added to the term definition.

In the RSL, there are six types of stakeholders: Organization, OrganizationalUnit (applied to departments of companies, for example), Team, Person, System (it can be sub-categorized as internal or external) and Other. The ITLingo-Studio also allows users to indicate if a stakeholder is part of another stakeholder (e.g. a person works for a company) and if they are a special type of stakeholder (e.g. a systems administrator is a user).

The RSL allows to classify an actor as a user, an external system or a timer, and to state the stakeholder associated (optional).

A DataEntity is an *“individual structural entity that might include the specification of attributes, primary keys, foreign keys and other check data constraints”* [5]. A primary key is one or more attributes that are able to identify univocally an object [9]. A foreign key is one or more attributes that identify a relationship between different entities [53]. There are five types of DataEntities in RSL [5]:

- 1- Parameter (To define functional or behavioural parameters. Data that is composed by just one item and where the attributes are values for settings);
- 2- Reference (Simple data in a small quantity, for example, postal codes);
- 3- Master (Data assets of the business, such as people, places and concepts);
- 4- Document (Operational data of the business, with a complex structure);
- 5- Transaction (Operational data of the business, for instance, balances).

If a DataEntity depends on another entity, this means that its sub-type is “Detail”. However, if it has its relevance and identification one considers that the sub-type is “Master”. Each entity has attributes attached to it and each one has a name and a type connected. It can also have its multiplicity explicit, values, default values and several properties, such as, if it is not null (isNotNull), unique (isUnique), encrypted (isEncrypted) or to read only (isReadOnly).

The DataEntityCluster is a *cluster of several structural entities that presents logical arrangements among themselves*. Each cluster has several entities, each with their specific role. All clusters must have one “master” that is the main entity if it a simple cluster beside the master it also must have at least one “reference” (represents an entity that depends or is used by the main one). A complex cluster has to have a master entity and at least one “detail” entity (represents a partOf or a child DataEntity) [5].

To characterize a use case, in RSL, some additional aspects can be defined, such as [5]:

- Use case type (e.g. EntityDashboard, EntitiesManage, EntitiesInteropExport, EntitiesMapInteract);
 - o EntityDashboard – *“Dashboard and data analyze based on a dataentityview's item”*;
 - o EntititesManage – *“Manage (CRUD operations) dataentityview's items”*;
 - o EntitiesInteropExport – *“Export dataentityview's items to an external data source”*;
 - o EntitiesMapInteract – *“Show dataentityview's items as a geographical layer supported by GIS technology and supporting some interaction features (e.g., zoom-in)”*.
- DataEntityCluster to which is applicable;
- The actor that initiates the action (obligatory) and actors that participate in it (optional);
- Preconditions and/or post-conditions that must be fulfilled;
- Actions that can occur (e.g. Search, Create, Read, Delete, Cancel);
- Relationships with other use cases, using either the “include” or “extend” relation.

3. Bibliographic Review

This Chapter presents an overview of some articles and related work regarding water resources, in Section 3.1. In turn, Section 3.2. describes previous work on requirements engineering. Section 3.1.2 and 3.2.2 overview and discuss the topics approached in Section 3.1 and 3.2, respectively.

3.1. Water Resources

Water is such an important resource for humanity, which makes its management of the utmost importance. To manage it correctly, efficiently and sustainably, many systems, models and standards were created. Each one has different characteristics, domains and goals, such as allowing the population to access certain data and being able to perform simulations with the data available on the system.

3.1.1. General Aspects

N. Charneca in [27] presented the process of development and implementation of a geographic data model that supports the planning and management of superficial water. This takes into consideration all the technical requirements and legal rules, national and international, such as Infrastructure for Spatial Information in the European Community (INSPIRE)⁷, ISO 19103⁸, ISO 19109⁹ and the Water Information System for Europe (WISE)¹⁰. The project had four stages: i) Identification of relevant rules and specifications regarding geographical information; ii) Definition of the conceptual model; iii) Development of the application model; iv) Implementation of the physical data model. Since the project is object-oriented, the objects are linked to each other and grouped in classes, in which they have their own attributes. In this work the UML language was applied to develop the data model, the GML language was researched to check the viability of using it to describe several object categories, and the WaterML was analysed as a possible specification for the development of the UML classes' diagrams of the application model. The geographic data model developed in this thesis allowed to represent physical objects coherently, validate tools for planning water resources scenarios, share data regarding the management of water resources and create a single data repository, since the legal, functional and technical requirements were merged.

A. Almoradie *et al* [14] tested the viability of publishing datasets related to water resources on the web using the **WaterML 2.0 - GeoServer**, a combination of the WaterML 2.0 standard and the Commonwealth Scientific and Industrial Research Organization's (CSIRO) GeoServer Web Feature Services (WFS). The original server has the advantages of being open-sourced and re-configurable. By simply installing plug-ins, for updating schema files (e.g. WaterML 2.0 structure), servers'

⁷ <https://inspire.ec.europa.eu/>

⁸ <https://www.iso.org/standard/56734.html>

⁹ <https://www.iso.org/standard/59193.html>

¹⁰ <https://water.europa.eu/>

properties and database structure, the WaterML 2.0 - GeoServer was created. The data is structured in four parts in this framework: timeseries, geographical information and geometry, details from the data provider, and stations' details. A case study was used to demonstrate the system with data from the Somes Mare basin in Romania. A web-based Flood Information System (FIS), called Some Mare FIS Portal, was created using the WaterML 2.0 - GeoServer, allowing the access to flood information, the participation of citizens, by discussing related issues and reporting floods, and raising awareness to flood risk management. This portal has many components, such as, the "Data Access" component, which contains spatial and timeseries data, and the "Hydrometeorological Data", that leads to a map-based interface (using Google maps, in this case) that shows the monitoring stations and their information regarding precipitation, discharge or temperature. The downside of this implementation is the vast quantity of configurations needed to reach the functionalities.

The **Sava GIS Geoportal** is a web application for sharing data regarding water resources and water management in the Sava River Basin [54]. It contains a metadata catalogue that is responsible for "*finding, analysing and using datasources of spatial data*" [54]. This platform has an interactive map with several layers that contain different types of information regarding river basin management, flood risk management, hydrological data and meteorological data. This platform can interoperate between open sourced (e.g. PostGIS or GeoServer) and proprietary software (e.g. ArcGIS) since it follows open industry standards and protocols, which is a major advantage. There are five types of users defined, including the public users which can export areas of the map, view public spatial data, attributes and features without login in. The International Sava River Basin Commission (ISRBC) WebGIS Viewer is able to view and download data from the Sava Web GIS module, and the ISRBC WebGIS Editor can view, upload and download a restricted set of data defined according to a specific country. The ISRBC Metadata User can view, upload (e.g. GML format) and download metadata in several format focusing on attribute (e.g. WaterML 2.0) or spatial (e.g. GML2) data. However, the metadata user's permissions can be restricted in terms of data access, which can be limited to a specific set from a specific country, and according to the specified user role specified. The administrator is responsible for managing the users, which includes the definition of the users' permissions on the geoportal. Connected to this geoportal, there is another webpage, named Sava HIS, which is a portal that presents on an interactive map real-time data acquired from hydrological and meteorological stations from the Sava river basin [54]–[56].

The **Iowa Flood Information System (IFIS)** is an online platform that allows checking the weather, flood inundation maps, real-time flood conditions, flood forecasts, flood-related data, information, applications and interactive visualizations for the population [60]. Several system requirements were defined, including the fact that the platform must be user-friendly, interactive, accessible through many different devices and it must not require technical skills or external applications. Data in real-time, from 500 stream sensors and from different organizations, is collected every 15 min, saved for 10 days and available on the web-platform. Four flood levels (action, flood, moderate flood and major flood) with different colours are associated with these sensors and are triggered when the values registered values are higher than the threshold defined by the National Weather Service and the Iowa Flood

Center. The system is also able to present graphs of the time-series from the last six days and present interactively the data from the last ten days. Data from rainfall is acquired by Weather Surveillance Doppler Radars and it can be visualized in the portal. Every twenty minutes hydrological models are run, and the results are uploaded to IFIS, providing this way several flood inundations scenarios and an hourly prediction of the stream conditions. An important aspect of this portal is the definition of a basin boundary which includes the rivers and the communities that are located near streams. One major advantage of this system is its adaptive capability and integrated structure, which allows the system to be applied to other geographical areas and different environmental domains. With this platform, the users are able to monitor the community for floods through a single webpage, creators are encouraging the data exchange and collaboration, and decision-makers can make more informed decisions, with the help of real-time data and forecasting models [57]–[59].

The **Hydrologic Engineering Center Data Storage System (HEC-DSS)** by the US Army Corps of Engineers is a database system that stores data, especially sequential data (e.g. timeseries). This system has an MS-Excel add-in that permits to read and write regular interval timeseries and supports modelling software, such as the HEC-HMS [60]. To be able to visualize and alter the data in this database, a program named HEC-DSSVue is used, allowing this way to edit data sets and to obtain graphs, for example. One of the features of the HEC-DSSVue is the ability to add plug-ins, which for instance support retrieving data from other databases and importing data from MS-Excel files or WaterML document [61].

The **CUAHSI HIS** was developed with the intent to provide web services, tools, standards and procedures to improve hydrologic data analysis and access. An internet-based system was created, allowing different entities to share hydrologic data, by using web services to connect servers and databases to applications that allow the publication, search and access of data. This system is composed by three types of computers: i) HIS Central: functions as a catalogue that contains the metadata associated with the time series, which is linked to the owners' server, ii) HydroServer: saves, organizes and publishes data, iii) Client (e.g. HydroDesktop): an interface to access the data, collecting metadata from the HIS Central and data from the HydroServer. The HydroServer has several components, such as the WaterOneFlow Web Services, that allow access to the ODM Database via the web, and some parts of the ODM model (e.g. ODM Database, ODM Tools and ODM Data Loaders) [62].

The **ODM** by CUAHSI [3] is a relational database schema that aims to store and retrieve water data, specifically time series of observations acquired from experimental sites. The data stored from observation points is saved and associated to its metadata, which is composed by date, time, variable name, location, units, interval, offset (distance between the reference point and observation point), reference point, data type, organization (the entity that provides the measurement), data qualifying comments (can be helpful when interpreting it), analysis procedure, source, sample medium, quality control level and value. Although this model saves data for individual nodes and links, it does not allow users to relate them [63]. Tools can be associated, allowing users to export data and metadata, to visualize the data in plots, to modify data, to edit the controlled vocabulary and to use web services.

WaterOneFlow (WOF) is another product from CUAHSI HIS, corresponding to a web service that aims to transmit and receive information from other computers via the web, using applications. It has four main functions: GetSiteInfo (returns the metadata of a specific site), GetSites (returns the metadata of the several sites requested), GetVariableInfo (returns variable's information) and GetValues (returns a specific time series) [28]. Several applications allow importing data directly to their systems, such as MS-Excel, Matlab and ArcGIS. This web service can be used to access the ODM Database, but also other remote repositories of observation data [28]. It is important as well to highlight that WOF is used in the HydroServers and consequently, its metadata is also linked to the HIS Central, by being part of the central's catalogue.

A. Abdallah and D. Rosenberg [63] state that there are many data systems/models, such as the HEC-DSS, Hydra Platform, Arc Hydro, Observation Data Model (ODM), MS-Excel, Water Evaluation and Planning System (WEAP) and RiverWare. However, they do not fulfil all the requirements. For instance, some allow network connectivity and others do not. To solve the problems that these systems/models have the authors proposed the creation of a generalized data model, named the **Water Management Data Model (WAMDaM)**. This model will *“help organize, join, compare and analyze multiple water resources, datasets and models”*. The authors defined eight requirements for the database to be successful: it must have an extensible design allowing to aggregate several model types and reuse the components of the systems as data objects, which means that a value of an attribute can be shared with many water resources systems components, for example. It also must allow the representation of networks of nodes and links that will help to search for systems components. The model should create scenarios in order to cover possible changes, physical, operational or social-economical and it enables the users to add comments so that the components are easier to comprehend. The fifth requirement that the developers of the model took into consideration was the possibility to save and define distinct types of data (e.g. time series, multi-attribute, numeric, categorical values, seasonal parameters) and the sixth refers to the need to control the vocabularies. The seventh states that the data and metadata must be accessed directly, this allows the users to load and retrieve subsets of data and the last requirement is the implementation using open-source software tools. This last aspect will possibly lead to an easier process of further development of the model. Upon the implementation of the previously stated requirements, the model was demonstrated using thirteen different datasets and models. In the end, after implementation, the proposed model unified data in terms of format, structure and vocabulary of fifteen attributes from six different sources.

3.1.2. Discussion

The papers presented in the water resources section describe several data models/systems, which have distinct features. Some of the features are present in more than one system, which reflects their importance and the impact of the benefits associated.

Many systems allow the association of other tools, systems or applications, which allows adding new features, that can improve the service provided and the users' experience. For instance, HEC-DSS

allows adding HEC-HMS as an extension, which allows the user to use a simulate a scenario using the data in the database.

The ability to present real-time data acquired from monitoring stations is a feature of the Sava GIS Geoportal and IFIS systems have. With this feature available it is possible to better forecast floods and to manage water resources more efficiently.

HEC-DSS and IFIS are able to support simulation scenarios, which is an important feature because it can help define proper plans to mitigate the consequences of a flood and prevent major consequences in the communities.

Throughout the several papers, the importance of the vocabulary used and the application of standards in the different systems and models are stated. The WaterML and the ODM were the ones referred for standardizing terms, despite the similarities, they are not equal. Detailed information, such as source information is included in the WaterML and not in the ODM [28]. By following a standard, it becomes must easier to understand a system, to interoperate between software and to improve the system.

Both, the IFIS and the Sava GIS Geoportal provide tools that allow the public to analyse and visualize data regarding water resources, which can bring major benefits, such as raising a higher level of awareness among the population regarding floods, namely, its risks and consequences.

The open-source feature is present in many of the models and systems that are previously shown, which proves that its benefits have an impact on the systems developed and consequently on the scientific community. This feature will allow raising more awareness to the topic, namely by allowing faculty students to use these systems and to facilitate discussions regarding the systems, improving the system/models faster [64].

The CUAHSI HIS, by allowing the combination of the different products such as web services, tools, standards and procedures, was able to provide more consistent hydrologic data, making the process of data discovery and analysis between many sources more efficient and accurate [64].

3.2. Requirements Engineering

There are several languages and tools that help specify the requirements of the systems, each with different levels of formality and rigor. A model can be defined using either a general-purpose modeling language or a more requirements-specific modeling language, examples of the two types of modeling languages are presented in this section.

3.2.1. General Aspects

Systems Modeling Language (SysML) by OMG is a general-purpose modelling language used for solving systems engineering problems [65]. SysML uses UML 2.5 as a base for its structure and gives extensions to connect to UML, which brings a benefit, it facilitates the communication between different stakeholders (e.g. system and software engineers). The level of formality, rigours and

understandability are due to the combination of the UML with a precise natural language [65]. The basic unit of the SysML structure is the block, which can be utilized in diagrams to represent hardware, software, personnel, facilities and other system elements. This language has three main types of diagrams, behaviour, requirement and structure diagram. Besides those diagrams exist others that present some modifications compared to the UML 2 diagrams, such as, the activity diagram, which is included in the behaviour diagrams, the block definition diagram, which represents the system hierarchy and classifications, and the internal block diagram (defines the internal structure of a system). New diagram types are possible to be developed in SysML, such as, requirement diagram, which allows connecting classical requirements management tools and system models, and parametric diagram, which is a special type of internal block diagram that represents the system constraints, like the performance and reliability. Rational Rhapsody¹¹, Eclipse Papyrus¹², Visual Paradigm¹³ and Sparx Enterprise Architect¹⁴ are some of the tools that can be used to model in SysML [66, 67].

A. Alanazi et al in [15] present the importance of the implementation of **Requirement Modeling Language (RML)** in system engineering projects, like a CubeSat system. RML is a language used by executive, business and technical stakeholders since it is designed to visually model requirements and it is focused on the project's goals. It has four types of models: objective, system, people and data. The fact that functional and non-functional requirements can be described in the objective models allows a better interpretation and elicitation activities. The object of this case study, the CubeSat, is a miniature satellite that orbits Earth at a low altitude, and it can be used to observe the atmosphere, to track space debris and to detect gamma rays or magnetic fields. To manage the requirement model, a tool was used, the Requirement Mapping Matrix (RMM), which is an RML objective model that can show graphically the systems requirements and business rules. This tool allows to ensure the project progress and to validate and verify all the requirements. The use of the RML in the CubeSat project was important to identify the requirements missing, the ones superfluous and which ones should be prioritized. Moreover, it may improve the quality of requirements, since all the types of stakeholders will comprehend each other and the models much better during the process of defining the requirements.

According to A. Chen and J. Beatty [67], the RML has models that connect the requirements to the business value and models that show the system from the end-user perspective, which the UML does not have. Besides that, the RML is simpler to embrace by a business stakeholder compared with UML, since its models are not configured for the definition of the software structure.

A. Silva and L. Gonçalves [50] proposed to help organizations plan and be aware of security concerns from the starting point so that costs can be reduced and mistakes avoided. Catalogue4CyberSecurity is the name of the proposed RSL, which has followed the Common Criteria for Information Technology

¹¹ <https://www.ibm.com/products/systems-design-rhapsody>

¹² <https://www.eclipse.org/papyrus/>

¹³ <https://www.visual-paradigm.com/solution/uml/sysml-modeling-tools/>

¹⁴ <https://sparxsystems.com/>

Security Evaluation standard, assuring this way that the process of specifying, implementing and evaluating was done rigorously. This work presents an existent RSL language that can specify different types of requirements, such as quality requirements and user stories, and security-specific constructs, for instance, vulnerabilities and threats. Moreover, it developed a catalogue of security concerns, by grouping several packages with reusable specifications [68].

3.2.2. Discussion

Several languages were approached in the previous Chapter and after analysing them it can be said that there are two types of languages, the general-purpose (UML and SysML) and the requirement specific (RSL and RML). The UML is normally used in software engineering, the SysML in system engineering and all the requirement-specific languages are applied in requirements engineering. RSL can describe the constructs (e.g. use cases) in detail, which includes sub-classifying them and indicating the relations between them. This level of construct specification is not applied to the UML and SysML and it is a highly important aspect since it allows the specification to be more rigorous and to prevent possible miscommunications between stakeholders. Even so, the RML has some constructs and types of models, that can be found in UML or SysML, it is a less rigorous and precise than RSL, UML or SysML, according to A. Silva [5]. The graphical representation is used in the UML and SysML languages, and it has some advantages. Namely, it provides a faster understanding of the initial phase of the requirements definition process. However, it can be quite confusing if there are many components specified or if the model has several elements. Some of these languages can be too complex for a business stakeholder since they require technical knowledge. This fact can discourage organizations to embrace these languages.

4. Related Systems

To better understand the specifications of the RiverCure portal it is important to present the SNIRH and SVARH specifications since the SVARH is one of the main inputs of the RiverCure system. The specifications of HiSTAV are also presented in this project since this tool will be integrated into the portal.

This section is divided into two sub-sections: SNIRH and SVARH (section 4.1), and HiSTAV (section 4.2). In each sub-part it is presented the glossary, stakeholders, actors, data entities and requirements of the corresponding system, using RSL language. Some UML diagrams were also created in order to represent the data entities and the requirements. The diagrams allow a quicker understanding of the model compared with the RSL specification, but it presents fewer details. To understand in-depth the data entities, the RSL file must be consulted.

4.1. SNIRH and SVARH

In this section, it is presented the glossary with some of the terms of the system, the stakeholders, the actors, the data entities and the requirements using RSL language. UML diagrams regarding the data entities and requirements of these systems are also presented in this section.

To be able to achieve some conclusions regarding these systems, some assumptions had to be made.

- 1- The SNIRH system was responsible for all the maintenance of the stations. Even so, the SVARH uses stations that require maintenance, those stations are also included in the SNIRH. Note that a station is a place that allows making certain observations/measurements through a set of sensors, which measure different parameters.
- 2- Datalogger, the system that records the data capture by the sensors, is a sub-system of SNIRH.
- 3- The GFiltro software, which is responsible for converting ASCII files and upload them into the database [24] is used for every type of station (with or without GSM) in the SNIRH.

4.1.1. Glossary

A glossary was developed to define the terms used in the systems, in order to avoid communication misunderstandings. Since different systems can have different understandings of the same concept.

In these systems, the glossary defined is very similar since the SVARH is considered a sub-system of the SNIRH system.

Most of the definitions of the concepts presented on the model were based on APA's glossary. However, it is important to state that the terms and their definition on the website were written in Portuguese, so they had to be translated into English.

Hydric terms had to be defined, such as flow, parameter, hydrological station and meteorological station. Other terms related to the system also had to be described to comprehend the model, namely the term user, public viewer, restricted user and system administrator. Some of these terms are described in Specification 1 using RSL.

```
Term trm_user "User" : Noun [ applicableTo Stakeholder, Actor description
"A user of the system is someone that has an user account and a password associated with it;
Depending on that user profile he has permission to do different functionalities.
There are the following types of user profiles:
- Restricted User, can check certain informations in the RIOS software (SVARH)
- System Administrator, responsible for managing users, managing stations, managing the data base"]

Term trm_res_user "Restricted user" : Noun [isA trm_user applicableTo Stakeholder, Actor description
"User that has visualization access to certain parts of the RIOS application (SVARH);
This user has permission to check data related to a specific station or a set of stations.
This user can have writing permissions, which means that they can save data in their computer.
The user can be notified in case of alert, if it is in the permission's list of that station."]

Term trm_sys_admin "System administrator" : Noun [isA trm_user applicableTo Actor, Stakeholder description
"User that belongs to APA's water resources team, that has the responsibility to:
- Manage users accounts (e.g. Adding new users, changing users' permissions in the system),
- Manage the stations (e.g. adding / eliminating stations, changing data's collection frequency),
- Update the terminology list,
- Manage the data base system (e.g. uploads data) and
- Validate the data from the stations.
This user also receives notification in case of an alert level is reached."]
```

Specification 1: Partial specification of the SVARH's glossary.

4.1.2. Stakeholders

Defining the stakeholders is an important step when developing systems since they can impact the system or/and be impacted by it. By allowing to associate the stakeholders with the business goals and the actors, the project can be understood in a more detailed way.

In both systems, there is a user (that has a username and a password to enter the system). This person can be a system administrator or a restricted user.

A system administrator has the same responsibilities in both systems, such as, to manage the stations, to manage the users and to manage the database. The restricted user in the SNIRH refers to the regional environmental authorities and on the SVARH represents the civil protection users, fire department users and regional authorities' users, who have limited access to the information. This means that civil protection and fire departments are stakeholders.

APA is the most important stakeholder since it owns the systems. CPPE and CHT are also stakeholders in both systems as a result of providing data to the SNIRH and SVARH. These last two entities have a person that is responsible to communicate with the systems, so it was necessary to create two new stakeholders, CHT's responsible and CPPE's responsible, as it can be seen in Specification 2 and Specification 3.

Some of the stations do not communicate directly with APA's central, so a maintenance company is required and consequently, a technician, that must go to the field to collect the data. Therefore, the technician and the maintenance company must be considered stakeholders in the SNIRH system, so they are featured in Specification 2. Since the SVARH's system only uses data collected in real-time, they are not considered stakeholders of this system.


```

//Persons
Stakeholder stk_user "User" : Person : Person_User [description "(See Glossary)"]
Stakeholder stk_sys_admin "System administrator" : Person : Person_Manager [isA stk_user partOf stk_APA description]
Stakeholder stk_p_viewer "Public viewer" : Person [description "(See Glossary)"]
Stakeholder stk_res_user "Restricted user" : Person [isA stk_user description "(See Glossary)"]
Stakeholder stk_tech "Maintenance technician" : Person : Person_Technical [partOf stk_MC description]
Stakeholder stk_user_REA "Regional Environmental Authorities user" : Person [isA stk_res_user partOf stk_REA description]
Stakeholder stk_resp_CPPE "Responsible CPPE" : Person [partOf stk_CPPE description]
Stakeholder stk_resp_CHT "Hydrologic confederation's responsible" : Person [partOf stk_CHT description]

//Systems
Stakeholder stk_SNIRH "SNIRH" : System [description "Sistema Nacional de Informação de Recursos Hídricos"]
Stakeholder stk_INAG32 "INAG 32" : System : System_External [description]
Stakeholder stk_INAG_SIF "INAG SIF" : System : System_External [description]
Stakeholder stk_GFiltro "GFiltro" : System : System_External [description "Application that upload the data to the server."]
Stakeholder stk_Geolog "Geolog" : System : System_External [description]

//Others
Stakeholder stk_APA "Agência Portuguesa do Ambiente" : Organization : Organization_Owner [description]
Stakeholder stk_REA "Regional Environmental Authorities" : OrganizationalUnit [description]
Stakeholder stk_CPPE "CPPE" : Organization : Organization_Partner [description "Provides data to the system."]
Stakeholder stk_CHT "Confederação Hidrológica do Tejo - Tejo's hydrologic confederation" : Organization [description]
Stakeholder stk_MC "Maintenance company" : Organization [description "Company that does the maintenance of the stations."]

```

Specification 2: SNIRH stakeholders in RSL.

For the systems to function properly they require access to some external applications, such as INAG 32, INAG SIF (only used in SNIRH), GFiltro and Geolog. In Specification 3 can be seen that Rios is an internal system of SVARH.

```

//Persons
Stakeholder stk_user "User" : Person : Person_User [description "(See Glossary)"]
Stakeholder stk_sys_admin "System administrator" : Person : Person_Manager [isA stk_user partOf stk_APA description]
Stakeholder stk_res_user "Restricted user" : Person [isA stk_user description "(See Glossary)"]
Stakeholder stk_user_CP "Civil Protection user" : Person [isA stk_res_user partOf stk_CP description]
Stakeholder stk_user_FD "Fire Department user" : Person [isA stk_res_user partOf stk_FD description]
Stakeholder stk_user_REA "Regional Environmental Authorities user" : Person [isA stk_res_user partOf stk_REA description]
Stakeholder stk_resp_CPPE "CPPE responsible" : Person [partOf stk_CPPE description]
Stakeholder stk_resp_CHT "Hydrologic confederation responsible" : Person [partOf stk_CHT description]

//Systems
Stakeholder stk_SVARH "SVARH" : System [description "Sistema de Vigilância de Recursos Hídricos - SNIRH's sub-system"]
Stakeholder stk_INAG32 "INAG 32" : System : System_External [description]
Stakeholder stk_GFiltro "GFiltro" : System : System_External [description]
Stakeholder stk_Geolog "Geolog" : System : System_External [description]
Stakeholder stk_Rios "Rios" : System : System_Internal [partOf stk_SVARH description]

//Others
Stakeholder stk_APA "Agência Portuguesa do Ambiente" : Organization : Organization_Owner [description]
Stakeholder stk_CP "Civil Protection" : Organization [description]
Stakeholder stk_FD "Fire Department" : Organization [description]
Stakeholder stk_REA "Regional Environmental Authorities" : OrganizationalUnit [description]
Stakeholder stk_CPPE "CPPE" : Organization : Organization_Partner [description "Provides data to the system."]
Stakeholder stk_CHT "Confederação Hidrológica do Tejo - Tejo's hydrologic confederation" : Organization [description]

```

Specification 3: SVARH stakeholders in RSL.

4.1.3. Actors

In this Chapter, 4.1.3, the actors of the related systems are presented using RSL since this language allows defining the actors in detail by stating the type, the stakeholder associated (if there is any) and if it is linked to another actor (by using isA).

Specification 4 presents all the actors involved in the SNIRH system and the corresponding stakeholders' names. Each actor is responsible for either initiating or participating in a use case. The system administrator is considered a user since it has an account on the system.

```

Actor aU_user "User" : User [stakeholder SNIRH_Business.stk_user]
Actor aU_SysAdmin "System administrator" : User [isA aU_user stakeholder SNIRH_Business.stk_sys_admin]
Actor aU_Tech "Maintenance technician" : User [stakeholder SNIRH_Business.stk_tech]
Actor aU_p_viewer "Public viewer" : User [stakeholder SNIRH_Business.stk_p_viewer]
Actor aU_resp_CPPE "Responsible CPPE" : User [stakeholder SNIRH_Business.stk_resp_CPPE]
Actor aU_resp_CHT "Responsible CHT" : User [stakeholder SNIRH_Business.stk_resp_CHT]

Actor aS_INAG32 "INAG 32" : ExternalSystem [stakeholder SNIRH_Business.stk_INAG32]
Actor aS_INAG_SIF "INAG SIF" : ExternalSystem [stakeholder SNIRH_Business.stk_INAG_SIF]
Actor aS_GFiltro "GFiltro" : ExternalSystem [stakeholder SNIRH_Business.stk_GFiltro]
Actor aS_Geolog "Geolog" : ExternalSystem [stakeholder SNIRH_Business.stk_Geolog]

```

Specification 4: SNIRH actors in RSL.

The sub-system datalogger has five actors, two of the users (system administrator and maintenance technician) two external systems (GFiltro and Geolog software) and a timer that it is responsible for sending data periodically to APA.

By analysing the Specification 5 it can be seen that the system administrator and the restricted user are both considered users. The users from CPPE, CHT and civil protection were defined as restricted users. Rios software was defined as a user since it is a major part of the SVARH system. Two timers were specified, one for when the alarm level is reached in a station and other for when the data from the stations is received.

```

Actor aU_SysAdmin "System administrator" : User [isA aU_user stakeholder SVARH_Business.stk_sys_admin]
Actor aU_resp_CPPE "Responsible CPPE" : User [isA aU_res_user stakeholder SVARH_Business.stk_resp_CPPE]
Actor aU_resp_CHT "Responsible CHT" : User [isA aU_res_user stakeholder SVARH_Business.stk_resp_CHT]
Actor aU_user_CP "User Civil Protection" : User [isA aU_res_user stakeholder SVARH_Business.stk_user_CP]
Actor aU_user "User" : User [stakeholder SVARH_Business.stk_user]
Actor aU_res_user "Restricted user" : User [isA aU_user stakeholder SVARH_Business.stk_res_user]

Actor aS_INAG32 "INAG32" : ExternalSystem [stakeholder SVARH_Business.stk_INAG32]
Actor aS_Geolog "Geolog" : ExternalSystem [stakeholder SVARH_Business.stk_Geolog]
Actor aS_GFiltro "GFiltro" : ExternalSystem [stakeholder SVARH_Business.stk_GFiltro]
Actor aS_Rios "Rios" : User [stakeholder SVARH_Business.stk_Rios]

Actor aT_AlertLevel "Alert level reached" : Timer [description "Alert level reached in a certain station."]
Actor aT_DataReceived "Data received" : Timer [description "Data from the stations is received."]

```

Specification 5: SVARH actors in RSL.

Note that not all the stakeholders are actors and not all actors have a stakeholder associated (e.g. aT_AlertLevel).

4.1.4. Data Entities

In this section, the data entities of the SNIRH and SVARH will be presented either using RSL or UML class diagrams. Due to space constraints, only the most important specifications are shown in Chapter 4.1.4 and the others are presented in the annexes.

To better comprehend the SNIRH's data entities four UML class diagrams were made (Basin, Location, Station, User), instead of just one with all the entities. The same procedure was applied to SVARH, dividing it into five packages, Station, Location, Alarm, User and Basin.

Specification 6 shows the DataEntities in the SNIRH's basin package, the aquatic environment data entity allows characterizing rivers, by stating some of its attributes, such as the name, the name of its basin and the length of it. A station is associated with a specific aquatic environment.

```

DataEntity e_AquaticEnv "Aquatic Environment" : Document [
  attribute Name "Name" : String [isNotNull isUnique]
  attribute drain_area "Drained area" : Decimal (2) [description "measured in km2"]
  attribute basin_name "Basin name" : DataEnumeration de_BasinName [isNotNull]
  attribute perimeter "Basin Perimeter" : Decimal [isNotNull]
  attribute area "Basin area" : Decimal [isNotNull]
  attribute isMain_River "is Main River" : Boolean
  attribute isAffluent_River "is Affluent River" : Boolean
  attribute type "Environment type" : String [isNotNull description "e.g. river, lake, lagoon, stream"]
  attribute total_length "Total length" : Decimal [isNotNull]
  primaryKey (Name)]

DataEntity e_Station "Station" : Master : Main []

DataEnumeration de_Status "Status" : Simple [values "Active", "Extinct", "Inactive", "Suspended"]

DataEnumeration de_NetworkType "Network type" : Simple [values "Hydrometric", "Meteorological"]

DataEnumeration de_BasinName "Basin names" : Simple []

```

Specification 6: RSL specification of SNIRH DataEntities: Aquatic Environment.

As seen in Figure 17, the SVARH's basin package is structured differently compared with the one in the SNIRH system. In this case, the basin data entity is composed of a group of stations and it is linked to one main river and many effluents.

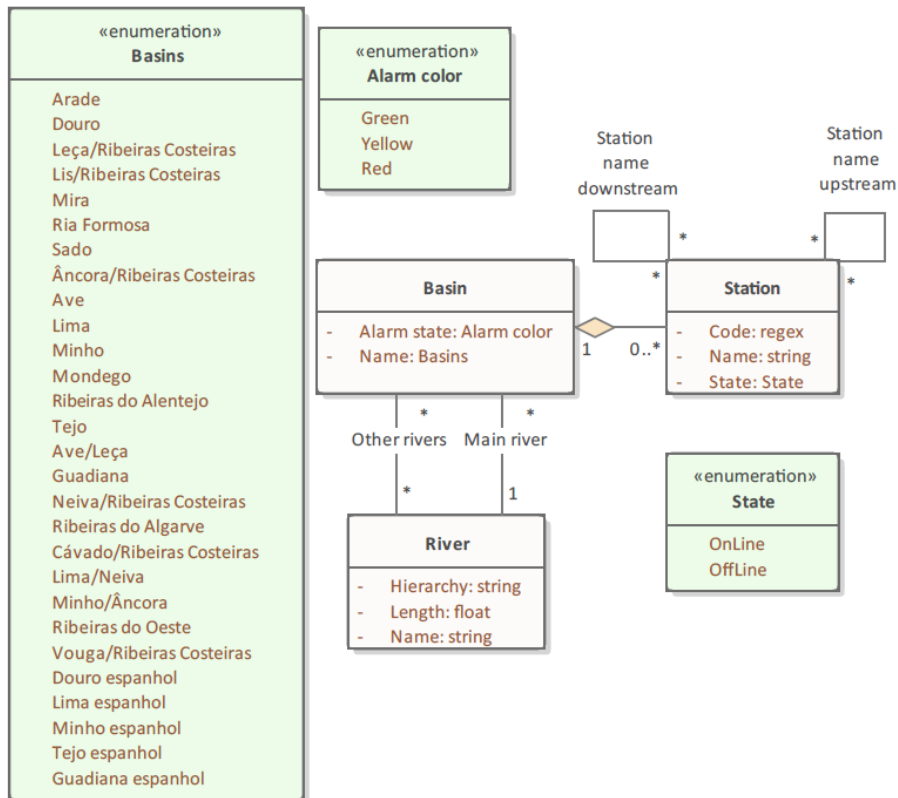


Figure 17: UML diagram of SVARH DataEntities: Basin package.

The data entity station in SVARH states the name, code and the actual state of the station if it is online or offline. Figure 18 shows a class diagram of the main data entities associated with the station concept, such as sensor, data log and station type. According to the diagram, a station is composed of many sensors that can record several data logs. The attributes of the data logs vary according to the type of station except for the date, time and the total number of values presented. For example, a data log from meteorological station presents data regarding the precipitation and the wind velocity.

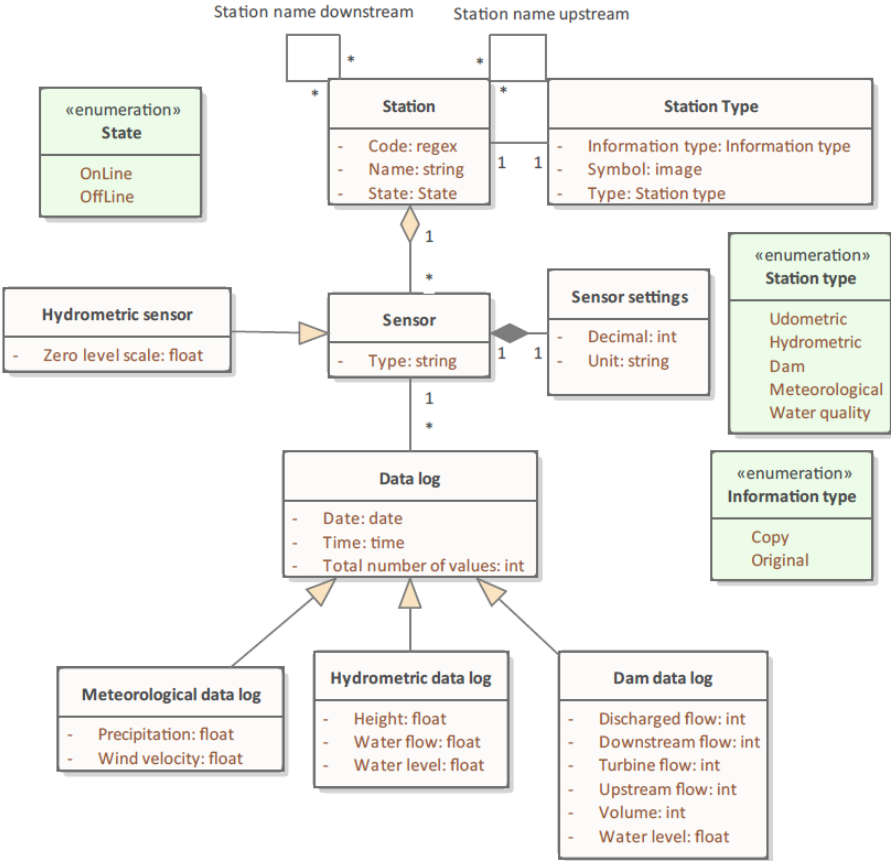


Figure 18: UML diagram of SVARH DataEntities: Station package.

Figure 19 displays the schemas of a dam and a hydrometric station in the Rios application, which support some of the data entities and attributes presented in Figure 18. Namely, the attributes associated with the different types of stations and the names of the stations downstream and upstream.

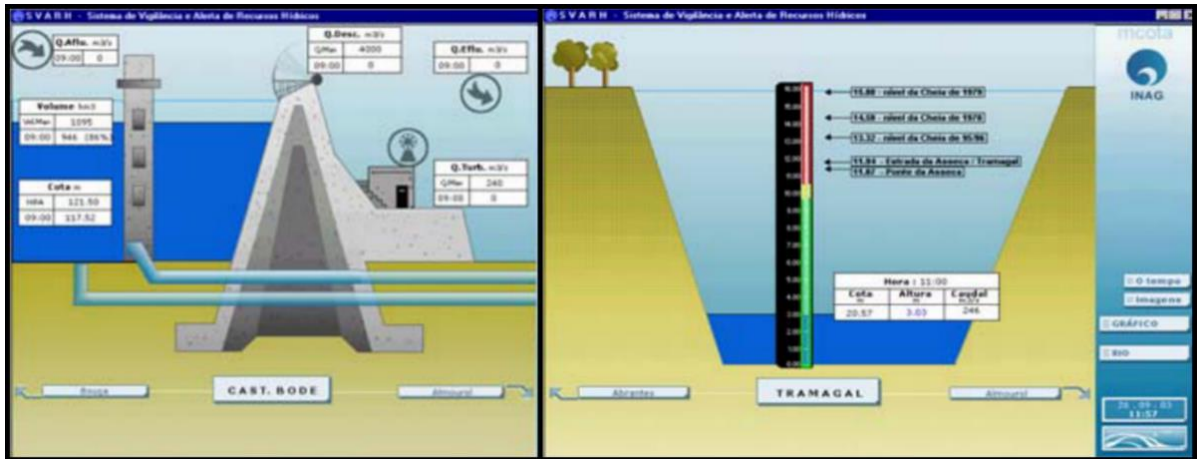


Figure 19: Example of a station in a dam and a Hydrometric station in Rios [24].

From Figure 20 is possible to understand that the data entity “Sensor” from SVARH can have many limit descriptions associated and that its attributes differ according to the type of station. Furthermore, it can also be seen that a sensor can only have three alerts, with three different colours, names and limits. The basins and the districts have an attribute that is the “Alarm state” since they present the alarm level of the stations assigned to each area. For instance, if the alarm of the station Ponte de Águeda is red, the Vouga / Ribeiras Costeiras basin alarm is also red. Figure 21 corroborates the alarm definition and the limit description data on the diagram elaborated in Figure 20.

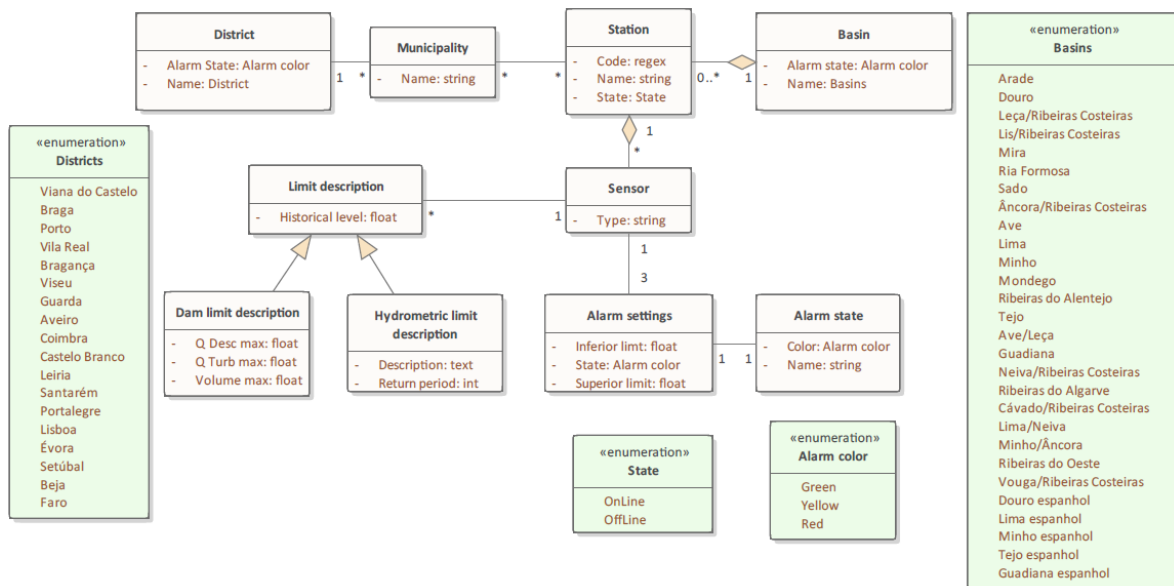


Figure 20: UML diagram of SVARH DataEntities: Alarm package.

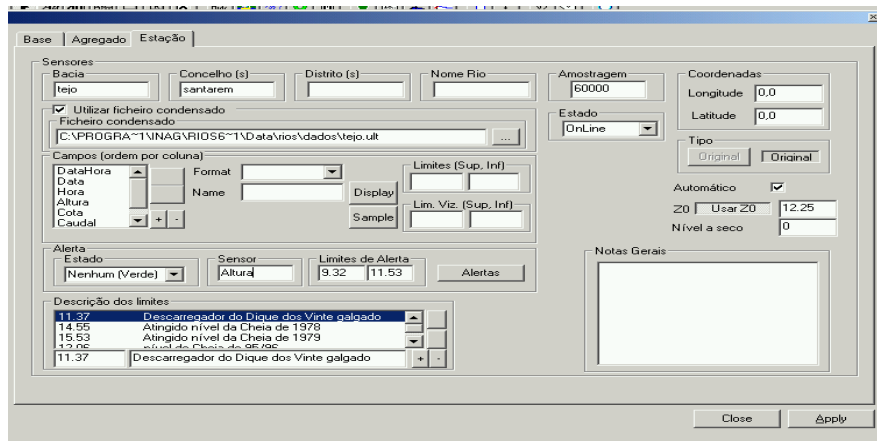


Figure 21: Station definition in Rios editor.

As seen in Figure 22, SNIRH station data entity has five attributes, name, code, network type, status and if it has teletransmission. The location of the stations defined specifies the XY coordinates, latitude, longitude, parish, municipality and district. One location has only one station, but a parish can have many stations.

The description of the location of a station in SVARH presents fewer details compared with SNIRH since it only states the municipality and the district that it is located (Figure 20).

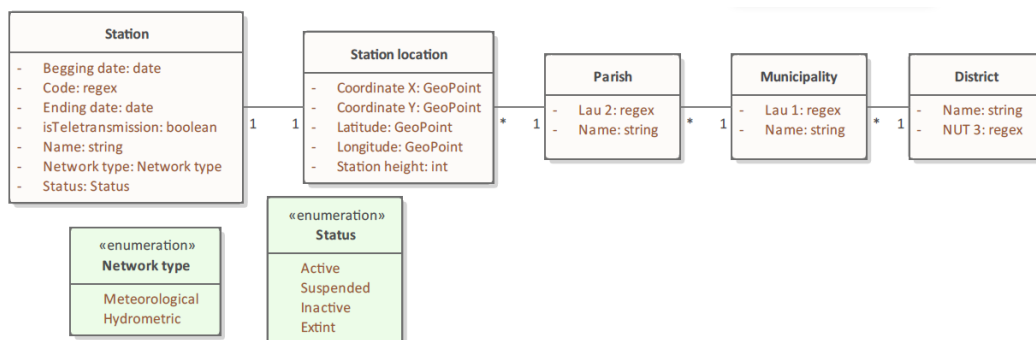


Figure 22: UML diagram of SNIRH's DataEntities: Location package.

A station has an entity responsible for it that is linked to several users, as it can be observed in Figure 23. The “User” data entity states the name, contacts, password, username and if it is an administrator or not.

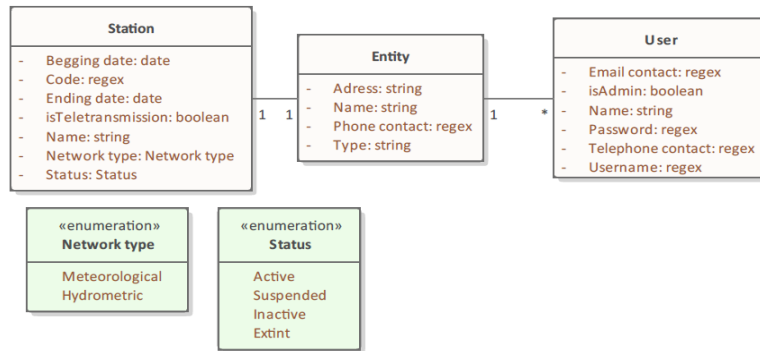


Figure 23: UML diagram of SNIRH's DataEntities: User package.

The SVARH data entities related to the user are present in Specification 7, such as the group, which allows characterizing groups of users, like a group for the Tejo basin or the Alentejo region, and the user settings, which allows defining certain attributes of the system, namely the view of the initial page. By analysing the specification, it can be concluded that the user settings have two data entities related to it, the visualization mode and the connection mode since it specifies that “isA e_UserSet” which indicates that a generalization relationship is used. The data that the user can access and the ability to edit data are also known as the users’ permissions and are defined according to the basins or districts in the data entity “User”.

```

DataEntity e_User "User" : Master [
  attribute political_permi "Political permissions" : DataEnumeration de_Districts [isNotNull]
  attribute tech_permi "Technical permissions" : DataEnumeration de_Basins [isNotNull]
  attribute isWrite_permi "Writing permission" : Boolean [isNotNull]
  attribute username "Username" : String [isNotNull isUnique]
  attribute Description "Description" : String [isNotNull description "e.g. Intern Marta Gonzalez"]
  attribute password "Password" : Regex [isNotNull isEncrypted]
  attribute phone "Phone number" : Regex [isUnique]
  attribute email "Email" : Regex [isUnique]
  attribute fax "Fax number" : Regex [isUnique]
  attribute group_name "Group name" : String [description "Entity which the user belongs"]
  primaryKey (Description)
  foreignKey e_Group (group_name)]

DataEntity e_Group "Group" : Reference []

DataEntity e_UserSet "User settings" : Document [
  attribute cache "Cache size" : Integer [isNotNull]
  attribute update_time "Data update time" : Integer [isNotNull]
  attribute storage_time "Disc data storage time" : Integer [isNotNull]
  attribute ini_page "Intial page" : String [isNotNull]
  attribute h_graph "Hours shown on the graphics" : Integer [isNotNull]
  attribute scroll "Scroll velocity" : Integer [isNotNull]
  attribute t_backup "Time between backup copies" : Integer [isNotNull]
  attribute Description "Description" : String [isNotNull description "e.g. Intern Marta Gonzalez"]
  primaryKey (ini_page, Description)
  foreignKey e_User (Description)]

DataEntity e_VisMode "Visualization mode" : Document : Detail [
  isA e_UserSet
  attribute filter "Filter stations" : DataEnumeration de_StationType [isNotNull]
  attribute hydro "Present levels instead of heights" : Boolean
  attribute dam "Present volumes instead of filling %" : Boolean]

DataEntity e_ConnectMode "Connection mode" : Document : Detail [
  isA e_UserSet
  attribute isAuto "is Automatic" : Boolean
  attribute isDataTrans "is Data transfered" : Boolean
  attribute isManual "is Manual" : Boolean
  attribute type "Connection type" : DataEnumeration de_ConnectionType [isNotNull]
  attribute site "Update site" : Regex]

DataEnumeration de_ConnectionType "Connection type" : Simple [values "HTTP", "FTP"]
  
```

Specification 7: RSL specification of SVARH DataEntities: User.

Figure 24 shows some of the settings that a user can define in its account, as the view of the initial page, the interval of time between backup copies and the types of stations that the user wishes to view. These settings are specified in RSL in the Specification 7 in the data entities “e_UserSet”, “e_VisMode” and “e_ConnectMode”.

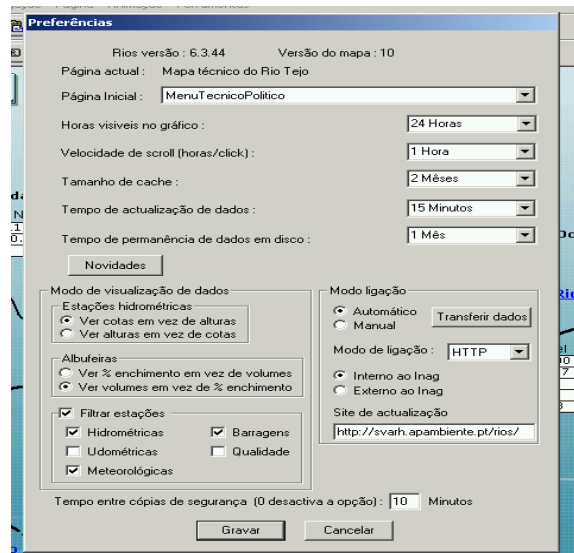


Figure 24: User preferences in Rios.

From Specification 8 is possible to observe the DataEntityClusters of SNIRH, namely the “ec_station_simple” and the “ec_station_complex”, in which the only difference is that the complex has the “e_S_Location” specified as a detail.

```
DataEntityCluster ec_station_simple "Station (simple)" : Complex[
  master e_Station
  detail e_Sensor
  detail e_Parameter [reference e_Data, e_P_Settings]
  detail e_S_Type]
DataEntityCluster ec_station_complex "Station (complex)" : Complex[
  master e_Station
  detail e_Sensor
  detail e_Parameter [reference e_Data, e_P_Settings]
  detail e_S_Location [reference e_Parish, e_Municipality, e_District]
  detail e_S_Type]
DataEntityCluster ec_environment "Environment" : Simple [master e_Station reference e_AquaticEnv]
DataEntityCluster ec_station_local "Station location" : Complex []
DataEntityCluster ec_user "User" : Complex []
```

Specification 8: RSL specification of SNIRH DataEntityClusters.

The DataEntityCluster “ec_Alarm” and “ec_Station_Sensor” of the SVARH system, represented in Specification 9 in RSL, are illustrated in a UML class diagram in Figure 20 and Figure 18.

```
DataEntityCluster ec_Basin "Basin" : Simple [master e_Station detail e_Basin [reference e_River]]
DataEntityCluster ec_Station_Sensor "Station (sensor)" : Complex [
  master e_Station
  detail e_Sensor [reference e_Sensor_S]
  detail e_Data
  reference e_StationType]
DataEntityCluster ec_User "User" : Complex []
DataEntityCluster ec_Station_Location "Station location" : Simple
[master e_Station detail e_Municipality reference e_District]
DataEntityCluster ec_Alarm "Alarm" : Complex [
  master e_Station
  detail e_Basin
  detail e_Municipality [reference e_District]
  detail e_Sensor [reference e_LimitDesc, e_AlarmS, e_AlarmSet]]
```

Specification 9: RSL specification of SVARH DataEntityClusters.

4.1.5. Requirements

There are several requirements types, such as goals, functional requirement, constraint, quality requirement, use case and user story [5]. However, this thesis focuses only on the use case requirement type.

Due to space constraints, some of the specifications in RSL are presented in the annexes and the ones shown in Chapter 4.1.5 only a few are expanded.

Since part of the SNIRH system is open to the public, the public viewer can consult certain contents, such as documents, photos, graphs, reports and flow curves related to the station selected by the user. The public viewer can even print or export data from a station, as it can be observed in Specification 10. It is also publicly available on the SNIRH webpage the definitions of the terms used in it. The restricted user inherits all the use cases associated with the public viewer and the system administrator also acquires the ones associated with the restricted user.

```
UseCase uc_10_ReadReports "Read station reports" : EntitiesBrowse []
UseCase uc_11_ConsultPhotos "Consult photos of the stations" : EntityRead []
UseCase uc_12_ReadDocs "Read documents" : EntitiesBrowse []
UseCase uc_13_ConsultGraphs "Consult graphs" : EntitiesBrowse []
UseCase uc_14_ConsultStats "Consult statistical data" : EntitiesBrowse [
  actorInitiates aU_p_viewer
  dataEntity ec_station_complex
  actions aRead, aPrint, aExport
  description "Consult statistical data regarding a specific parameter from several stations"]
UseCase uc_15_ConsultFlowCurve "Consult flow curve" : EntitiesBrowse []
UseCase uc_16_ReadDefinitions "Read terms' definitions" : EntityRead []
UseCase uc_17_ViewData "View station data" : EntitiesBrowse [
  actorInitiates aU_p_viewer
  dataEntity ec_station_simple
  actions aRead, aPrint, aExport
  description "Read the data from the stations selected"]
```

Specification 10: RSL specification of SNIRH UseCases: Public viewer.

However, for all the information formerly stated to become available on the webpage, several actions must be executed previously by the System Administrator. To acquire data from a station with teletransmission the system administrator must inquire the stations, otherwise, the data must be obtained from the maintenance server. In Figure 25 it is possible to observe all the SNIRH use cases related to the data recorded by the stations. The use case "Process data" includes three other use cases, "Validate data", "Convert data", and "Upload data to the database". Besides, the actor that initiates these use cases, aU_SysAdmin, the last two referred use cases also have actors that participate on it, the INAG 32, INAG SIF and GFiltro, which are the external systems used to fulfil the task.

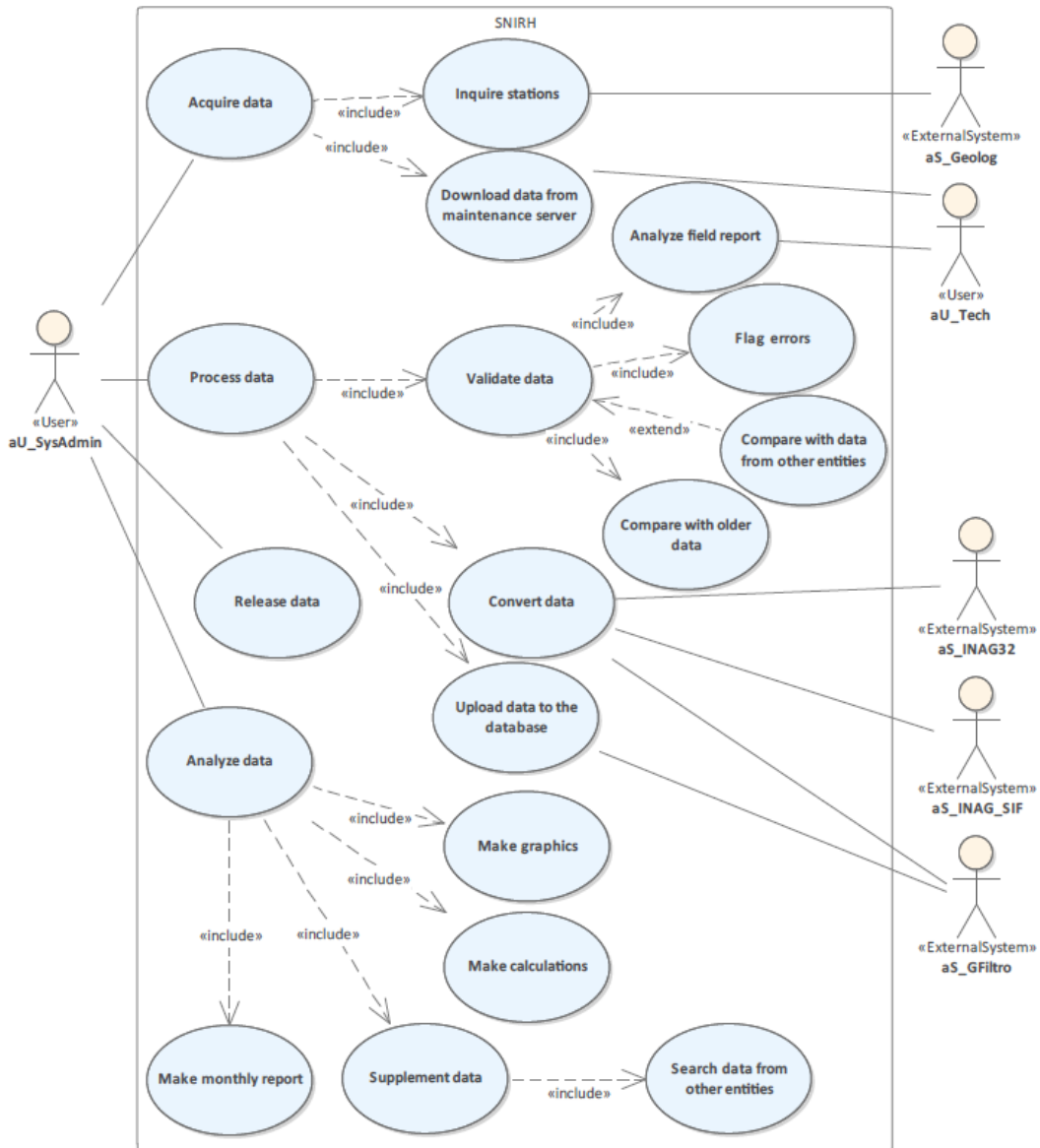


Figure 25: UML diagram of SNIRH UseCases: Data package.

From the Specification 11 is possible to perceive that the four main use cases present in SVARH are the same as the ones in the SNIRH system (Figure 25), “Acquire data”, “Process data”, “Release data” and “Analyse data”. Despite that, the SVARH system has much fewer use cases compared with SNIRH, because it does not download data from the maintenance server, make monthly reports, validate or supplement data. The use case “make automatic calculus” is trigger by the actor aT_DataReceived, which is a timer, and it is executed by Rios software.

```

UseCase uc_7_AcquireData "Acquire data" : EntitiesInteropServiceInvocation []
UseCase uc_7_1_InquireTele "Inquire stations" : EntitiesInteropServiceInvocation []
UseCase uc_8_ProcessData "Process data" : EntityCreate []
UseCase uc_8_1_ConvData "Convert data" : EntityCreate []
UseCase uc_8_2_UploadDB "Upload data in Database" : EntityCreate []
UseCase uc_9_ReleaseData "Release data" : EntitiesOther [
  actorInitiates aU_SysAdmin
  actorParticipates aS_Rios
  dataEntity ec_Station_Sensor
  actions aUpload
  description "Release the data in Rios"]
UseCase uc_10_AnalyzeData "Analyze data" : EntitiesDashboard []
UseCase uc_10_1_AutoCalculus "Make automatic calculus" : EntitiesDashboard [
  actorInitiates aT_DataReceived
  actorParticipates aS_Rios
  dataEntity ec_Station_Sensor
  precondition "Data must be in collected and transformed first"
  actions aCreate
  extends uc_10_AnalyzeData onExtensionPoint EPAutoCalculus
  description "The system makes automatically calculus based on the data collected"]
UseCase uc_10_2_MakeGraphics "Make graphics" : EntityCreate []

```

Specification 11: RSL specification of SVARH UseCases: Data.

The Rios software allows the user to make graphics based on the acquired data from the stations and the interval of time selected, as it can be seen in Figure 26, which means that is a use case initiated by the system administrator (Specification 11).

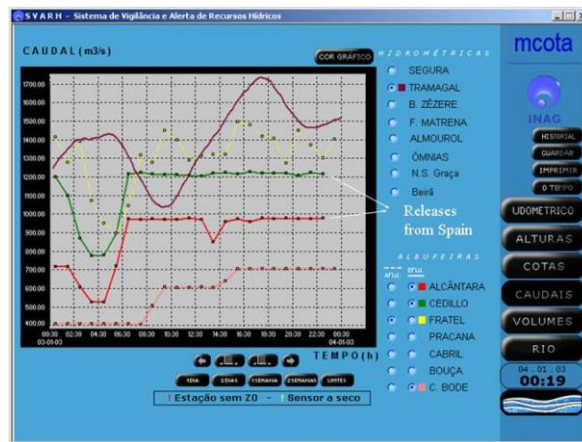


Figure 26: Example of a graphic in Rios [69].

Figure 27 illustrates the SVARH use cases regarding management tasks, in which the system administrator is considered an actor involved. The “manage communication with the civil protection” use case employs the include relation to indicate that help to interpret data must be provided to the civil protection responsible. As previously stated, APA uses a script to import information from CPPE and CHT, and sometimes it may require an update in order to function properly. So, an extended relation was used to connect the use case “Edit script” with the “Manage communication with CPPE” and the “Manage communication with CHT”. The management of the users is executed by the system administrator, which includes adding, deleting, editing the user accounts, namely, the users’ permissions. The Rios software is used by the system administrator to manage the stations, which includes the management of the alarms and it can include the definition of the parameters and

historical limits. Extension point “Send alarms” is initiated by the actor “aT_AlertLevel”, which is a timer set to get triggered when an alarm level is reached, and it also has the participation of aS_Geolog, which is the system responsible for the process of sending the alarm notification.

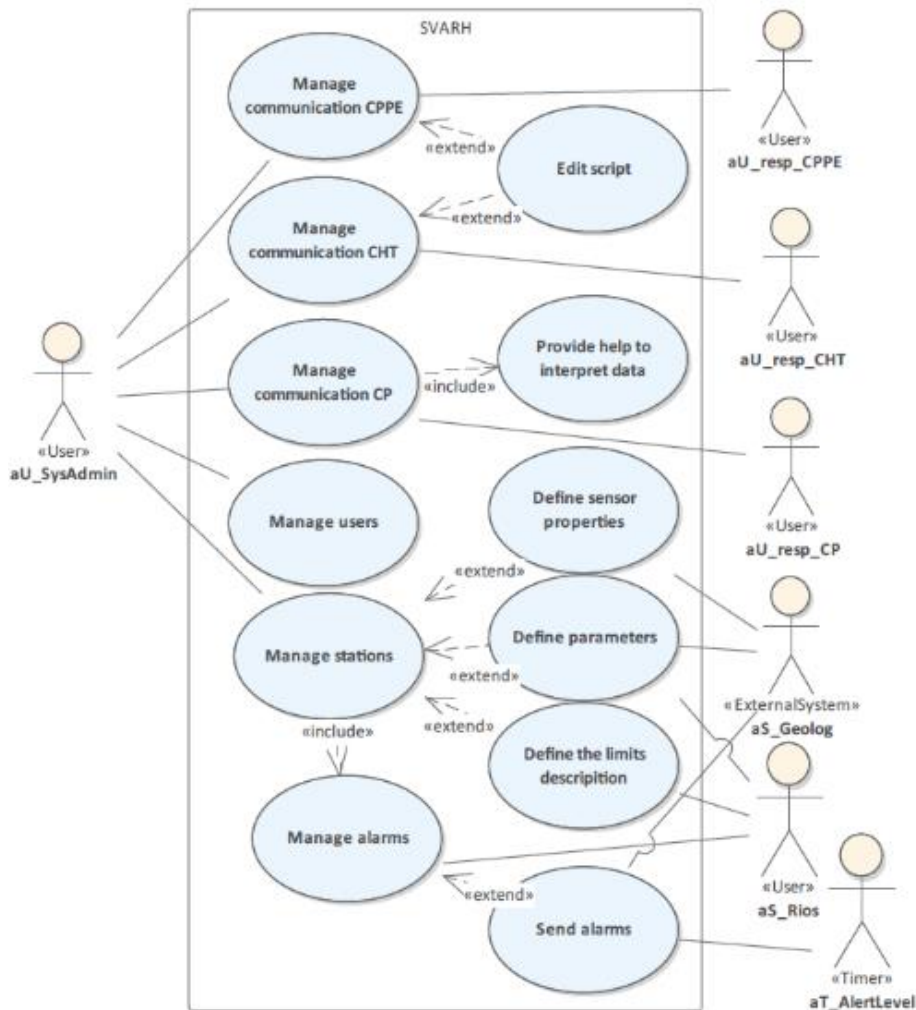


Figure 27: UML diagram of the SVARH UseCases: Manage package.

SNIRH use cases regarding the management of the system are quite similar to the SVARH use cases since it also manages the communication with the partners, users and the stations. To manage the stations more efficiently, they are divided into groups and assigned to an APA technician. Unlike the SVARH, the management of the stations does not include the management of the alarms, since they are not included in this system.

4.2. HiSTAV

For this work, it was assumed that the pre-processor is a sub-system of HiSTAV and that it is considered an internal system of the simulation tool.

The glossary, stakeholders, actors, data entities and requirements were written using RSL language to assure that the system would be rigorously defined. The data entities and requirements were also written in UML, so that a domain expert could understand the system more quickly.

4.2.1. Glossary

The HiSTAV uses terms related to a certain subject, such as simulations and GIS. Specification 12 presents some of the HiSTAV terms and its definitions, which are written in RSL using the ITLingo-Studio.

```
Term trm_mesh "Mesh" : Noun [applicableTo DataEntity description]
Term trm_node "Node" : Noun [applicableTo DataEntity synonym "Vertex" description "XY(Z) points (in the layer's CRS)."]
Term trm_edge "Edge" : Noun [applicableTo DataEntity synonym "Arrest" description "Element that connects pairs of vertices."]
Term trm_face "Face" : Noun [applicableTo DataEntity synonym "Cell" description]

Term trm_raster_data "Raster data" : Noun [applicableTo DataEntity description]
Term trm_DTM "Digital Terrain Model" : Noun [acronym "DTM" applicableTo DataEntity description
"A DTM is a mathematical representation (model) of the ground surface, most often in the form of a regular grid,
in which a unique elevation value is assigned to each pixel. DTMs represent the bare ground of the terrain."]

Term trm_vector_data "Vector data" : Noun [applicableTo DataEntity description]

Term trm_layer "Layer" : Noun [applicableTo DataEntity description
"The mechanism used to display geographic datasets.
Each layer references a dataset and specifies how that dataset is portrayed using symbols and text labels.
When you add a layer to a map, you specify its dataset and set its map symbols and labeling properties."]

Term trm_inlet "Inlet boundary" : Noun [applicableTo DataEntity description "Inlet boundaries specifies where the fluid enters."]
Term trm_outlet "Outlet boundary" : Noun [applicableTo DataEntity description "Outlet boundaries specifies where the fluid leaves."]

Term trm_start_time "Start time value" : Noun [applicableTo DataEntity description "The starting time of your simulation."]
Term trm_end_time "End time value" : Noun [applicableTo DataEntity description "The end time of your simulation (in simulation time)."]

Term trm_time_step "Time step" : Noun [applicableTo DataEntity description
"Time step calculation type defines how time increments should be computed in case of an adaptation event.
The time between two computation steps of a simulation (in simulation time)."]
Term trm_write_interval "Write interval" : Noun [applicableTo DataEntity description
"Write interval controls how often the results' values should be written. Time steps can be used to define these intervals."]
Term trm_runtime "Maximum runtime" : Noun [applicableTo DataEntity description
"Restricts the computation time (real time) to a chosen value.
If the calculation is still running at that point of time, it is forced to terminate."]
```

Specification 12: Partial specification of the HiSTAV glossary.

In this tool, a user was defined as someone that has access to the system and can run a simulation. There is a special type of user, the system administrator, which is responsible for updating the executable file of the simulation tool.

4.2.2. Stakeholders

This system was developed by IST members and sponsored by the Fundação para a Ciência e Tecnologia (FCT). The Instituto Português do Mar e da Atmosfera (IPMA) and the APA are the two main partners, they provide most of the data for the system. So, all these entities are key stakeholders. In order to have a good flow of communication between the system and the partners, an individual at each partner was selected to facilitate it (Specification 13).

For the simulator model to work the data must be prepared properly, to achieve that the Pre-processor should be run previously. To define certain data, such as the inlet and outlet cells, the user may use QGIS to visualize it. ParaView is the software used to visualize the results obtained from the HiSTAV. These are the three systems involved in the HiSTAV, making them important stakeholders.

```

//Persons
Stakeholder stk_user "User" : Person : Person_User [description "(See Glossary)"]
Stakeholder stk_sys_admin "System Administrator" : Person : Person_User [isA stk_user description "(See Glossary)"]
Stakeholder stk_resp_APA "APA Responsible" : Person [partOf stk_APA description ]
Stakeholder stk_resp_IPMA "IPMA Responsible" : Person [partOf stk_IPMA description ]

//Systems
Stakeholder stk_HiStav "HiStav" : System [description "(See Glossary)"]
Stakeholder stk_Pre_processor "Pre processor" : System : System_Internal [description "(See Glossary)"]
Stakeholder stk_QGIS "QGIS" : System : System_External [description "(See Glossary)"]
Stakeholder stk_ParaView "ParaView" : System : System_External [description "(See Glossary)"]

//Others
Stakeholder stk_IST "Instituto Superior Técnico" : Organization : Organization_Owner [description ]
Stakeholder stk_FCT "Fundação para a Ciência e Tecnologia" : Organization : Organization_Sponsor
Stakeholder stk_APA "Agência Portuguesa do Ambiente" : Organization : Organization_Partener [description ]
Stakeholder stk_IPMA "Instituto Português do Mar e da Atmosfera" : Organization : Organization_Partener [description ]

```

Specification 13: HiSTAV stakeholders in RSL.

4.2.3. Actors

In the HiSTAV system, it is only defined one user since the system administrator it is also considered a user. The difference between the actors of system HiSTAV (Specification 14) and the Pre-processor is that one requires the external system ParaView and the other the QGIS, respectively.

```

Actor aU_User "User" : User [stakeholder HiStav_Business.stk_user]
Actor aS_ParaView "ParaView" : ExternalSystem [stakeholder HiStav_Business.stk_ParaView]

```

Specification 14: HiSTAV actors in RSL.

4.2.4. Data Entities

HiSTAV data entities will be presented either using RSL or UML class diagrams. Due to space constraints, only the most important specifications are shown in this Chapter and the others are presented in the annexes.

HiSTAV pre-processor uses several different files as an input to run, the types of files (e.g. Vector, Raster) and their subject-matters vary according to the available information. For instance, in some cases, it is possible to obtain a digital surface model (DSM) of the area and on others, the only option accessible is a DTM with low resolution.

For this specification, it was assumed that the user was able to obtain files stating the buildings and obstacles in vector format, the borders in vector format, a DTM in raster format and hydrometric timeseries. Therefore, the DataEntities defined as inputs of the pre-processor sub-system are the buildings and obstacles, borders, domain, DTM and boundary conditions, which are associated with the hydrometric timeseries.

Figure 28 shows a UML diagram with the DataEntities of the outputs of the pre-processor, which allows understanding that the data entities initial condition and boundary condition are associated with the data entity mesh. Each boundary condition is associated with a specific node of the mesh and to a specific hydrometric timeseries. An initial condition value is linked to a specific element of the mesh, so both data entities are associated. Note that a mesh must divulge the coordinated reference system (CRS) used since its nodes have XYZ coordinates.

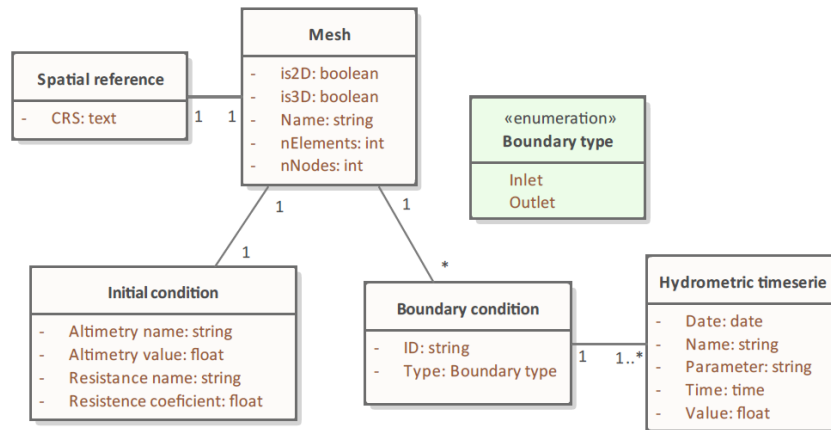


Figure 28: UML diagram of the pre-processor outputs DataEntities.

In this case, the DataEntities defined as inputs for this simulation were the ones presented in Specification 15 (“e_DistrCond”, “e_PhysicalConst”, “e_Time”, “e_Numerical”) and Figure 28 (Mesh, Initial condition, Boundary condition, Hydrometric timeseries and Spatial reference). The physical constants employed in this system are divided into two data entities since “e_PhysicalConst” presents the general constants that can be applied in every model and “e_SpecPhysical” shows physical constants specific for this case, which means that its attributes can change according to the simulation that is being executed.

```

DataEntity e_DistrCond "Distributed conditions" : Master []

DataEntity e_PhysicalConst "Physical constants" : Reference [
  attribute FriCoef "Friction coefficient" : Decimal
  attribute g "Gravitation acceleration" : Decimal
  attribute w_density "Water density" : Decimal
  attribute VisCoef "Viscosity coefficient" : Decimal
  attribute Name "File name" : String [isNotNull isUnique]
  primaryKey (Name)]

DataEntity e_SpecPhysical "Specific physical constants" : Reference : Detail [
  isA e_PhysicalConst
  attribute SedDiameter "Sediments diameter" : Decimal
  attribute SedDiameterUnit "Sediments diameter unit" : String
  attribute lift "Lift coefficient" : Decimal
  attribute drag "Drag coefficient" : Decimal
  attribute LubriCoef "Lubricated friction coefficient" : Decimal]

DataEntity e_Time "Time information" : Reference [
  attribute end_time "End time" : Time
  attribute start_time "Start time" : Time
  attribute Max_runtime "Max runtime" : Integer
  attribute Max_runtime_unit "Max runtime unit" : String
  attribute timestep "Timestep" : Integer
  attribute Name "File name" : String [isNotNull isUnique]
  primaryKey (Name)]

DataEntity e_Numerical "Numerical information" : Reference [
  attribute CFL "CFL" : Integer
  attribute write_inter "Write interval" : Integer [isNotNull]
  attribute write_inter_unit "Write interval unit" : String
  attribute Name "File name" : String [isNotNull isUnique]
  primaryKey (Name)]
  
```

Specification 15: RSL specification of HiSTAV DataEntities: Inputs.

The DataEntities of the HiSTAV outputs follow the RSL specification shown in Specification 16, the data entity “Result matrix” presents the values that can be reached at each moment regarding a

certain parameter. Each result matrix has associated some metadata, such as the time that the solver took to simulate, the format of the file and the folder name.

```
DataEntity e_ResMeta "Results metadata" : Reference [  
  attribute dataset "Dataset" : String [isNotNull]  
  attribute format "File format" : String [isNotNull]  
  attribute Name "File name" : String [isNotNull isUnique]  
  attribute Folder "Folder name" : String [isNotNull isUnique]  
  attribute Solver "Solver time" : Decimal [isNotNull]  
  attribute Version "Version" : String [isNotNull]  
  attribute parameter "Parameter" : String [isNotNull]  
  attribute time "Time" : Time  
  primaryKey (Name)  
  foreignKey e_ResMatrix (parameter, time)]  
  
DataEntity e_ResMatrix "Result matrix" : Master [  
  attribute time "Time" : Time  
  attribute Value "Value" : Decimal  
  attribute parameter "Parameter" : String [isNotNull]  
  primaryKey (parameter, time)]
```

Specification 16: RSL specification of HiSTAV DataEntities: Output.

From the Specification 17 is possible to see that there is only one DataEntityCluster complex, the “ec_Input_mesh”, in the HiSTAV system.

```
DataEntityCluster ec_mesh_complex "Mesh (complex)" : Complex [  
  master e_Mesh  
  detail e_BoundaryCond [reference e_Timeserie]  
  detail e_InitialCond  
  reference e_Spatial]  
DataEntityCluster ec_DistrCond "Distributed condition" : Simple [master e_DistrCond reference e_Domain]  
DataEntityCluster ec_BoundCond "Boundary condition" : Simple [master e_BoundaryCond detail e_Timeserie]  
DataEntityCluster ec_Results "Results" : Simple [master e_ResMatrix detail e_ResMeta]
```

Specification 17: RSL specification of HiSTAV DataEntityClusters.

4.2.5. Requirements

The use case is the requirement type that this work will focus on and due to space constraints, only a few specifications in RSL are expanded.

From Specification 18 is possible to understand that one of the first steps that a user must take when performing a simulation is the definition of the domain and to execute this use case QGIS (an external system) must be used. The uc_2_DefineBoundary has two extension points, the “Define inlet cells” and the “Define outlet cells”, and both require the software QGIS, causing it to be considered a participant actor. In the HiSTAV sub-system, the pre-processor executor must be run in order to obtain the results (e.g. mesh), so this action is considered a use case.


```

UseCase uc_1_DefineDomain "Define domain" : EntityCreate []

UseCase uc_2_DefineBoundary "Define boundary conditions" : EntityCreate [
  actorInitiates aU_User
  actorParticipates aS_QGIS
  dataEntity ec_BoundCond
  actions aCreate
  extensionPoints EPDefineInlet, EPDefineOutlet
  description "Define boundary conditions of the area that is being studied."]

UseCase uc_2_1_DefineInlet "Define inlet cells" : EntityCreate []
UseCase uc_2_2_DefineOutlet "Define outlet cells" : EntityCreate []

UseCase uc_3_RunPreProcessor "Run Pre Processor" : EntitiesOther []

UseCase uc_4_ViewMesh "View mesh" : EntitiesMapInteract [
  actorInitiates aU_User
  actorParticipates aS_QGIS
  dataEntity ec_mesh_complex
  actions aView]

```

Specification 18: RSL specification of HiSTAV pre-processor UseCases.

To create a HiSTAV simulation the physical constants, numeric and time information must be defined, as can be seen in Figure 29. If data regarding distributed conditions is available, then it can be defined as an input in the simulation. The actor aS_ParaView is the external system responsible for reading the results matrix obtained, however for this to be developed the aU_User must run the software. For the action, "Visualize results", to occur the user requires the participation of the external system ParaView.

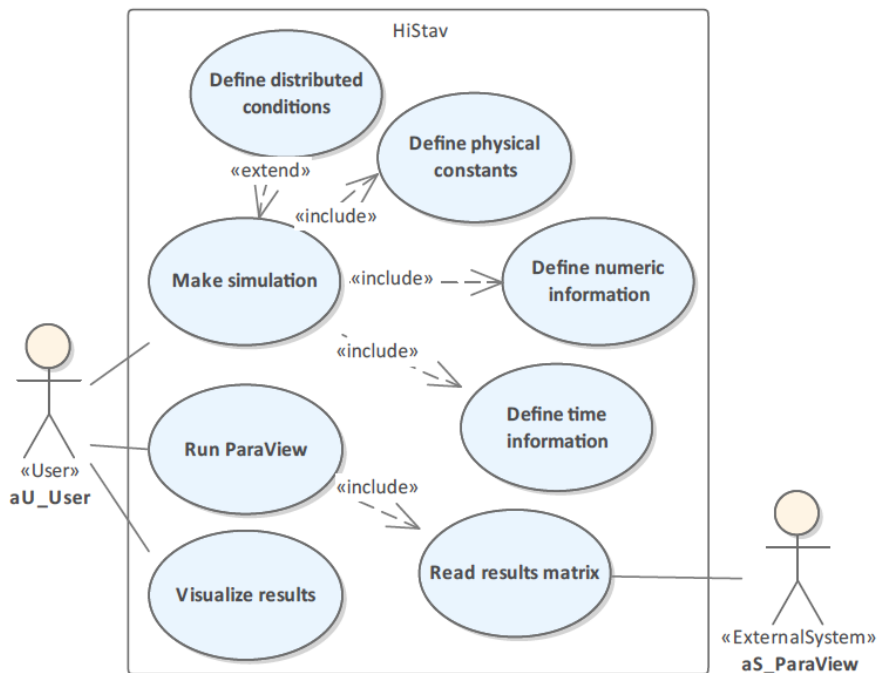


Figure 29: UML diagram of the HiSTAV UseCases.

5. RiverCure Portal

This Chapter describes the RiverCure portal (RCP) system, all its stakeholders, actors, data entities and requirements. The specifications presented in this Chapter are the result of the information and feedback collected during team meetings from several members. These meetings and the portal demonstration using the Águeda case study allowed to evaluate the specifications.

This system has four main data inputs from different sources: the weather radar information from the IPMA website; the data from SVARH; the photos uploaded by the users or collected from the social networks; and the simulations obtain from the HiSTAV software. Figure 30 shows the data has three stages of transformation, acquisition, processing and view, and that the process that each data pass-through depends on the source and the type of data. Data from the IPMA website and SVARH server are obtained using a script that may store the data in the RiverCure server, so it can become available on the portal. The process to obtain the simulations results depends on the inputs available. If the initial conditions, the mesh, and the boundary conditions are not defined then the data must be prepared using the QGIS and the Pre-processor. Otherwise, the files can be selected from the RiverCure server and the HiSTAV can be executed immediately. Note that data from SVARH, IPMA and photos can be inputs of the HiSTAV and are available in the RiverCure server. To view the outputs of the simulations the portal uses the ParaView¹⁵ software that reads the results files stored in the RiverCure server. Additionally, after acquiring and storing the photos, the Social Flood application gathers the photo metadata and the estimated water height, that are saved in the RiverCure server, so it can become available on the RCP.

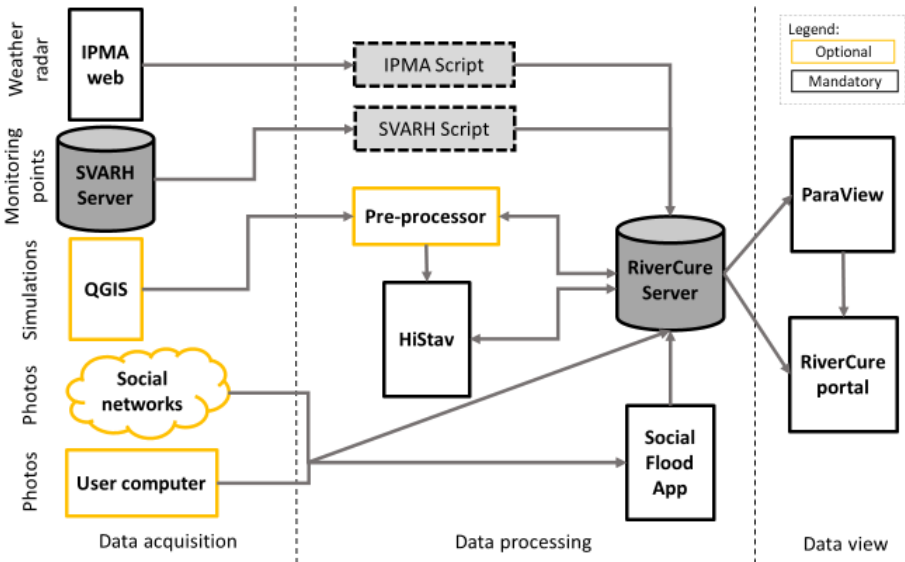


Figure 30: Data transformation schema.

¹⁵ <https://www.paraview.org/>

5.1. Glossary

The glossary of the RiverCure portal that is partially showed in Specification 19 was based on several standards such as:

- WaterML 2.0;
- ISO 19156:2011 (Geographic information - Observations and measurements);
- ISO 772:2011 (Hydrometry);
- ISO 16781:2013 (Space systems - Simulation requirements for control system);
- ISO 19101:2002 (Geographic information – Reference model).

By using these standards to define the terms used throughout the system it ensures a coherent system and makes it easier to understand the system. Consequently, it avoids possible errors due to miscommunication between stakeholders.

Term trm_hydrofeature "Hydrometric feature" : Noun [applicableTo DataEntity description "Feature of a type which denotes a physical structure intended to observe properties of a hydrologic feature."]

Term trm_hydro_net "Hydrometric network" : Noun [applicableTo DataEntity synonym "Hydrological network" description "Aggregate of hydrologically connected monitoring stations used for hydrologic observation of features."]

Term trm_hydrographicNet "Hydrographic network" : Noun [applicableTo DataEntity description "Aggregate of rivers and other permanent or temporary watercourses, and also lakes and reservoirs [WMO, 2016]."]

Term trm_property "Property" : Noun [applicableTo DataEntity synonym "Parameter, variable" description "Facet or attribute of an object referenced by a name."]

Term trm_monitoring_point "Monitoring point" : Noun [applicableTo DataEntity synonym "Station, Site, Locations" description "In WaterML2.0, in-situ monitoring points are described using the Sampling Features packages of the O&M model. Measurement and observation of natural phenomenon, which can be continuous or periodic, used several times for control and prevention."]

Term trm_gauging_station "Gauging station" : Noun [applicableTo DataEntity synonym "Hydrometric station" description "Site selected on a stream, river or open channel at which systematic measurements of water level or discharge, or both, are made."]

Term trm_sensor "Sensor" : Noun [applicableTo DataEntity description "Type of observation procedure that provides the estimated value of an observed property at its output."]

Specification 19: Partial specification of the RiverCure glossary in RSL.

Table 3 presents some terms that are used in the RiverCure portal, that have the same meaning as some applied in the WaterML 2.0, SNIRH and SVARH, but have a different word associated.

Table 3: Conversion of some system's terms [7].

WaterML 2.0	RiverCure portal	SNIRH and SVARH
Property	Parameter	Parameter
Vertical datum	Altitude	Station height or altitude
Catchment	River basin	Basin
Monitoring Point	Monitoring point	Station
Responsible Party	Responsible party	Description

Another important term is the concept of event since it can have many different definitions, that can vary according to authors and contexts. According to L.H. Broska et al. an event can be defined as a "Dynamic occurrence within a limited time frame fixed in time. Typically, events are not strictly spatially bounded, although ex post the spatial extent of an event should be clearly definable" [70]. An extreme

event can be characterized according to the intensity, frequency, duration or magnitude of it. To assess an extreme event a measurement index must be selected and thresholds can be defined based “on historical values, *probabilities of occurrence or on the point where they have potential consequences or impacts*” [71].

5.2. Stakeholders

This system has several stakeholders, each has a different impact, and it can be impacted differently.

Since the RCP is part of the RiverCure project, the stakeholders are the same. Specification 20 shows that the RiverCure project is sponsored by FCT and it has three partners: IST, INESC-ID and APA.

There are several people involved in the project, some are part of the team RiverCure and others belong to the partners’ organization (e.g. APA responsible). The project manager is responsible for managing the development of the RiverCure project, the team and the communication between the stakeholders.

In the RCP, a user is someone that has a user account. SuperAdmin, Manager, Technician and Citizen are the four main types of users, each with different access permissions. These are explained in detail in the next section.

HiSTAV and Social flood are considered as system internal (Specification 20) since the two tools might be incorporated into the RCP system. SVARH will transfer its data to the RiverCure server, so it is considered an external system.

```
//Persons
Stakeholder stk_user "User" : Person : Person_User [description "(See Glossary)"]
Stakeholder stk_SuperAdmin "SuperAdmin" : Person : Person_User [isa stk_user partOf stk_team_RC description "(See Glossary)"]
Stakeholder stk_manager "Manager" : Person : Person_User [isa stk_user partOf stk_team_RC description "(See Glossary)"]
Stakeholder stk_technician "Technician" : Person : Person_User [isa stk_user description "(See Glossary)"]
Stakeholder stk_citizen "Citizen" : Person : Person_User [isa stk_user description "(See Glossary)"]
Stakeholder stk_resp_APA "APA Responsible" : Person [partOf stk_APA description □]

//Systems
Stakeholder stk_SVARH "SVARH" : System : System_External [partOf stk_APA description "(See Glossary)"]
Stakeholder stk_HiStav "HiStav" : System : System_Internal [partOf stk_RC_System description "(See Glossary)"]
Stakeholder stk_SocialFlood "Social flood" : System : System_Internal [partOf stk_RC_System description "(See Glossary)"]
Stakeholder stk_RC_System "RiverCure System" : System : System_Internal [description "(See Glossary)"]

//Others
Stakeholder stk_project_manager "RiverCure project manager" : Person : Person_Manager_ProjectManager [partOf stk_team_RC]
Stakeholder stk_team_RC "RiverCure team" : Organization [isa stk_SuperAdmin]
Stakeholder stk_IST "Instituto Superior Técnico" : Organization : Organization_Partner
Stakeholder stk_INESC_ID "INESC-ID" : Organization : Organization_Partner
Stakeholder stk_APA "Agência Portuguesa do Ambiente" : Organization : Organization_Partner [description □]
Stakeholder stk_FCT "Fundação para a Ciência e Tecnologia" : Organization : Organization_Sponsor
```

Specification 20: RiverCure project stakeholders.

5.3. Actors

The RiverCure portal has three types of actors, external system, user and timer, as it can be observed in Specification 21. The SVARH and all the user types are actors of the system and each one has a specific stakeholder associated with it. The “aT_NotifyAlarm” actor is triggered when an alarm from a sensor is active and the “aT_AccountNotValidated” actor is responsible for initiating a use case when

an account is not validated after 24 hours of being submitted. These timers and its use cases are explained in detail in section 5.5.

```

Actor aS_SVARH "SVARH" : ExternalSystem [stakeholder RCPortal_Business.stk_SVARH]

Actor aU_User "User" : User [stakeholder RCPortal_Business.stk_user]
Actor aU_Citizen "Citizen" : User [isA aU_User stakeholder RCPortal_Business.stk_citizen]
Actor aU_Technician "Technician" : User [isA aU_User stakeholder RCPortal_Business.stk_technician]
Actor aU_Radar "Radar technician" : User [isA aU_Technician stakeholder RCPortal_Business.stk_technician]
Actor aU_Num "Numeric technician" : User [isA aU_Technician stakeholder RCPortal_Business.stk_technician]
Actor aU_Udo "Udometric technician" : User [isA aU_Technician stakeholder RCPortal_Business.stk_technician]
Actor aU_Spatial "Spatial technician" : User [isA aU_Technician stakeholder RCPortal_Business.stk_technician]
Actor aU_Crowd "Crowdsourcing technician" : User [isA aU_Technician stakeholder RCPortal_Business.stk_technician]
Actor aU_Hydro "Hydrometric technician" : User [isA aU_Technician stakeholder RCPortal_Business.stk_technician]
Actor aU_Manager "Manager" : User [isA aU_User stakeholder RCPortal_Business.stk_manager]
Actor aU_SuperAdmin "SuperAdmin" : User [isA aU_User stakeholder RCPortal_Business.stk_SuperAdmin]

Actor aT_NotifyAlarm "Notify active alarm" : Timer [description "Notify user that alarm is active"]
Actor aT_AccountNotValidated "Account not validated after 24h" : Timer [description "Account is not validated after 24h of submission"]

```

Specification 21: RiverCure actors in RSL.

A RiverCure portal user has a main role (SuperAdmin, Manager, Technician or Citizen) and if the main role is “Technician” it can have one or more secondary roles. Figure 31 presents the six types of secondary roles:

- Radar technician: Person that is specialized in radar images (e.g. IPMA employee);
- Numeric technician: User that knows about numeric models (e.g. Researcher);
- Udometric technician: Specialized in udometric stations and consequently data (e.g. APA responsible);
- Spatial technician: GIS is the area of expertise of this person (e.g. Researcher, APA or IPMA employee);
- Crowdsourcing technician: User that is specialized in the crowdsourcing topic (e.g. Researcher);
- Hydrometric technician: Specialist in hydrometric stations and data (e.g. APA employee).

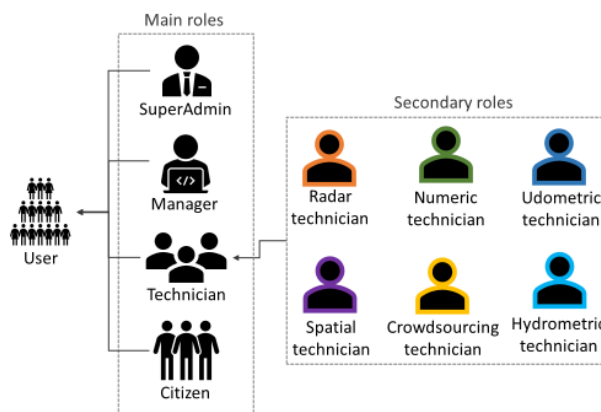


Figure 31: RiverCure portal user roles.

The permissions of user roles accumulate, hereby, for instance, the actor manager (father actor) has all the permissions that the technician (son actor) have and a couple more.

Table 4 (in the annexes) presents the tasks that each user role has permission to execute. This table supports the affirmations made in Figure 31, the SuperAdmin has accumulated the permissions of the manager, that can perform the tasks of the technicians and citizens. However, managing alarms has a

restriction, a Hydrometric/Udometric technician can only edit and delete the sensor alarms if the legislation of the country in question allows it.

5.4. Data Entities

The data entities of this information system are defined in this section and presented using either UML diagrams or RSL specifications.

It was defined eight main data entities packages in the RiverCure system: user, simulation, sensor, event, interactive map, about, spatial and feature. Part of the structure of these entities was inspired by the SVARH data entity structure since it is one of the main contributors of this portal.

To describe the spatial location of an object, five data entities were defined, as can be seen in Specification 22. The data entity “e_Spatial_Location” describes the specific location of an object by stating its coordinates and it is associated with a specific parish. Consequently, a parish belongs to a municipality, that is linked to a district of a specific country. To describe a parish, municipality, district and country, two main attributes were defined for each, name and code. Since the case study occurs in Portugal, the parish, municipality and district codes (dicofre, dico and di respectively), were defined according to the Portuguese constitution. The country code chosen was alpha-2 code from ISO 3166, since one of the future goals is to expand this portal to other countries.

```
DataEntity e_Spatial_Location "Spatial location" : Master [
  attribute coord_x "Coordinate X" : GeoPoint
  attribute coord_y "Coordinate Y" : GeoPoint
  attribute latitude "Latitude" : GeoPoint [isNotNull]
  attribute longitude "Longitude" : GeoPoint [isNotNull]
  attribute CRS "Coordinate Reference System" : String [isNotNull]
  attribute dicofre "Dicofre" : Regex [isNotNull isUnique]
  primaryKey (latitude, longitude)
  foreignKey e_Parish (dicofre)]

DataEntity e_Parish "Parish" : Reference []

DataEntity e_Municipality "Municipality" : Reference []

DataEntity e_District "District" : Reference [
  attribute district_name "District name" : String [isNotNull isUnique]
  attribute di "Di" : Regex [isNotNull isUnique]
  attribute alpha2Code "Alpha 2-code" : Regex [isNotNull isUnique]
  attribute AlarmName "Alarm name" : String [isNotNull isUnique]
  primaryKey (di)
  foreignKey e_Country (alpha2Code)
  foreignKey e_Alarm (AlarmName)]

DataEntity e_Country "Country" : Reference []
```

Specification 22: RSL specification of RCP DataEntity: Spatial.

A “Hydrometric feature” data entity is used to describe a hydrometric object, such as a river, a dam or a basin. It can state its type, name, code and geometric shape (point, line or polygon). However, a river basin can also state its area and describe the rivers that composed it, by defining its hierarchy (main river or tributary) and length, as it can be seen in Figure 32. Each hydrometric feature is associated with one or more municipalities since it can cross several municipal frontiers. For instance, the Vouga river crosses Oliveira de Frades, Vouzela and many other municipalities.

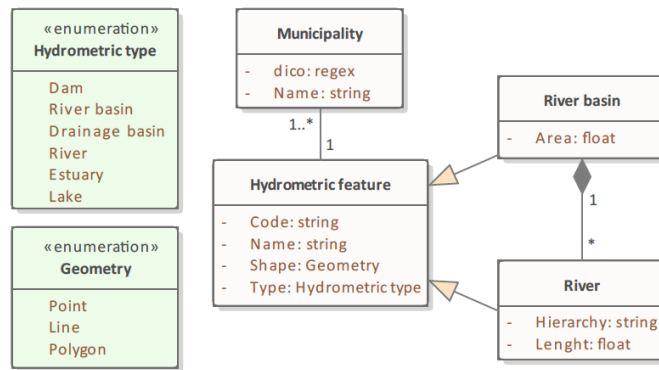


Figure 32: UML diagram of the RCP DataEntities: Feature package.

From Figure 33 it is possible to understand that the sensor entity is characterized by code, name, type and it measures a specific parameter. There are three types of sensors, the fixed in-situ, which are sensors that register values regarding an exact, established location (e.g. SVARH sensors), the weather radar, which is a fixed sensor that can capture data remotely about several places (e.g. weather radars from IPMA) and the photo, that it is a mobile sensor since the camera operator can shift location. Fixed in-situ sensors and weather radars have more than one user responsible for it, but the photo sensor normally only has one.

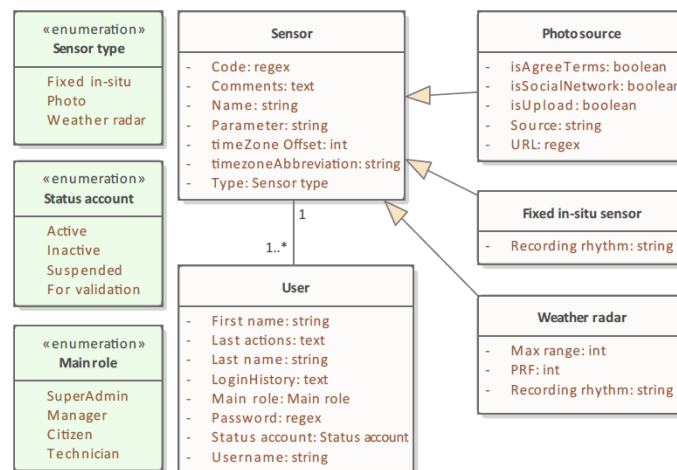


Figure 33: UML diagram of the RCP DataEntities: Sensor simple package.

As it can be seen in Figure 34, a monitoring point is composed of one or more sensors, that register several data timeseries, which are a series of time-value pairs. The “e_MonitoringPoint” data entity states the equipment altitude, “descriptionReference”, which is a list of documents related to the monitoring point, code, name, status (e.g. active, suspended) and the type of monitoring point (e.g. Hydrometric, Meteorological), which depends on the type of sensor associated. To be able to know the closest monitoring points of a specific point, the next/previous monitoring point relation was defined. By using upstream/downstream instead of next/previous, it would only be possible to use it on Hydrometric and Dam monitoring points. This way it is possible to apply this relation to all the types of monitoring points. Each sensor has a specific location set and at least one hydrometric feature associated. For example, a meteorological sensor can state the name of the basin to which is associated and a hydrometric indicates the basin and river name. A recording rhythm is defined for

each sensor, which means that a value is recorded every 15 minutes, for instance. The hydrometric sensor has a special attribute, the zero level scale, which indicates the level where zero of the scale is set.

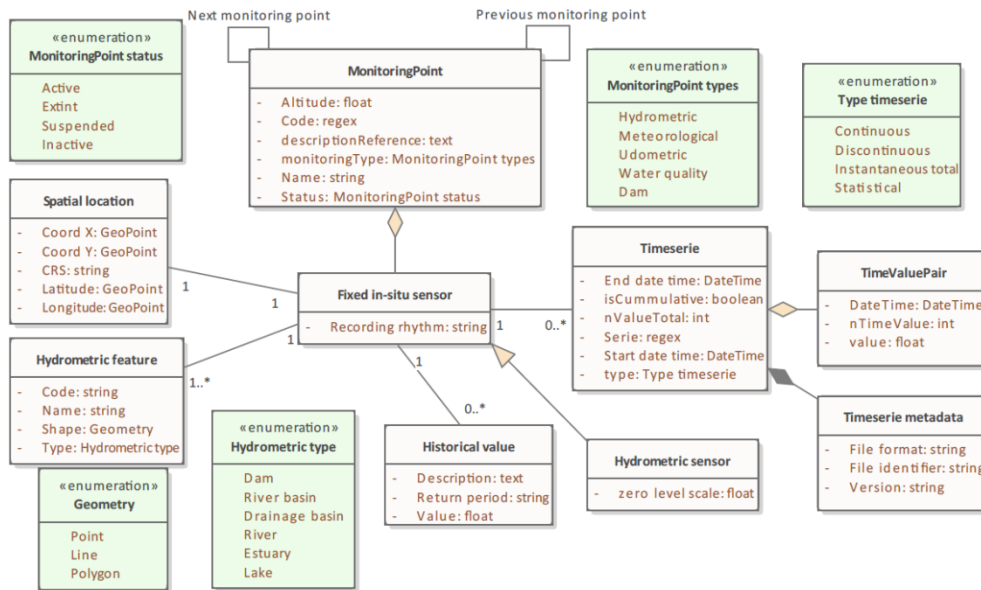


Figure 34: UML diagram of the RCP DataEntities: Fixed in-situ package.

A fixed in-situ sensor allows to define up to three alarms for a parameter, as it can be seen in Figure 35. Each one with a specific colour (red, yellow or green), name, description and action, which can be to send a notification to a certain user, for example. A low and up threshold must be defined for each alarm and to do, so the historical measurements (“Historical value”) of each location can be consulted, in case there is any. The “District” and the “River basin” data entities are also associated with the “Alarm” data entity since the alarm of sensors can be represented using those entities.

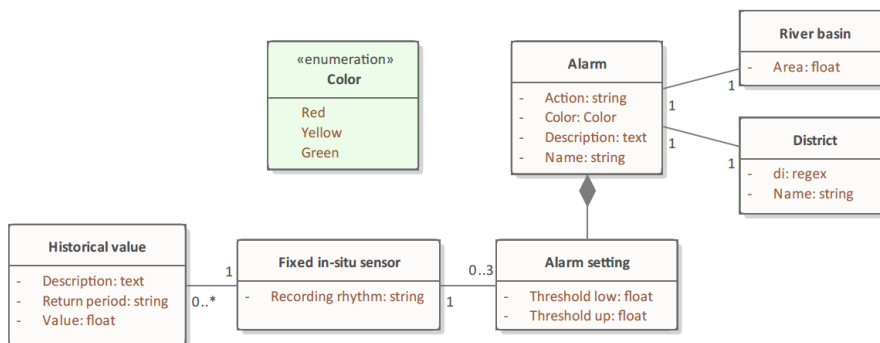


Figure 35: UML diagram of the RCP DataEntities: Alarm package.

From Specification 23 it is possible to understand that the weather radar is a generalization of the “e_Sensor” and it has a latitude and longitude associated, which represents the location of the equipment. The “e_RadarObservation” is the data entity that defines the data obtained from the radar. Since the data obtained is in image format, the only attributes are the ones associated with it such as date, time, ID, file identifier and version.


```

DataEntity e_WeatherRadar "Weather radar" : Master [
  isA e_Sensor
  attribute MaxRange "Max range" : Integer [isNotNull]
  attribute PRF "Pulse repetition frequency (PRF)" : Integer [isNotNull]
  attribute recRhythm "Recording rhythm" : String [isNotNull]
  attribute SensorCode "Sensor code" : Regex [isNotNull isUnique]
  attribute latitude "Latitude" : GeoPoint [isNotNull]
  attribute longitude "Longitude" : GeoPoint [isNotNull]
  foreignKey e_Spatial_Location (latitude, longitude)]

DataEntity e_RadarObservation "Radar observation" : Reference [
  attribute date "Date" : Date [isNotNull]
  attribute time "Time" : Time [isNotNull]
  attribute ID "ID" : String [isNotNull isUnique]
  attribute SensorCode "Sensor code" : Regex [isNotNull isUnique]
  attribute fileFormat "File format" : String [isNotNull]
  attribute fileIdentifier "File identifier" : String [isNotNull isUnique]
  attribute Version "Version" : String [isNotNull]
  primaryKey (fileIdentifier)
  foreignKey e_WeatherRadar (SensorCode)]

```

Specification 23: RSL specification of RCP DataEntity: Radar.

Figure 36 shows the data entities related to the photo sensor, which includes the "Photo source" entity, that states if the photo was uploaded or obtained from social networks. The "Photo observation" entity defines the main attributes of the photo, namely its ID, date, time, the value obtained from the social flood application and last modifications made, also known as history. Each photo observation has a specific location and it can have a hydrometric feature associated.

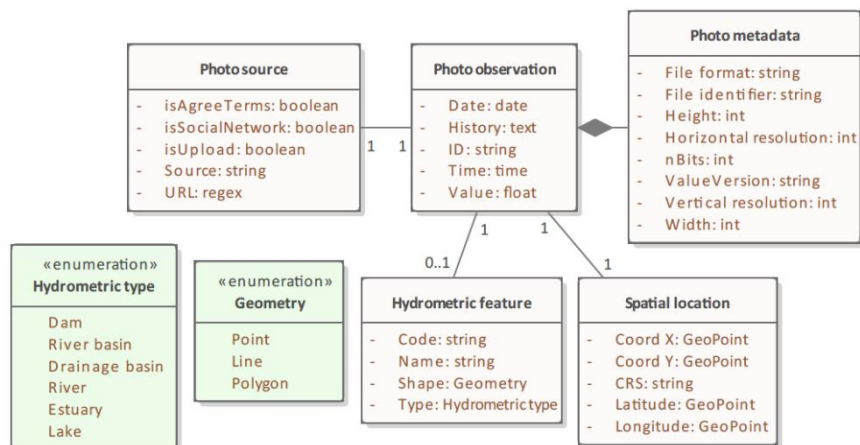


Figure 36: UML diagram of the RCP DataEntities: Photo package.

From Figure 37 it is possible to understand that an event is created by a single user, that is responsible for defining all its attributes, namely a code, a description, the state (if it is occurring or over), the type of event (e.g. Flood, Hydrological drought), and the start and end date-time. Each event has one or more fixed in-situ sensor alarms, which means that the "Alarm" and the "Fixed in-situ" data entities are linked to the "Event" data entity, and it can also have photos associated.

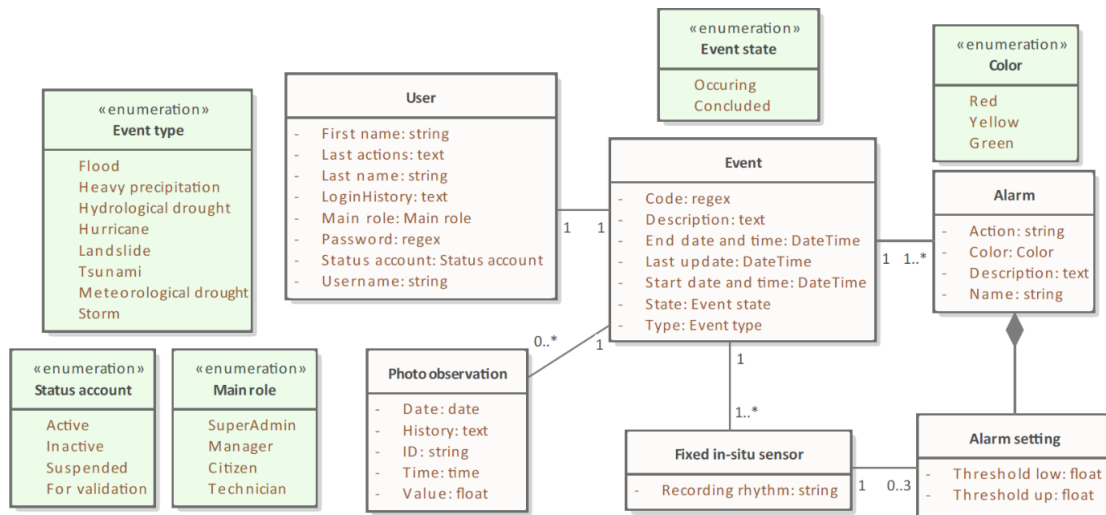


Figure 37:UML diagram of the RCP DataEntities: Event package.

To proper define a simulation the entities in Figure 38 were specified. The physical constants, time and numeric information and input files are the data entities required for the simulation to work. There are two types of simulations, the simulation and the scenario, which is a type based on the main simulation in which the “Time information” is the only data entity that is modified. The simulation results are a set of several output files, each one with a specific name and folder. Both simulation and input file have a user associated, that corresponds to the owner of the file. Each simulation is associated with one or more basins and districts, which allows an easier search process since it is not required to open each file to discover the location of a simulation.

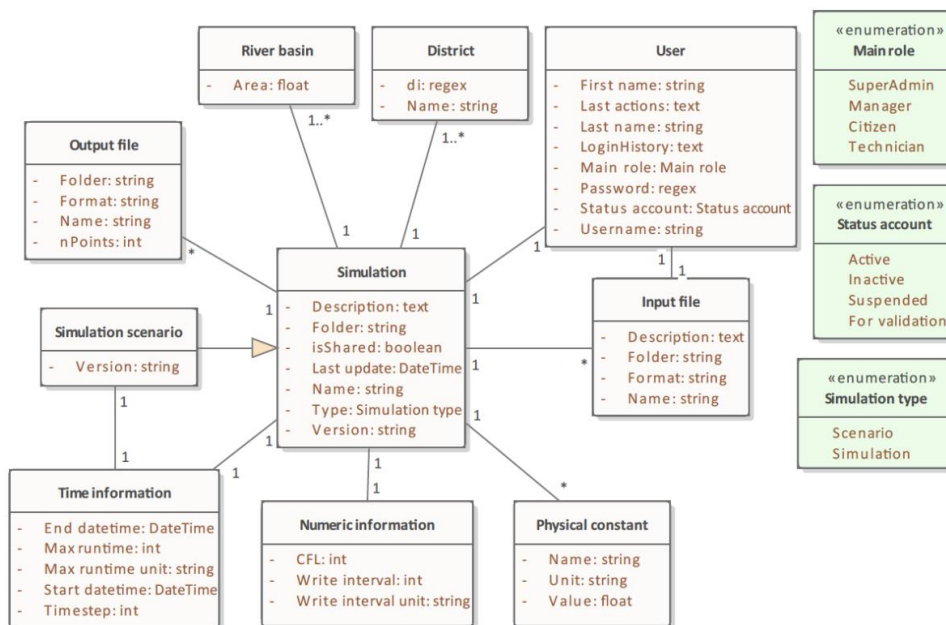


Figure 38: UML diagram of the RCP DataEntities: Simulation main package.

Specification 24 presents some of the possible “Input file” data entities that can be defined. Photos and weather radar observations regarding the Águeda case study were available, so they were selected as inputs for the simulation. The “e_InitialCondition”, “e_Mesh” and “e_BoundaryCond” data

entities are types of input files since they state in their specifications “isA e_InputFile”, which means that the e_InputFile” entity is a generalization of those entities.

```
DataEntity e_InputFile "Input file": Master []
DataEntity e_InitialCondition "Initial condition": Master : Detail []
DataEntity e_Mesh "Mesh": Master : Detail []
DataEntity e_BoundaryCond "Boundary condition": Master : Detail [
  isA e_InputFile
  attribute ID "ID" : String [isNotNull isUnique]
  attribute type "Type": DataEnumeration de_BoundaryType [isNotNull]
  attribute serie "Serie number" : Regex [isNotNull isUnique]
  attribute nElements "Number of elements" : Integer [isNotNull]
  foreignKey e_Timeserie (serie)
  foreignKey e_Mesh (nElements)]
DataEnumeration de_BoundaryType "Boundary type": Simple [values "Inlet", "Outlet"]
DataEntity e_DistributedCond "Distributed condition": Master : Detail []
DataEntity e_PhotoObservation "Photo observation": Document []
```

Specification 24: RSL specification of RCP DataEntity: Simulator input files.

The interactive map is composed of layers, that show different types of information (e.g. simulation layer, sensor layer), as it can be seen in Figure 39. On the map data regarding the last days or a time interval can be presented and the basemap can be altered. Each type of layer has a specific symbol, colour and set of attributes associated. For example, the event layer can show events that are occurring or that occurred in the last 48 or 72 hours and the simulation layer can present simulations or scenarios.

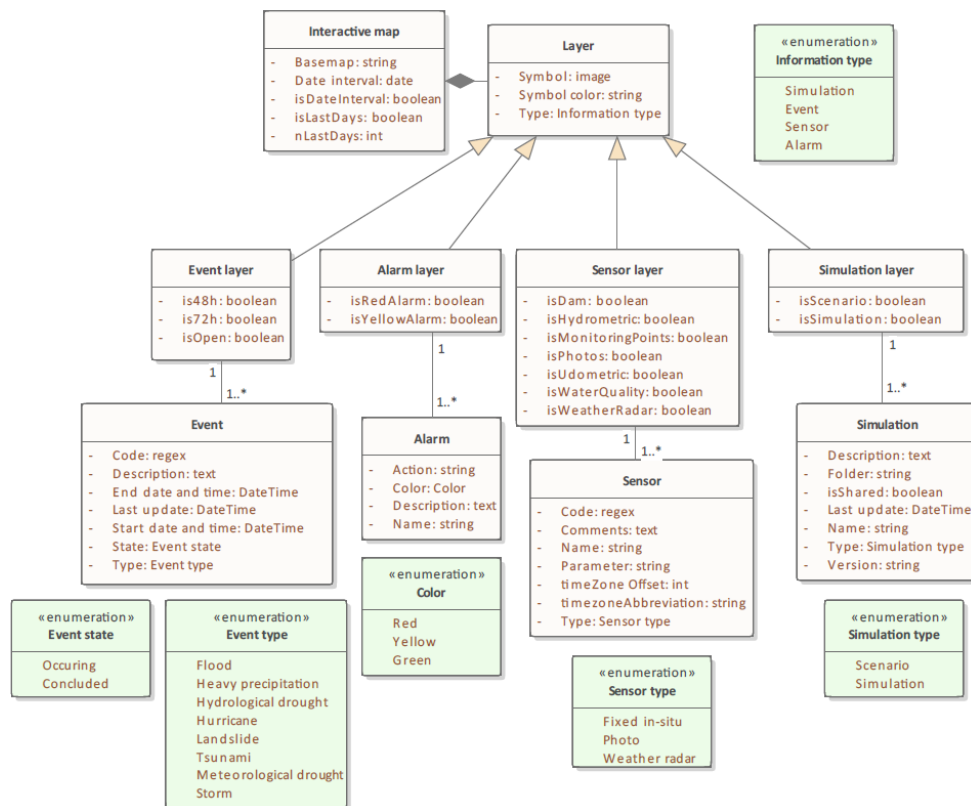


Figure 39: UML diagram of the RCP DataEntities: Map package.

In Specification 25 there are two data entities specified, one that describes the RiverCure project and its partners, and the other that defines the attributes required for the user to provide feedback regarding the portal. The “e_RCProject” data entity is associated with the “e_Entity” and “e_Contact” entities. The feedback requires the username of the user, so it is associated with the “e_User” data entity.

```
DataEntity e_RCProject "RiverCure project" : Master [
  attribute Description "Description" : Text [isNotNull]
  attribute lastUpdate "Last update" : Datetime [isNotNull isReadOnly]
  attribute entityName "Entity partner name" : String [isNotNull isUnique]
  attribute isPartner "is partner" : Boolean [defaultValue "False"]
  attribute email "Email address" : Regex [isNotNull isUnique isEncrypted]
  primaryKey (Description)
  foreignKey e_Entity (entityName, isPartner)
  foreignKey e_Contact (email)]

DataEntity e_Feedback "Feedback" : Master [
  attribute comment "Comment" : Text [isNotNull]
  attribute username "Username" : String [isNotNull isUnique]
  attribute number "ID Number" : Integer [isNotNull isUnique]
  primaryKey (number)
  foreignKey e_User (username)]
```

Specification 25: RSL specification of RCP DataEntity: About.

To define the user’s profile information several data entities were created, as it can be seen in Specification 26. The “User” entity is connected to the “e_Responsible Party” data entity, which describes the position that a user has in an entity and to the “Contact” data entity which states the email and phone number of the user. An “Entity” can be for example a company or an association and it must have an official address linked. Note that an “Entity” has many employees, each with a specific position, which means that an “Entity” has several users associated with it.

```
DataEntity e_User "User" : Master [
  attribute firstName "First name" : String [isNotNull isEncrypted]
  attribute lastName "Last name" : String [isNotNull isEncrypted]
  attribute mainRole "Main role" : String [isNotNull]
  attribute pass "Password" : Regex [isNotNull isEncrypted]
  attribute Status "Status account" : DataEnumeration de_StatusAccount [isNotNull]
  attribute username "Username" : String [isNotNull isUnique]
  attribute lastActions "Last actions": Text [isReadOnly]
  attribute loginHistory "Login history" : Text [isReadOnly]
  primaryKey (username)]

DataEnumeration de_MainRole "Main role" : Simple [values "SuperAdmin", "Manager", "Technician", "Citizen"]

DataEnumeration de_StatusAccount "Status account" : Simple [values "Active", "Inactive", "Suspended", "For validation"]

DataEntity e_Contact "Contact" : Reference [ ]

DataEntity e_ResponsibleParty "Responsible party" : Reference [
  attribute positionName "Position name" : String [isNotNull]
  attribute username "Username" : String [isNotNull isUnique]
  attribute entityName "Entity name" : String [isNotNull isUnique]
  primaryKey (positionName, username)
  foreignKey e_User (username)
  foreignKey e_Entity (entityName)]

DataEntity e_Entity "Entity" : Reference [ ]

DataEntity e_Adress "Adress" : Reference [ ]
```

Specification 26: RSL specification of RCP DataEntities: User main.

Specification 27 presents the data entities regarding user settings. To define the user settings, it is required to associate them with a user, so the username is the foreign key used to link the two data entities. The “e_UdoHydroSettings” entity describes special attributes for udometric and hydrometric technicians and the “e_TechnicianSettings” data entity specifies particular attributes for the technician users, which means that this entity is a generalization of the first.

```

DataEntity e_UserSettings "User settings" : Parameter [
  attribute NotFunction "Notification new functionalities" : Boolean [defaultValue "True"]
  attribute NotGallery "Notification new photo in gallery" : Boolean [defaultValue "False"]
  attribute NotSimul "Notification new simulation" : Boolean [defaultValue "False"]
  attribute NotSoftware "Notification software" : Boolean [defaultValue "True"]
  attribute areasInterest "Areas of interest" : GeoPoint
  attribute prefInfoType "Preferred information type" : DataEnumeration de_InformationType
  attribute homePageView "Home page view" : Text [defaultValue "Interactive map, Last actions, Last events"]
  attribute username "Username" : String [isNotNull isUnique]
  primaryKey (username)
  foreignKey e_User (username)]

DataEnumeration de_InformationType "Information type" : Simple [values "Simulation", "Event", "Sensor", "Alarm"]

DataEntity e_TechnicianSettings "Technician settings" : Parameter [
  isA e_UserSettings
  attribute viewCitizen "View perspective citizen" : Boolean [defaultValue "False"]
  attribute prefParameter "Preferred parameters" : Text]

DataEntity e_UdoHydroSettings "Udometric and Hydrometric settings" : Parameter [
  isA e_TechnicianSettings
  attribute isAuthoritiesAlarm "is Authorities alarms" : Boolean [defaultValue "True"]
  attribute isMyAlarm "is My alarms" : Boolean [defaultValue "False"]
  attribute isAlarmRed "is alarm red notification" : Boolean
  attribute isAlarmYellow "is alarm yellow notification" : Boolean
  attribute isViaSMS "is via sms notification" : Boolean
  attribute isViaEmail "is via email notification" : Boolean
  attribute MyAlarms "My alarms" : Text
  attribute AlarmName "Alarm name" : String [isNotNull isUnique]
  foreignKey e_Alarm (AlarmName)]

```

Specification 27: RSL specification of RCP DataEntities: User settings.

From Figure 40 is possible to understand that the user permissions depend on the user role and are associated with the river basins, which means that the user can only access the information regarding a specific set of river basins. In terms of alarm permission, some users can access all the alarms, others can only access alarms created by the authorities, others depend on the secondary role can or cannot access the alarms and some do not have access to any alarm.

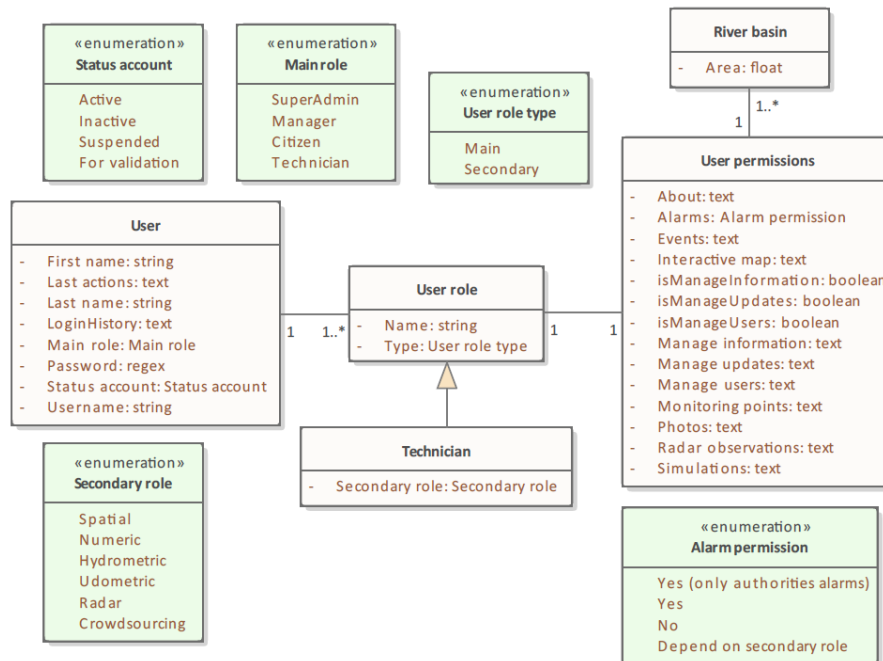


Figure 40: UML diagram of the RCP DataEntities: User permissions package.

To demonstrate the logical connections between the data entities, data entity clusters were created, and some can be seen in Specification 28. For example, “ec_IntercractiveMap” is composed by a

master data entity, the “e_IntegrativeMap” and a detail entity, “e_Layer”, which has several reference data entities linked, such as, “e_Event” and “e_Sensor”.

```

DataEntityCluster ec_Sensor "Sensor" : Complex []
DataEntityCluster ec_Fixed_Insitu "Sensor fixed in-situ" : Complex []
DataEntityCluster ec_Weather_radar "Sensor weather radar" : Simple [master e_WeatherRadar detail e_RadarObservation]
DataEntityCluster ec_Photo "Sensor photo" : Complex []
DataEntityCluster ec_Alarm "Alarm" : Complex []
DataEntityCluster ec_Event "Event" : Complex []
DataEntityCluster ec_About "About" : Complex [
  master e_RCProject
  detail e_Contact [reference e_User]
  reference e_Entity]

DataEntityCluster ec_Feedback "Feedback" : Simple [master e_Feedback reference e_User]

DataEntityCluster ec_Spatial "Spatial" : Complex []
DataEntityCluster ec_Feature "Feature" : Simple [master e_HydroFeature reference e_Municipality]

DataEntityCluster ec_Simulation "Simulation" : Complex []
DataEntityCluster ec_BoundaryCond "Boundary condition" : Complex []
DataEntityCluster ec_InitialCond "Initial condition" : Complex []
DataEntityCluster ec_InteractiveMap "Interactive map" : Complex [
  master e_InteractiveMap
  detail e_Layer [reference e_Event, e_Alarm, e_Sensor, e_Simulation]]

DataEntityCluster ec_User_simple "User (simple)" : Complex [
  master e_User
  detail e_UserSettings [reference e_Alarm]
  detail e_Contact
  detail e_ResponsibleParty [reference e_Entity, e_Adress]
  reference e_UserRole]

DataEntityCluster ec_UserInformation "User (information)" : Complex []
DataEntityCluster ec_User_Role "User role" : Complex []

```

Specification 28: RSL specification of RCP DataEntityClusters.

5.5. Requirements

This section describes the use cases of the RiverCure portal using either UML diagrams or RSL specifications. The UML allows the user to have a quicker overview and the RSL provides more details regarding the use cases, namely the data entities associated with it.

To suggest the user interaction and the involved information, RiverCure portal mock-ups were also developed.

Figure 41 shows the home page mock-up of the RiverCure portal and as it can be seen, it has five main tabs on the top of the page: data, photos, simulations, events and about. The information presented on this page can be changed according to the user’s preferences. The last actions made by the user on the platform, the last events added, and the interactive map are the information displayed in the mock-up. The user permissions can also be visualized on the top right section of the page, below the username, and the settings and the log out button. The icon located next to the mock-up indicates the users’ roles that can access the mock-up view.

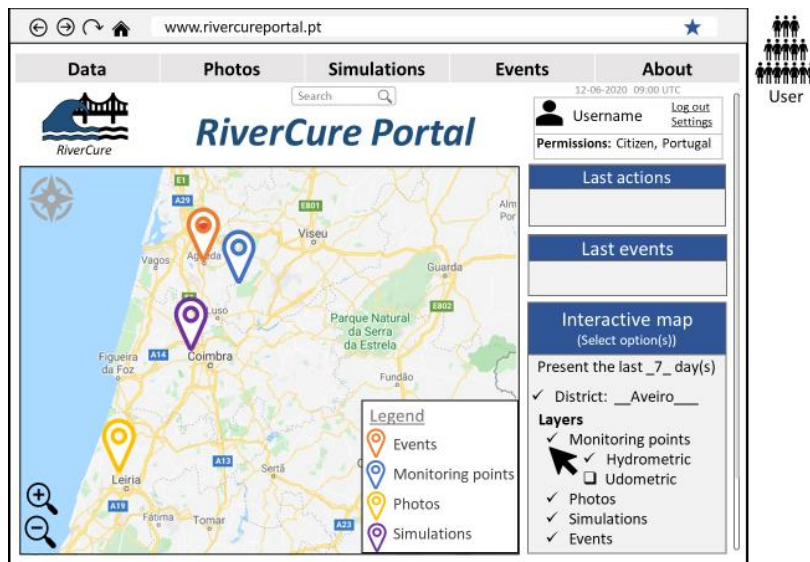


Figure 41: Home page mock-up of the RCP.

The data tab allows the user to access the interactive map tool, information regarding SVARH monitoring points and radar observations, as it can be seen in Figure 42 a). If the user selects the interactive map tool, the interface in Figure 42 b) will appear, allowing the user to view the data selected on the control panel on the right. Each point of information is represented with a distinct symbol and it has a set of data associated that pops-up by clicking on it. Note that the information available on this tool varies according to the user role.

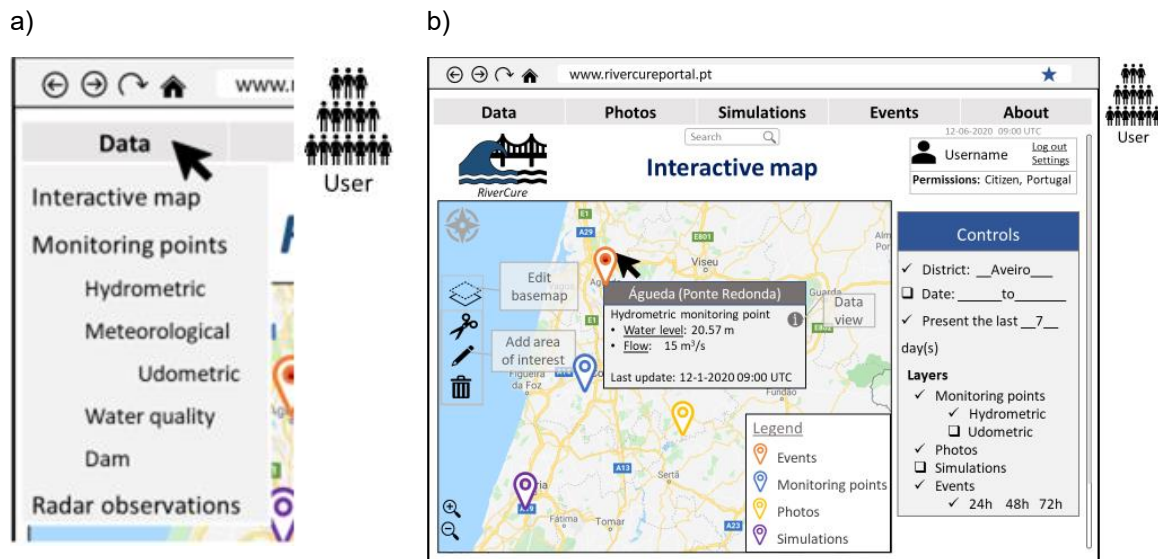


Figure 42: a) Data menu mock-up of the RCP; b) Interactive map mock-up of the RCP.

Figure 43 presents the UML diagram of the use cases that a user can do when consulting the interactive map. The use case “Consult interactive map” includes the use case “Select layers”, because the user must choose the types of information that wishes to see, otherwise, it would not appear any information. A user can also edit the basemap (e.g. satellite, administrative) and manage areas of interest, which includes adding, deleting, editing areas selected on the map. Figure 42 b) demonstrates all these use cases defined from the user perspective.

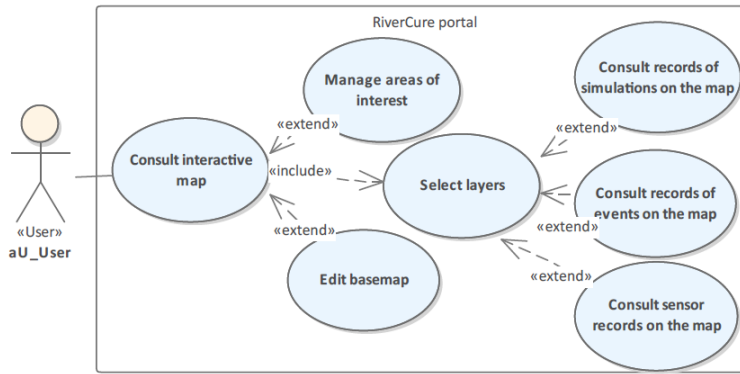


Figure 43: UML diagram of the RCP UseCases: Map package.

To consult the monitoring points data the user must choose one of three views, by basin, by district or by monitoring point, as it can be seen in Figure 44. By selecting the option by basin or by district, a schema of a monitoring point in that area opens showing the last data record stored (Figure 45 a)). The option by monitoring point shows a list of all the points with data available and the user can select which one it wishes to read the data records (Figure 45 b)). The user can also view the data in an alternative mode, by creating a graphic with the data, so the “Consult monitoring point data” use case has a use case extension, the “Create graphic”.

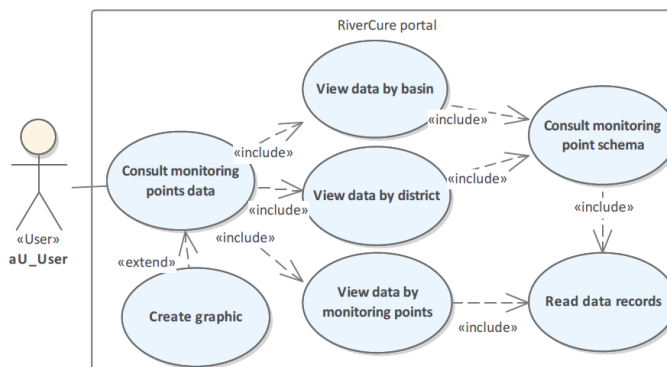


Figure 44: UML diagram of the RCP UseCases: Monitoring points package.

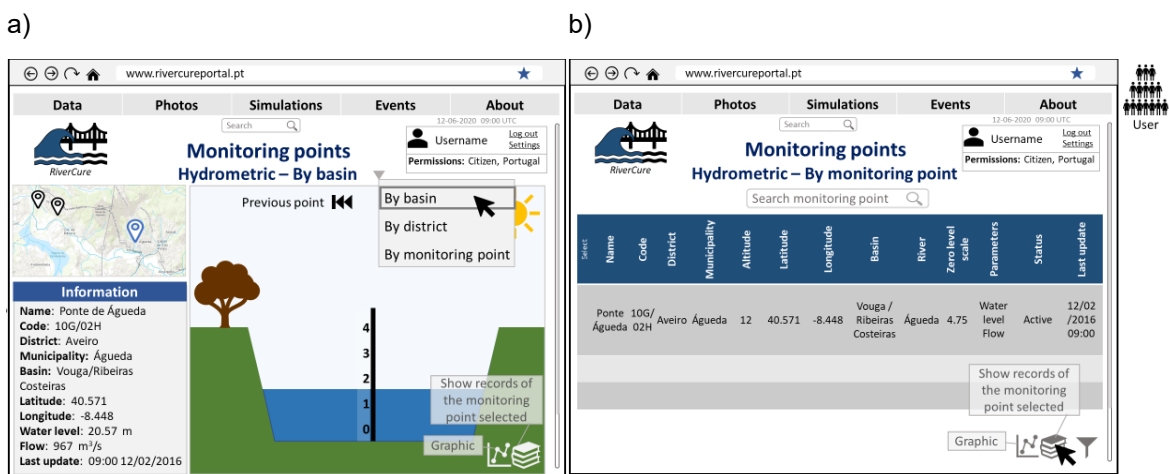


Figure 45: a) Monitoring points by basin (User) mock-up; b) By monitoring points (User) mock-up.

From Figure 46 is possible to understand that the hydrometric and udometric technicians have some use cases specific to them regarding the monitoring points and the interactive map. These user roles can check real-time information from the monitoring points (“Read real-time data”) and to flag errors in any monitoring point or data.

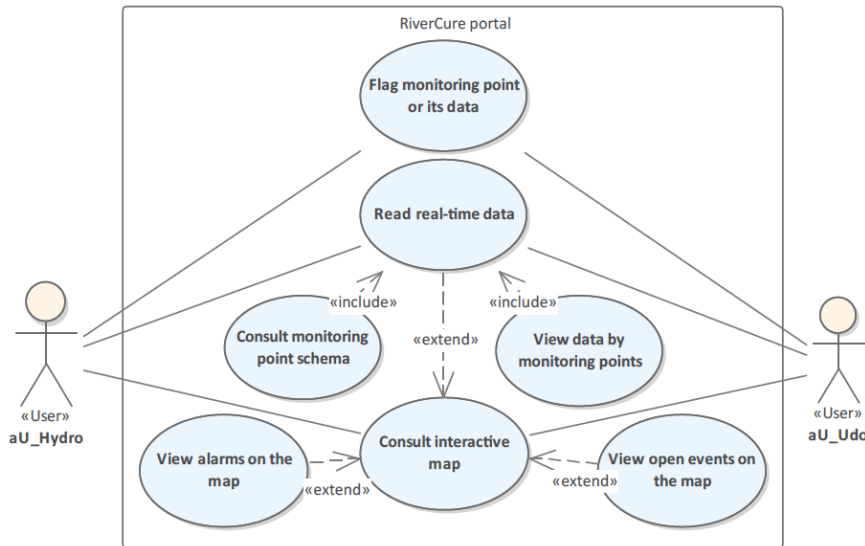


Figure 46: UML diagram of the RCP UseCases: Hydrometric and Udometric data package.

Figure 47 shows the mock-ups of the hydrometric monitoring point interface in two different perspectives, the user (Figure 47 a)) and the hydrometric technician (Figure 47 b)). As it can be seen Figure 47 b) presents the alarm levels on the maps and the user mock-up (Figure 47 a)) does not, since this user does not have access to real-time data either to the alarm information.

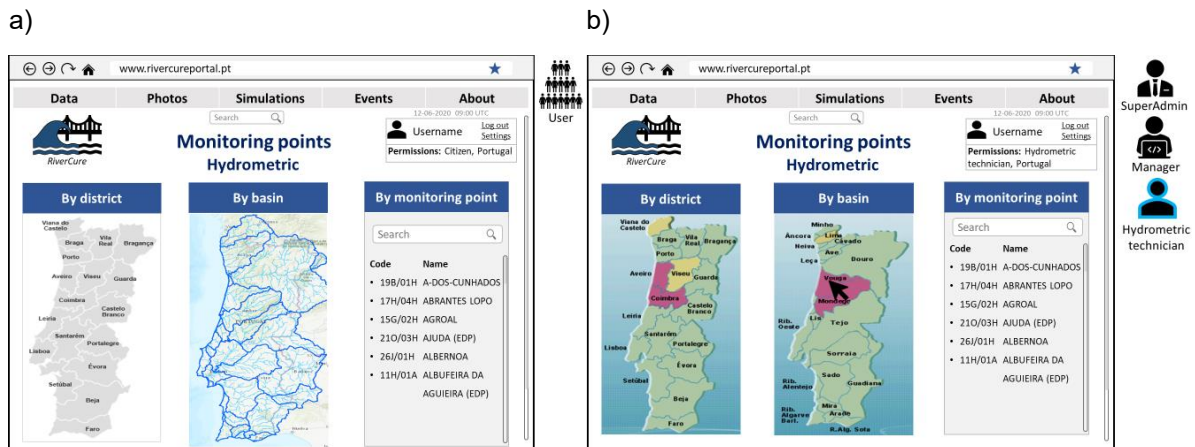


Figure 47: a) Hydrometric monitoring points (User) mock-up; b) Hydrometric monitoring points (Hydrometric technician) mock-up.

A user can consult radar observations, by selecting a radar, a parameter and a date on the search control, which means that “uc_3_ConsultRadarData” is a use case of the type “EntitiesBrowse”. This use case has an extension point the “uc_3_1_ExportRadar”, which allows the user to export the radar observations, as it can be checked in Specification 29. To demonstrate these sequences of actions a mock-up of the interface was made (Figure 48).

```

UseCase uc_3_ConultRadarData "Consult radar observations data" : EntitiesBrowse [
  actorInitiates aU_User
  dataEntity ec_Weather_radar
  actions aRead, aExport, aSelect
  extensionPoints EExportRadar
  description "Consult the data obtained from the weather radars"]

UseCase uc_3_1_ExportRadar "Export radar observations" : EntitiesInteropExport [

```

Specification 29: RSL specification of RCP UseCases: Radar.

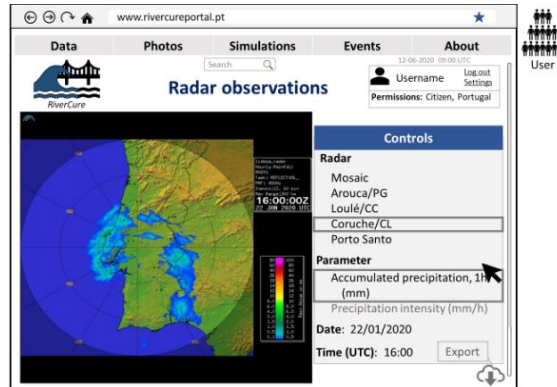


Figure 48: Radar observations mock-up of the RCP.

The “Photos” tab of the RiverCure mock-up in Figure 49 a) has two selection options, the first allows the user to upload photos in the portal and the second is the gallery, which is where all the photos are displayed.

For a user to upload a photo in the portal it is required to read and agree with the terms and conditions defined, as it is demonstrated in Figure 49 b) and specified in the “uc_4_1_AgreeTerms” (Specification 30).

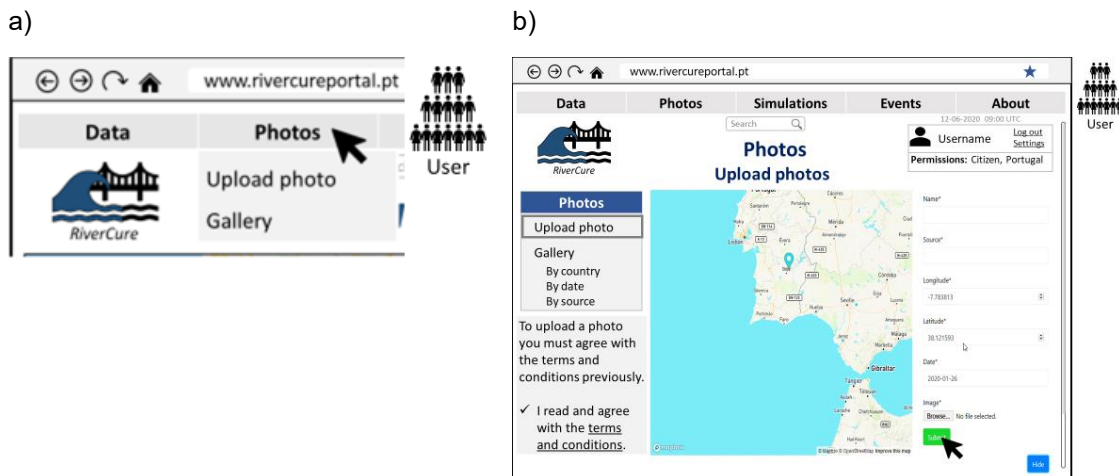


Figure 49: a) Photos menu mock-up of the RCP; b) Upload photos mock-up of the RCP.

Specification 30 defines the use cases triggered by the user, the spatial technician, and the crowdsourcing technician, regarding photos. All users have access to the photo gallery (Figure 50 a)) and can select a photo in order to read its information, which includes its metadata and the estimated water height. When consulting a photo, a user can export, select, flag, view the next or the previous image without any restrictions, but it can only delete the photo or edit its metadata if it is owned by the

user. The spatial and crowdsourcing technicians have use cases specific to them, such as “uc_7_1_EditPhotoInfo”, which allows editing the photo’s metadata and estimated height, “uc_7_2_ConsultPhotoInfoHistory”, which permits these users to check the modifications made to the photo’s information and “uc_7_3_DeletePhoto. All these use cases are extension points of the “uc_7_ManagePhotos” use case and their application can be seen in Figure 50 b).

```

UseCase uc_4_UploadPhotos "Upload photos" : EntitiesInteropImport [
  actorInitiates aU_User
  dataEntity ec_Photo
  actions aUpload
  includes uc_4_1_AgreeTerms]

UseCase uc_4_1_AgreeTerms "Read and Agree with the terms" : EntityRead []

UseCase uc_5_ConsultPhoto "Consult photo" : EntitiesBrowse [
  actorInitiates aU_User
  dataEntity ec_Photo
  actions aRead, aExport, aDelete, aFlag, aEdit, aSelect, aNext, aPrevious
  extensionPoints EPExportPhoto, EPDeletePhotoOwned, EPFlagPhoto
  extends uc_15_ConsultEventsRecords onExtensionPoint EPConsultPhoto]

UseCase uc_5_1_ConsultInfo "Consult photo information" : EntitiesBrowse []

UseCase uc_5_1_1_EditMetadataOwned "Edit metadata of photo owned" : EntityUpdate []

UseCase uc_5_2_ExportPhoto "Export photo" : EntitiesInteropExport []

UseCase uc_5_3_DeletePhotoOwned "Delete photo owned" : EntityDelete []

UseCase uc_5_4_FlagPhoto "Flag photo" : EntityOther []

UseCase uc_6_ConsultGallery "Consult photo gallery" : EntitiesBrowse []

UseCase uc_7_ManagePhotos "Manage photos" : EntitiesManage [
  actorInitiates aU_Crowd
  actorParticipates aU_Spatial
  dataEntity ec_Photo
  actions aEdit, aRead, aSelect
  extensionPoints EPEditPhotoInfo, EPConsultPhotoInfoHistory, EPDeletePhoto]

UseCase uc_7_1_EditPhotoInfo "Edit photo information (not owned)" : EntitiesManage []

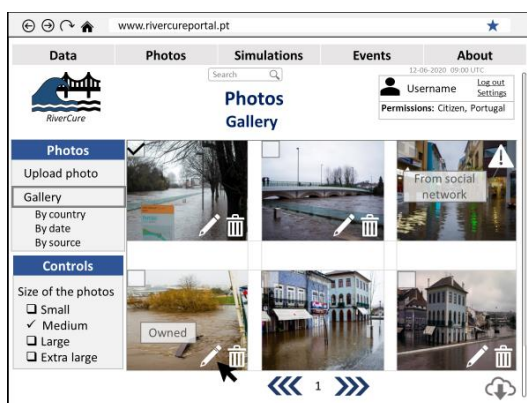
UseCase uc_7_2_ConsultPhotoInfoHistory "Consult photo information history" : EntitiesBrowse []

UseCase uc_7_3_DeletePhoto "Delete photo from other users" : EntityDelete []

```

Specification 30: RSL specification of RCP UseCases: Photos.

a)



b)



Figure 50: a) Photo gallery mock-up of the RCP; b) Photos gallery (Crowdsourcing technician) mock-up of the RCP.

The simulator tab has two versions, one for the general users (Figure 51 a)) and other for the numeric and spatial technician (Figure 51 b)) since these last two roles have access to special features, such as creating a new simulation from scratch.

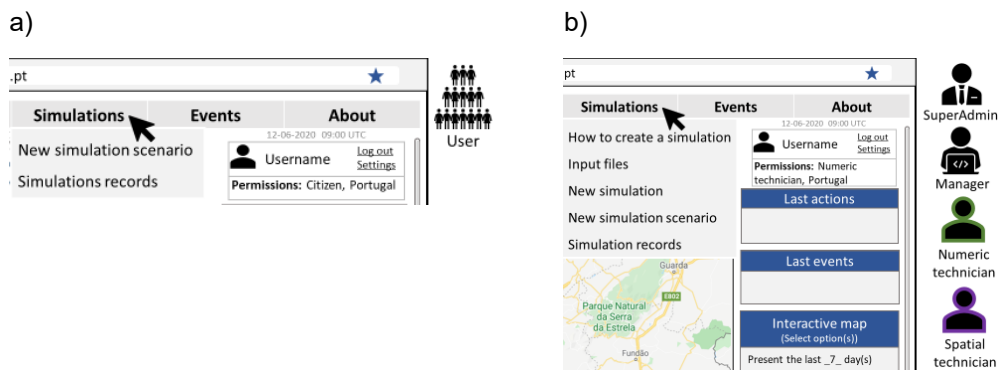


Figure 51: a) Simulations menu mock-up; b) Simulations menu (Numeric technician) mock-up of the RCP.

To create a new scenario the user must define its input data previously, which includes the area of study (also known as domain), and the start and end date-time, as it can be seen in Figure 52 a). While the simulator is running, a progress indicator is shown on the screen (Figure 52 b)), so the user can track the simulation status. During this time the portal can suggest data related to the simulation domain, such as photos and data from monitoring points.

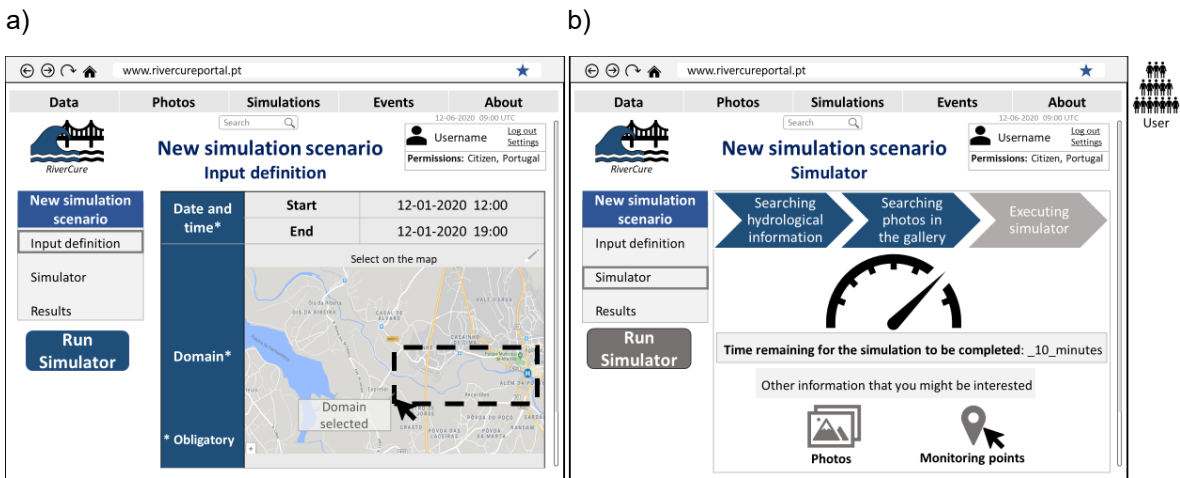


Figure 52: a) New simulation scenario (Input definition) mock-up; b) New simulator scenario (Progress indicator) mock-up of the RCP.

Besides the use case “Create new simulation scenario”, a user has other two main use cases regarding simulations “Consult results” and “Consult simulation records”, which are specified in Figure 53. A user can choose to save or not the simulation scenario obtained, and it can opt to export those results. In the simulation records tab is available all the simulations saved by the user and the ones published by specialized users, like spatial technicians, for instance. The “Consult simulation records” extension point, the “Manage scenarios owned” use case, allows the user to edit and delete simulation scenarios owned by the user, as it can be observed from the mock-up in Figure 54.

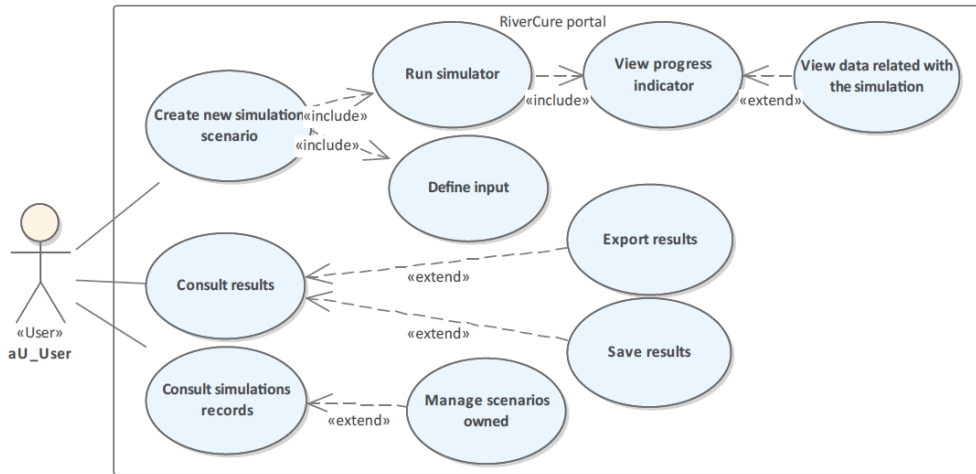


Figure 53: UML diagram of the RCP UseCases: Simulations package.

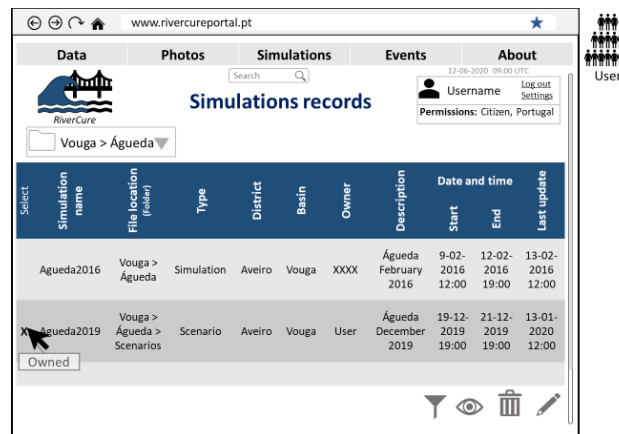


Figure 54: Simulations records mock-up of the RCP.

As formerly mentioned, the numeric and spatial technicians can create new simulations, but to understand how the simulation tool works it is important to read the instructions. For this tool to work, input files (e.g. DTM, Domain) are required, so in the simulations tab was created a sub-tab, "Input files", as can be observed in Figure 51 b). From Figure 55 it is possible to understand that a user can view the files, manage the files owned and upload new files, which may encourage information sharing.

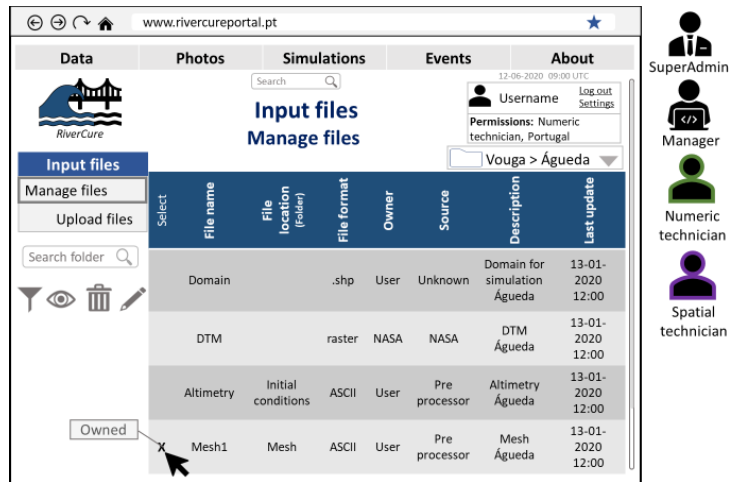


Figure 55: Simulations input files (Manage files) mock-up of the RCP.

Figure 56 presents all the use cases regarding simulations, that have the spatial and numeric technicians as actors. The “Create new simulation” use case has an extension point, “Run the Pre processor” since this use case is only required if a mesh of the area that is being studied is not created or if it can be updated to a better version. To develop a new simulation, the user must first configure the model and then execute the simulator tool (“Run simulator”). The “Configure simulation model” has some use cases associated, with the relation “include”, which means that the user must perform these actions, and with the relation “extend”, which means that use case can be executed if the user wants or if it is feasible. According to the UML diagram, these technicians can choose to share or not the results of the simulations performed.



Figure 56: UML diagram of the RCP UseCases: Technician simulation package.

Figure 57 a) displays the interface mock-up of the pre-processor, which shows that the user must select the input files before running the pre-processor.

From Figure 57 b) it is possible to understand that a simulation model can use data from the radar observations, photos and monitoring points.

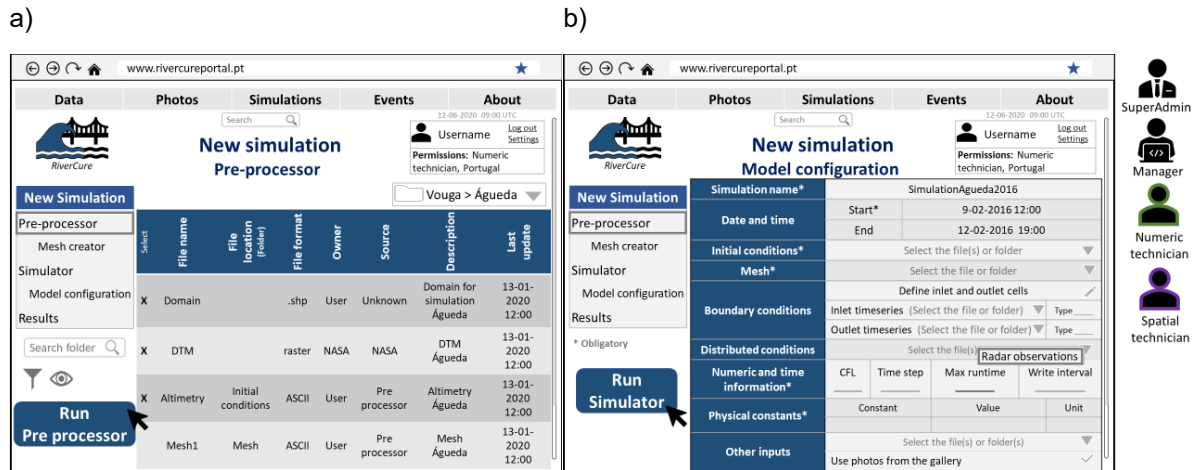


Figure 57: a) New simulation (Pre-processor) mock-up; b) New simulation (model configuration) mock-up of the RCP.

The user's events tab has only one division, that presents the events records (Figure 58 a)), but the udometric and hydrometric technicians events tab version has two, "Open events" and "Events records", as it can be seen in Figure 58 b).

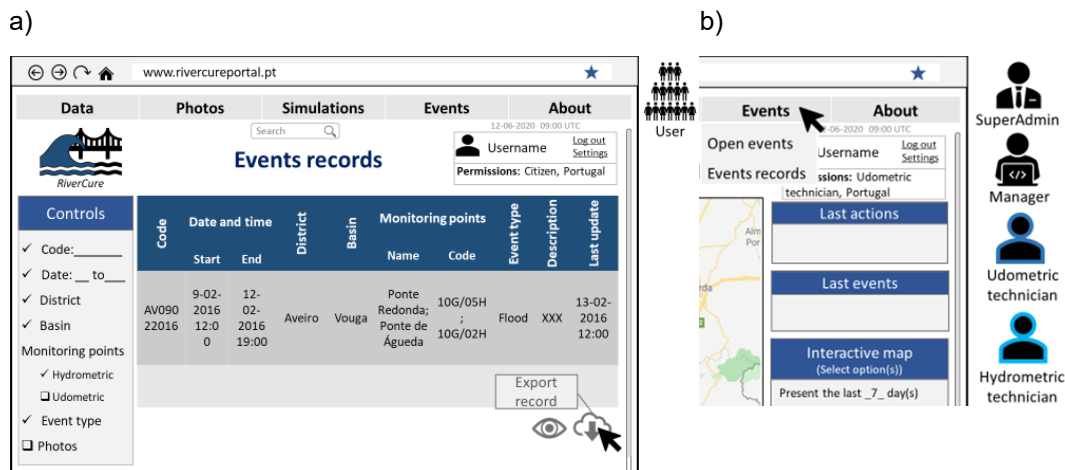


Figure 58: a) Events records mock-up of the RCP; b) Events menu (Udometric technician) mock-up.

From Specification 31 it is possible to understand that in an event record a user can export its data and consult the photos associated with it, in case there are any. Besides being able to execute the same actions as a user, a udometric and hydrometric technician can also flag event data errors and consult events that are still occurring. The use case "uc_16_ConsultOpenEvents" shows that it has three extension points "EPConsultPhoto", "EPFlagEventDataError" and "EPExportEventData".


```

UseCase uc_15_ConsultEventsRecords "Consult events records" : EntitiesBrowse [
  actorInitiates aU_User
  dataEntity ec_Event
  actions aFilter, aRead, aSelect
  extensionPoints EPExportEventData, EPConsultPhoto, EPFlagEventDataError
  description "Consult the records from past events"]

UseCase uc_15_1_ExportEventData "Export event data" : EntitiesInteropExport []

UseCase uc_15_2_FlagEventDataError "Flag event data error" : EntitiesOther [
  actorInitiates aU_Hydro
  actorParticipates aU_Udo
  dataEntity ec_Event
  actions aFlag, aSelect
  extends uc_15_ConsultEventsRecords onExtensionPoint EPFlagEventDataError
  extends uc_16_ConsultOpenEvents onExtensionPoint EPFlagEventDataError]

UseCase uc_16_ConsultOpenEvents "Consult open events" : EntitiesBrowse [
  actorInitiates aU_Hydro
  actorParticipates aU_Udo
  dataEntity ec_Event
  actions aFilter, aRead, aSelect
  extensionPoints EPConsultPhoto, EPFlagEventDataError, EPExportEventData
  description "Consult the events that are occurring"]

```

Specification 31: RSL specification of RCP UseCases: Events.

The About tab is the same for all the users and it has two main divisions, “RiverCure project”, where a user can read about the project and the partners involved and “Feedback”, where a user can give its opinion regarding the portal. Specification 32 states the use cases that a user can do in this tab, such as “uc_20_1_ConsultContact”, that represents the action of a user reading the official email of the portal.

```

UseCase uc_20_ReadRCContext "Read about RiverCure project" : EntityRead [
  actorInitiates aU_User
  dataEntity ec_About
  actions aRead
  extensionPoints EPConsultContact
  description "Read about RiverCure project, the team and the partners."]

UseCase uc_20_1_ConsultContact "Consult contact information" : EntityRead []

UseCase uc_21_GiveFeedback "Give feedback about the RiverCure portal" : EntityCreate [
  actorInitiates aU_User
  dataEntity ec_Feedback
  actions aCreate]

```

Specification 32: RSL specification of RCP UseCases: About.

A user by clicking on the settings can access a secondary panel, that varies according to the user role. A citizen can view its profile information, which includes reading its permissions and editing personal data, and it can also manage its account settings, as it can be seen in Figure 59.

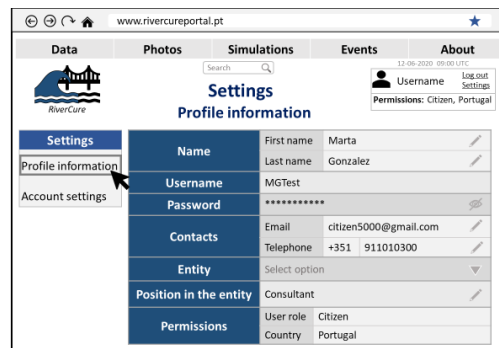


Figure 59: Settings (Profile information) mock-up of the RCP.

From Specification 33 is possible to understand that the technicians have certain use cases regarding the management of the account settings that are specific for them, like “uc_23_4_ViewCitizenPerspective”, which allows the user to view the portal as a citizen would. The “uc_23_5_EditParameterPreferences” and “uc_23_6_EditMonitoringPointsPreferences” use cases permit the technician to select its favourite parameters and monitoring points, which means that the most relevant data on the portal, for this user, is more easily accessible.

```

UseCase uc_22_ManageProfile "Manage profile information" : EntitiesManage [
  actorInitiates aU_User
  dataEntity ec_User_simple
  actions aEdit, aRead, aView, aDelete
  includes uc_22_1_ViewPermissions]

UseCase uc_22_1_ViewPermissions "View permissions" : EntitiesBrowse []

UseCase uc_23_ManageAccountSettings "Manage account settings" : EntitiesManage [
  actorInitiates aU_User
  dataEntity ec_User_simple
  actions aSelect, aRead, aEdit, aDelete
  extensionPoints EPViewCitizenPerspective, EPeditParametersPreferences, EPeditMonitoringPointsPreferences
  includes uc_23_1_EditViewHomePage, uc_23_2_EditNotificationsSettings, uc_23_3_EditPreferences]

UseCase uc_23_1_EditViewHomePage "Edit view of home page" : EntityUpdate []

UseCase uc_23_2_EditNotificationsSettings "Edit notifications settings" : EntityUpdate []

UseCase uc_23_2_1_EditNotificationAlarms "Edit notifications settings of alarms" : EntityUpdate [
  actorInitiates aU_Hydro
  actorParticipates aU_Udo
  dataEntity ec_User_simple
  actions aSelect, aEdit, aDelete, aCreate
  extends uc_23_2_EditNotificationsSettings onExtensionPoint EPDefineNotificationAlarms]

UseCase uc_23_3_EditPreferences "Edit preferences" : EntityUpdate []

UseCase uc_23_4_ViewCitizenPerspective "View in perspective of a citizen" : EntitiesBrowse [
  actorInitiates aU_Technician
  dataEntity ec_User_simple
  actions aSelect
  extends uc_23_ManageAccountSettings onExtensionPoint EPViewCitizenPerspective]

UseCase uc_23_5_EditParametersPreferences "Edit parameters preferences" : EntityUpdate [
  actorInitiates aU_Technician
  dataEntity ec_User_simple
  actions aSelect, aEdit, aDelete, aCreate
  extends uc_23_ManageAccountSettings onExtensionPoint EPeditParametersPreferences]

UseCase uc_23_6_EditMonitoringPointsPreferences "Edit monitoring points preferences" : EntityUpdate [
  actorInitiates aU_Technician
  dataEntity ec_User_simple
  actions aSelect, aEdit, aDelete, aCreate
  extends uc_23_ManageAccountSettings onExtensionPoint EPeditMonitoringPointsPreferences]

```

Specification 33: RSL specification of RCP UseCases: Settings.

Besides the previously mentioned use cases, the hydrometric and udometric technicians are also actors of the “uc_23_2_1_editNotificationAlarms” use case described in Specification 33. These actors can choose to be notified for all red and/or yellow alarms and if they wish to be alerted via SMS or email, as it can be observed in Figure 60.

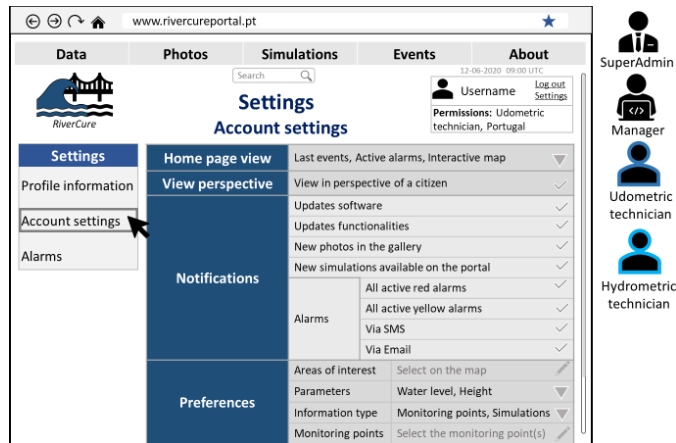


Figure 60: Settings (Account) mock-up of the RCP.

To understand the alarm settings, a mock-up of that interface was created, as it can be seen in Figure 61. This interface is only available for certain user roles, SuperAdmin, manager, udometric and hydrometric technician, since they are the only ones with access to the alarm feature. The portal offers the user two alarm modes, from which the user must select one. The first alarm mode follows the alarm definitions, such as alarm levels and thresholds, set by the authorities, which means that the user cannot edit them. The second mode, also known as “My alarms”, allows the user to specify its alarms, by defining a monitoring point, parameter, threshold and alarm level for each alarm. However, this mode is not available for all users and countries, since the legislation in some countries does not allow it.

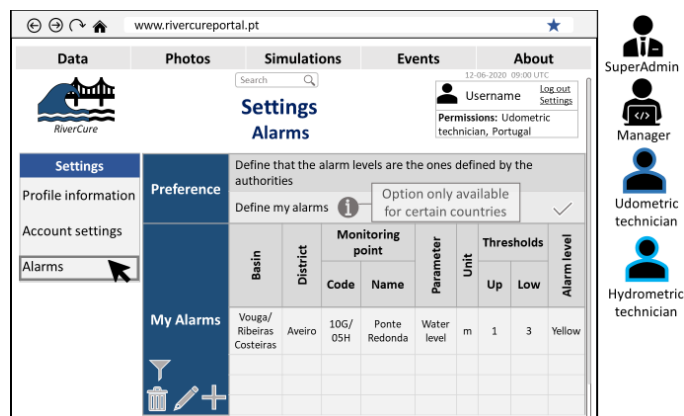


Figure 61: Settings (Alarms) mock-up of the RCP.

Specification 34 defines the use cases demonstrated in the alarms settings mock-up shown in Figure 61. The “uc_19_3_NotifyAlarm” use case is initiated by the actor “aT_NotifyAlarm”, which is a timer that is triggered when an alarm in a monitoring point is activated. “EntitiesInteropSendMessage” is the use case type, since the system sends a notification automatically to the authorized users.

```

UseCase uc_19_ManageAlarms "Manage alarms" : EntitiesManage [
  actorInitiates aU_Hydro
  actorParticipates aU_Udo
  dataEntity ec_Alarm
  actions aAdd, aEdit, aDelete, aRead, aSelect
  extensionPoints EPManageMyAlarms, EPNotifyAlarm
  includes uc_19_1_SelectAlarmPreference]

UseCase uc_19_1_SelectAlarmPreference "Select alarm preference" : EntityOther []

UseCase uc_19_2_ManageMyAlarms "Manage my alarms" : EntitiesManage []

UseCase uc_19_3_NotifyAlarm "Notify user of active alarm" : EntitiesInteropSendMessage [
  actorInitiates aT_NotifyAlarm
  actorParticipates aU_Udo, aU_Hydro, aU_Manager
  dataEntity ec_User_Role
  actions aCreate, aFilter, aSelect, aDelete, aEdit
  extends uc_19_ManageAlarms onExtensionPoint EPNotifyAlarm]

```

Specification 34: RSL specification of RCP UseCases: Alarms.

The manager is responsible for managing the information, as observed in Figure 62, which can include a variety of use cases, such as managing the monitoring points and radar observations. This actor has access to all the data flagged by other users, which means that “View flagged data”, “View records flagged” and “View list of flagged photos” are use cases that can be executed by the manager. Besides managing all the data regarding the monitoring points, radar observations, events and simulations, the user can view the portal analytics and the user accounts logs.

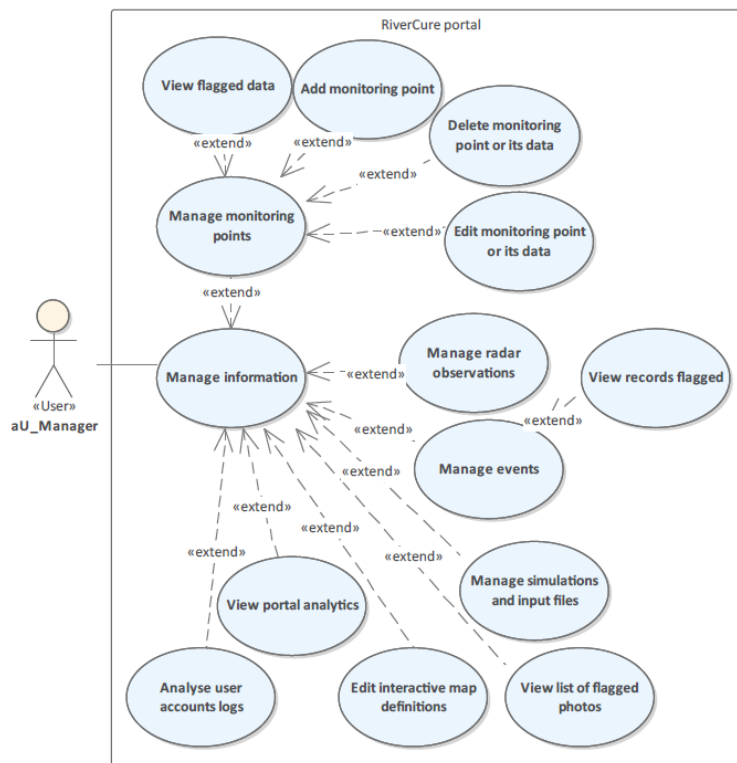


Figure 62: UML diagram of the RCP UseCases: Manager package.

The use case “Manage events” represented in Figure 62, includes several actions, creating, editing, deleting and viewing the records flagged, as it is demonstrated in Figure 63.

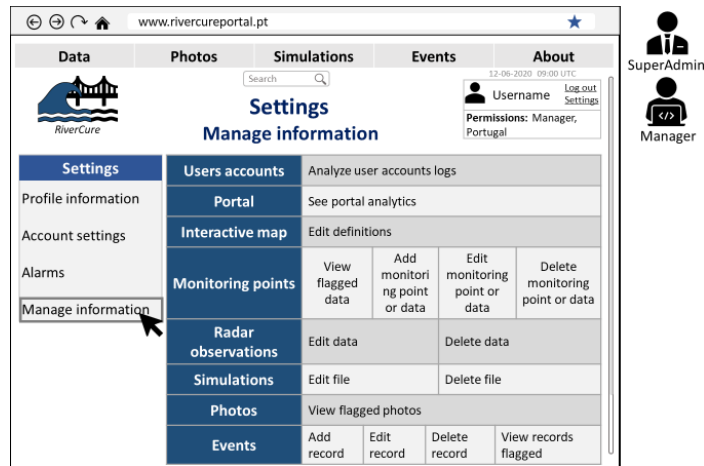


Figure 63: Settings (Manage information) mock-up of the RCP.

The use cases shown in Specification 35 have the actor “aU_SuperAdmin” as the primary actor. This actor is responsible for managing users, entities (e.g. companies, universities), the communication with the partners and the portal’s updates, such as the model of the photos (uc_28_2_UpdateModelPhotos). The “uc_25_ManageUsers” has the extension point “EPValidateAccounts” and includes two other use cases “uc_25_2_ManageUserRoles” (which allows this user to create, edit and delete user roles) and “uc_25_3_ManagePermissions” (which allows to edit the permissions associated with each user and user role). When an account is not validated after 24h (“aT_AccountNotValidated”) the SuperAdmin receives a notification automatically from the system.

```

UseCase uc_25_ManageUsers "Manage users" : EntitiesManage [
  actorInitiates aU_SuperAdmin
  dataEntity ec_User_simple
  actions aAdd, aEdit, aDelete, aValidate
  extensionPoints EPValidateAccounts
  includes uc_25_2_ManageUserRoles, uc_25_3_ManagePermissions]

UseCase uc_25_1_ValidateAccounts "Validate accounts" : EntitiesOther []

UseCase uc_25_1_1_NotifyAccountValidation "Notify account for validation" : EntitiesOther [
  actorInitiates aT_AccountNotValidated
  actorParticipates aU_SuperAdmin
  dataEntity ec_User_simple
  actions aNotification
  extends uc_25_1_ValidateAccounts onExtensionPoint EPNotifyAccountValidation]

UseCase uc_25_2_ManageUserRoles "Manage user roles" : EntitiesManage []

UseCase uc_25_3_ManagePermissions "Manage permissions" : EntitiesManage []

UseCase uc_26_ManagePartners "Manage communication with partners" : EntitiesManage []

UseCase uc_27_ManageEntities "Manage entities" : EntitiesManage []

UseCase uc_27_1_ViewEntitiesList "View entities list" : EntityRead []

UseCase uc_28_ManageUpdates "Manage updates" : EntitiesManage [
  actorInitiates aU_SuperAdmin
  dataEntity ec_UserInformation
  actions aEdit, aDelete, aUpdate
  extensionPoints EPUUpdateServerMap, EPUUpdateModelPhotos, EPUUpdateSimulatorExecutable,
  EPEditImportationParameters, EPEditProjectInformation, EPEditFeedbackForm, EPEditPortalCode]

UseCase uc_28_1_UpdateServerMap "Update server code of the map" : EntityUpdate []

UseCase uc_28_2_UpdateModelPhotos "Update model of the photos" : EntityUpdate []

UseCase uc_28_3_UpdateSimulatorExecutable "Update simulator executable" : EntityUpdate []

UseCase uc_28_4_EditImportationParameters "Edit data importation parameters" : EntityUpdate []

UseCase uc_28_5_EditProjectInformation "Edit project information" : EntityUpdate []

UseCase uc_28_6_EditFeedbackForm "Edit feedback form" : EntityUpdate []

UseCase uc_28_7_EditPortalCode "Edit portal code" : EntityUpdate []

```

Specification 35: RSL specification of RCP UseCases: SuperAdmin.

6. Conclusions and Future Work

This Chapter presents the conclusion of the research (section 6.1), namely the answers to the original questions introduced in section 1.3. Furthermore, it adds points to focus on the future (section 6.2).

6.1. Conclusions

The quantity and diversity of data sources, webpages and tools regarding water resources available made the scientific community more aware of the importance of proper data management. To improve this task a greater effort has been made in the specifying more accurately the requirements of some of those systems.

This work defines the requirements specification of the RiverCure portal, a web platform that intends to integrate data from different entities, such as official agencies and flood photos with interactive maps and a flood simulation tool. With the extensive data resources and capabilities, this portal can bring many benefits such as improving the flood forecasting and the decision-making process. By providing a user-friendly system, stakeholders can make smarter decisions. It can also support the development of plans to prevent and mitigate the impacts of floods. Consequently, it can improve the time and quality of the authority's responses during a flood, since the allocation of the means of assistance would be done more effectively. The fact that a single platform can integrate and present data from several sources it shall allow to better manage water resources.

To reach a proper outcome is important to answer the question originally introduced in Chapter 1.

First, *Which are the requirements necessary to define the RiverCure Portal?* Since this thesis used a use case approach to specify requirements, it was necessary to define the use cases and all the constructs associated to the use cases that are relevant to these information systems, namely, actors and data entities.

Second, *How should the specifications be represented?* Since this project has different stakeholders with distinct backgrounds, it was necessary to represent the specifications graphically, by using UML and mock-ups, in the first part of the project, so that the people involved could understand what was being proposed. In the second phase of the project, RSL was used to specify the systems which allowed to define the specifications more rigorously, namely by adding certain details (e.g. actions associated). It also helped avoid miscommunications with the stakeholders, inconsistencies and ambiguities.

Third, *Is the RSL language an adequate approach to define structured and rigorous requirements in the water domain?* Yes, RSL can be used in a water domain, since it allows different levels of detail, with different concerns and different types of requirements. However, it is important to understand, that specialists in the water domain might not be familiarized with RSL, or with other similar languages which can make the development process more complex since they need to learn how this tool works.

The RiverCure portal has features that were presented in the systems explained in Chapter 3.1. It displays data from monitoring stations, which also occurs in the Sava GIS Geoportal and the IFIS. It provides flood simulations, that it is also a feature in IFIS and HEC-HMS. It also uses an interactive map to show the portal data (e.g. monitoring points, photos, simulations), which is also used in the Some Mare FIS Portal, Sava GIS Portal and IFIS. The Some Mare FIS Portal, which uses the WaterML 2.0-GeoServer, has another feature in common with the RiverCure portal, it allows citizens to access and contribute to the portal.

6.2. Future Work

The future work should focus on creating the portal through the implementation of the system requirements preliminary defined in this dissertation. However, before implementing it is important to better standardize all the details of the system according to WaterML 2.0. After the implementation, the RCP must be tested with a specific dataset, by a group of controlled users. To evaluate the system in terms of features and usability, a survey must be sent to all actors.

To acquire more relevant photos, it is necessary to develop a plan to encourage the community to participate in this initiative. With the public help, more photos can be accessed making the estimated water height of a photo will be more accurate since the social flood model would have more data to perfect the system. More authentic simulations can be made using photos as inputs since the model would have more specific information regarding the area that is being studied.

Other developments, such as the creation of a flood risk index for infrastructures, with several levels (e.g. low, medium, critical), that would take into account the consequences and the probability of a flood occurring, can improve this portal, and consequently help the decision-maker to have more confidence in its choice. Adding a field that states the maintenance history of the equipment in the monitoring points, can also be an important development since it can help to decipher an error source.

Since one of the goals is to make this portal available for other countries, some data entities of the portal must be adapted. For instance, not all countries call their first sub-division district some defined it as a region. So, a standard should be used, the ISO 3166 defines the codes for countries and codes for subdivisions. Each subdivision has a name, a code and a category associated [72]. To achieve this goal and for the portal to be more accessible to the public, more options of idioms should be available.

For managing the alerts, a future improvement would be to develop an application based on the portal, so that the authorized users may receive push notifications with relevant information.

With all these improvements the expansion of the portal to other countries might be simpler, the stakeholders may be more satisfied with the portal, the decision-makers will make even more informed decisions, the population may join more easily and consequently, more information may be available especially photos.

References

- [1] Agência Portuguesa do Ambiente, “Avaliação Preliminar dos Riscos de Inundações em Portugal Continental,” Relatório Técnico, 2018.
- [2] T. Miksa, J. Cardoso, and J. Borbinha, “Framing the scope of the common data model for machine-actionable Data Management Plans,” in *Proceedings of the 2018 IEEE International Conference on Big Data*, 2018.
- [3] J. S. Horsburgh, D. G. Tarboton, D. R. Maidment, and I. Zaslavsky, “A relational model for environmental and water resources data,” *Water Resour. Res. Wiley*, vol. 44, no. 5, May 2008.
- [4] D. A. S. Conde, M. A. V. Baptista, C. Sousa Oliveira, and R. M. L. Ferreira, “A shallow-flow model for the propagation of tsunamis over complex geometries and mobile beds,” *Nat. Hazards Earth Syst. Sci. Copernicus Publ.*, vol. 13, no. 10, 2013.
- [5] A. R. Silva, “Rigorous Requirements Specification with the RSL Language: Focus on Uses Cases,” in *Proceedings of ISD’ 2019, AIS*, 2019.
- [6] A. R. Silva, “Linguistic Patterns and Linguistic Styles for Requirements Specification (I): An Application Case with the Rigorous RSL/Business-Level Language,” in *Proceedings of the EuroPLOP’2017, ACM*, 2017.
- [7] P. Taylor, “OGC WaterML 2.0: Part 1- Timeseries,” *Measurement*, vol. 2007, no. 05, 2011.
- [8] “Open Water Foundation - WaterML 2 Overview.” [Online]. Available: <http://learn.openwaterfoundation.org/owf-learn-waterml2/waterml2-overview/>. [Accessed: 14-May-2019].
- [9] A. R. Silva and C. Videira, *UML, Metodologias e Ferramentas CASE*, 2nd ed., vol. 1. Centro Atlântico Editora, 2005.
- [10] M. P. da Custódia Conceição, “RSLingo-Studio: A tool for Rigorous Requirements Specification, Validation and Transformation,” MSc Dissertation, Instituto Superior Técnico, Universidade de Lisboa, 2017.
- [11] M. Saramago, “Hydrologic Surveillance System Using Wireless Technologies,” in *Proceedings of 4th International Conference on Experiences with Automatic Weather Stations (ICEAWS)*, 2006.
- [12] “INESC-ID RiverCure project.” [Online]. Available: <https://www.inesc-id.pt/projects/II11041/>. [Accessed: 29-Apr-2019].
- [13] K. Peffers, T. Tuunanen, M. A. Rothenberger, and S. Chatterjee, “A Design Science Research Methodology for Information Systems Research,” *J. Manag. Inf. Syst.*, vol. 24, no. 3, 2007.
- [14] A. Almoradie, I. Popescu, A. Jonoski, and D. Solomatine, “Web Based Access to Water Related

- Data Using OGC WaterML 2.0,” *Int. J. Adv. Comput. Sci. Appl.*, vol. 3, no. Building a Regional Observation System in the Black Sea Catchment, 2013.
- [15] A. Alanazi, A. B. Jones, and J. Straub, “Requirements Modeling Language and Automated Testing for CubeSats,” in *Proceedings of 2019 IEEE AUTOTESTCON, IEEE*, 2019.
- [16] “SNIRH - Ponte Águeda (Nível hidrométrico instantâneo).” [Online]. Available: https://snirh.apambiente.pt/snirh/_dadosbase/site/janela_verdados.php?sites=1627758988¶=1850,1843&tmin=11/02/2016&tmax=13/02/2016. [Accessed: 04-Jun-2019].
- [17] “C. M. Águeda - Recursos Hídricos.” [Online]. Available: <https://www.cm-agueda.pt/pages/522>. [Accessed: 04-Jun-2019].
- [18] S. Mourato, M. Moreira, P. Fernandez, L. Pereira, M. Jesus, and F. Marques, “Flood mapping under uncertainty: a case study in Águeda, Portugal,” in *Proceedings of 2nd International Conference on Natural Hazards & Infrastructure (ICONHIC2019)*, 2019, pp. 23–26.
- [19] Câmara Municipal de Águeda, “Flood Águeda River 12/02/2016,” 2016. .
- [20] J. Rüegg *et al.*, “Completing the data life cycle: Using information management in macrosystems ecology research,” *Front. Ecol. Environ. Ecol. Soc. Am.*, vol. 12, no. 1, pp. 24–30, 2014.
- [21] “SNIRH - MEDiateca.” [Online]. Available: <https://snirh.apambiente.pt/index.php?idMain=5&idItem=2>. [Accessed: 09-Dec-2019].
- [22] “OGC WaterML 2: Part 3 - Surface Hydrology Features - Conceptual Model,” 2018.
- [23] “Sistema Nacional de Informação sobre Recursos Hídricos (SNIRH).” [Online]. Available: <https://snirh.apambiente.pt>. [Accessed: 13-May-2019].
- [24] R. Gomes, M. Saramago, and R. Rodrigues, “SVARH - Sistema de Vigilância e Alerta de Recursos Hídricos,” Relatório Técnico, Instituto da Água, 2003.
- [25] M. Saramago, “Redes de Monitorização Hidrometeorológicas,” *Rev. Recur. Hídricos, Assoc. Port. dos Recur. Hídricos*, vol. 38, no. 1, 2017.
- [26] F. Quadrado, “SISTEMA DE VIGILÂNCIA E ALERTA DE RECURSOS HÍDRICOS - SVARH,” Presented in the Green Business Week, Agência Portuguesa do Ambiente, 2016.
- [27] N. M. P. Charneca, “Modelação de Dados Geográficos Aplicada ao Planeamento e Gestão de Recursos Hídricos,” PhD Dissertation, Faculdade de Ciências, Universidade de Lisboa, 2012.
- [28] I. Zaslavsky, D. Valentine, and T. Whiteaker, “CUAHSI WaterML,” Open Geospatial Consortium Discussion Paper 07-041r1, 2007.
- [29] “ISO 19101-1:2014(en), Geographic information — Reference model — Part 1: Fundamentals.” [Online]. Available: <https://www.iso.org/obp/ui/#iso:std:iso:19101:-1:ed-1:v1:en>. [Accessed: 29-

Mar-2020].

- [30] “What is a layer? - ArcGIS for Desktop.” [Online]. Available: <https://desktop.arcgis.com/en/arcmap/10.3/map/working-with-layers/what-is-a-layer-.htm>. [Accessed: 29-Mar-2020].
- [31] “Raster Data — QGISDoc documentation.” [Online]. Available: https://docs.qgis.org/3.4/en/docs/gentle_gis_introduction/raster_data.html#overview. [Accessed: 30-Mar-2020].
- [32] “Introducing GIS — QGISDoc documentation.” [Online]. Available: https://docs.qgis.org/3.4/en/docs/gentle_gis_introduction/introducing_gis.html#gis-data. [Accessed: 29-Mar-2020].
- [33] Environmental Systems Research Institute, “ESRI Shapefile Technical Description - An ESRI White Paper,” Technical paper, 1998.
- [34] “Working with Mesh Data — QGISDoc documentation.” [Online]. Available: https://docs.qgis.org/3.4/en/docs/user_manual/working_with_mesh/mesh_properties.html. [Accessed: 29-Mar-2020].
- [35] D. A. S. Conde, M. J. Telhado, M. A. Viana Baptista, and R. M. L. Ferreira, “Severity and exposure associated with tsunami actions in urban waterfronts: the case of Lisbon, Portugal,” *Nat. Hazards*, vol. 79, no. 3, 2015.
- [36] D. A. S. Conde, “High-Performance Modelling of Tsunami Impacts on Built Environments,” PhD Thesis, Universidade de Lisboa, Instituto Superior Técnico, 2018.
- [37] D. A. S. Conde, “A Tsunami in Lisbon Where to run ?,” MSc Thesis, Instituto Superior Técnico, Universidade de Lisboa, 2012.
- [38] D. A. S. Conde, R. Canelas, J. Murillo, and R. M. L. Ferreira, “Simulating the 1755 tsunami propagation in present-day Lisbon with a shallow-water model,” *Rev. Recur. Hídricos, Assoc. Port. dos Recur. Hídricos*, vol. 33, no. 2, 2012.
- [39] B. C. Almeida, A. P. M. Saliba, and D. A. S. Conde, “Retroanálise da propagação decorrente da ruptura da barragem do Fundão através do modelo numérico HISTAV,” in *Proceedings of the III SIAAR - Simpósio Íbero-Afro-Americano de Riscos*, 2019, vol. 3, no. June.
- [40] C. Pohl, Klaus; Rupp, *Requirements Engineering Fundamentals: a study guide for the certified professional for requirements engineering exam, foundation level*, 2nd ed. Rocky Nook, 2015.
- [41] OMG, “OMG Unified Modeling Language (OMG UML).” OMG, 2017.
- [42] “System Boundary.” [Online]. Available: https://sparxsystems.com/enterprise_architect_user_guide/14.0/model_domains/systemboundary.html. [Accessed: 24-May-2019].

- [43] “Class Diagram.” [Online]. Available: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-class-diagram/>. [Accessed: 24-May-2019].
- [44] “Enumerations.” [Online]. Available: https://www.ibm.com/support/knowledgecenter/SS8PJ7_9.7.0/com.ibm.xtools.modeler.doc/topics/cenum.html. [Accessed: 05-Jan-2020].
- [45] “UML Association vs Aggregation vs Composition.” [Online]. Available: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/uml-aggregation-vs-composition/>. [Accessed: 29-May-2020].
- [46] C. Videira, J. L. Carmo, and A. R. Silva, “The ProjectIT-RSL Language Overview,” in *UML Modeling Languages and Applications: UML 2004 Satellite Activities*, 2004.
- [47] C. Videira and A. R. Silva, “ProjectIT-Requirements, a Formal and User-oriented Approach to Requirements Specification,” in *Actas de las IV Jornadas Iberoamericanas en Ingeniería del Software e Ingeniería del Conocimiento*, 2004, vol. 1, pp. 175–190.
- [48] J. Caramujo, A. R. Silva, S. Monfared, A. Ribeiro, P. Calado, and T. Breaux, “RSL-IL4Privacy: A Domain-Specific Language for the Rigorous Specification of Privacy Policies,” in *Requirements Engineering, Springer*, 2019, vol. 24, no. 1.
- [49] D. De Almeida Ferreira and A. R. Silva, “RSL-PL: A Linguistic Pattern Language for Documenting Software Requirements,” in *Proceedings of Third International Workshop on Requirements Patterns (RePa’ 13), in the 21st IEEE International Requirements Engineering Conference (RE’2013), IEEE Computer Society*, 2013.
- [50] L. P. Gonçalves and A. R. Silva, “Towards a Catalogue of Reusable Security Requirements, Risks and Vulnerabilities,” in *Proceedings of ISD’ 2018, AIS*, 2018.
- [51] M. Conceição, “RSLingo-Studio: User Guide.” 2017.
- [52] D. de Almeida Ferreira and A. R. Silva, “RSLingo: An Information Extraction Approach toward Formal Requirements Specifications,” in *Proceedings of 2nd IEEE International Workshop on Model-Driven Requirements Engineering (MoDRE), IEEE Computer Society*, 2012.
- [53] “Logical Data Model.” [Online]. Available: <https://www.1keydata.com/datawarehousing/logical-data-model.html>. [Accessed: 29-May-2020].
- [54] “International Sava River Basin Commission - Sava GIS Metadata Catalogue.” [Online]. Available: <http://savagis.org/metadatalogue/srv/eng/search#fast=index&from=1&to=50>. [Accessed: 12-Jun-2020].
- [55] International Sava River Basin Commission, “Metadata Catalogue Public User Manual.” 2016.
- [56] International Sava River Basin Commission, “Data and Information Exchange - Sava GIS & Sava HIS,” Presented for the Western Balkans Investment Framework.

- [57] I. Demir and W. F. Krajewski, "Towards an integrated Flood Information System: Centralized data access, analysis, and visualization," *Environ. Model. Softw.*, vol. 50, Dec. 2013.
- [58] I. Demir, E. Yildirim, Y. Sermet, and M. A. Sit, "FLOODSS: Iowa flood information system as a generalized flood cyberinfrastructure," *Int. J. River Basin Manag.*, vol. 16, no. 3, Jul. 2018.
- [59] W. F. Krajewski *et al.*, "Real-Time Flood Forecasting and Information System for the State of Iowa," *Bull. Am. Meteorol. Soc.*, vol. 98, no. 3, Mar. 2017.
- [60] D. Ford, N. Pingel, and J. J. DeVries, "Hydrologic Modeling System HEC-HMS: Applications Guide." U.S. Army Corps of Engineers Hydrologic Engineering Center, Washington, 2008.
- [61] "HEC-DSS." [Online]. Available: <https://www.hec.usace.army.mil/software/hec-dss/>. [Accessed: 13-Jun-2019].
- [62] "CUAHSI-HIS." [Online]. Available: <http://his.cuahsi.org/index.html>. [Accessed: 15-Jun-2019].
- [63] A. M. Abdallah and D. E. Rosenberg, "A data model to manage data for water resources systems modeling," *Environ. Model. Software, Elsevier*, vol. 115, 2019.
- [64] D. P. Ames, J. S. Horsburgh, Y. Cao, J. Kadlec, T. Whiteaker, and D. Valentine, "HydroDesktop: Web services-based software for hydrologic data discovery, download, visualization, and analysis," *Environ. Model. Software, Elsevier*, vol. 37, 2012.
- [65] OMG, "OMG Systems Modeling Language (OMG SysML™)," 2019.
- [66] M. Hause, "The OMG Modelling Language (SysML)," in *Proceedings of the Fifth European Systems Engineering Conference, INCOSE*, 2006.
- [67] A. Chen and J. Beatty, *Visual Models for Software Requirements*, 2nd ed. Microsoft Press, 2012.
- [68] L. Gonçalves and A. R. Silva, "A catalogue of reusable security concerns: Focus on privacy threats," in *Proceedings of IEEE CBI ' 2018, IEEE*, 2018, vol. 2, pp. 52–61.
- [69] R. Rodrigues, "Flood Surveillance and Early Warning System for Portugal," *Presentation for the Precautionary Flood Protection in Europe - International Workshop*. Bonn, 2003.
- [70] L. H. Broska, W. R. Pogonietz, and S. Vögele, "Extreme events defined—A conceptual discussion applying a complex systems approach," *Futures*, vol. 115, Jan. 2020.
- [71] L. E. McPhillips *et al.*, "Defining Extreme Events: A Cross-Disciplinary Review," *Earth's Futur.*, vol. 6, no. 3, pp. 441–455, Mar. 2018.
- [72] "ISO 3166 Country Codes." [Online]. Available: <https://www.iso.org/iso-3166-country-codes.html>. [Accessed: 04-Feb-2020].

Annexes

SNIRH

Term trm_user "User" : Noun [applicableTo Stakeholder, Actor description

"A user of the system is someone that has an user account and a password associated with it.

Depending on that user profile he has permission to do different functionalities.

There are the following types of user profiles:

- Restricted User, can access certain information of the system (has restricted access);
- System Administrator, responsible for managing users, managing stations, managing the data base."]

Term trm_res_user "Restricted user" : Noun [isA trm_user applicableTo Stakeholder, Actor description

"User that has restricted access to the system, can only view data of certain locations or certain parts of the system;

The permission to check specific data is defined by the system administrator.

This user can have writing permissions, which means that they can save data in their computer.

E.g. Regional environmental authorities."]

Term trm_sys_admin "System administrator" : Noun [isA trm_user applicableTo Actor, Stakeholder description

"User that belongs to APA's water resources team,that has the responsibility to:

- Manage users accounts (e.g. Adding new users, changing users' permissions in the system),
- Manage the stations (e.g. adding / eliminating stations, changing data's collection frequency),
- Update the terminology list,
- Manage the data base system (e.g. uploads data),
- Validate the data from the stations."]

Term trm_p_viewer "Public viewer" : Noun [applicableTo Stakeholder, Actor description

"A public viewer is someone who visits the SNIRH's portal, accessing the data available online public to the everyone."]

Specification 36: Partial RSL specification of SNIRH glossary.

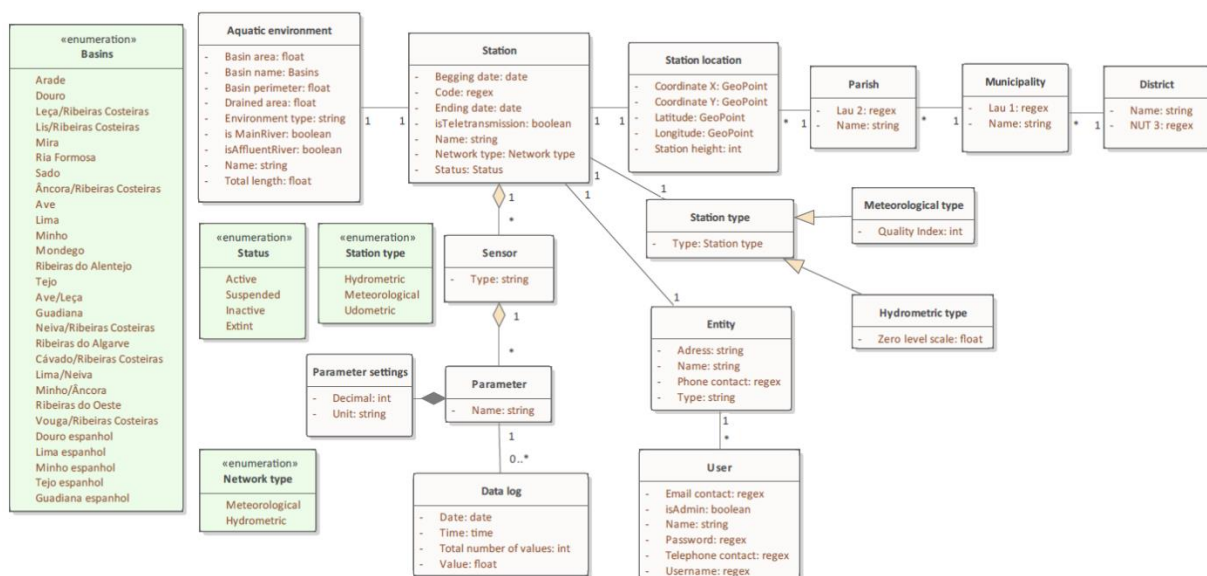


Figure 64: UML diagram of all SNIRH DataEntities.

```

DataEntity e_S_Location "Station location" : Document : Detail []
DataEntity e_Parish "Parish" : Reference []
DataEntity e_Municipality "Municipality" : Reference []
DataEntity e_District "District" : Reference []
DataEntity e_Sensor "Sensor" : Master []
DataEntity e_Parameter "Parameter" : Master []
DataEntity e_P_Settings "Parameter settings" : Reference []
DataEntity e_Data "Data log": Document []
DataEntity e_S_Type "Station type" : Reference []
DataEnumeration de_SType "Station type" : Simple [values "Hydrometric", "Meteorological", "Udometric"]
DataEntity e_HydroType "Hydrometric type" : Reference []
DataEntity e_MeteoType "Meteorological type" : Reference []
DataEntity e_Entity "Entity" : Parameter [
  attribute entity_type "Entity type" : String [isNotNull]
  attribute phone "Phone contact" : Regex [isNotNull]
  attribute address "Address" : Regex
  attribute Name "Entity name" : String [isNotNull isUnique]
  primaryKey (Name)]
DataEntity e_User "User" : Parameter [
  attribute Username "Username" : String [isNotNull isUnique]
  attribute email "Email contact" : Regex [isNotNull]
  attribute password : Regex [isNotNull isEncrypted]
  attribute Name "Name" : String [isNotNull]
  attribute telephone "Telephone contact" : Regex [isNotNull]
  attribute isAdmin "Administrator" : Boolean
  attribute entity_name "Entity name" : String [isNotNull isUnique]
  primaryKey (Username)
  foreignKey e_Entity (entity_name)]

```

Specification 37: Partial RSL specification of SNIRH DataEntities.

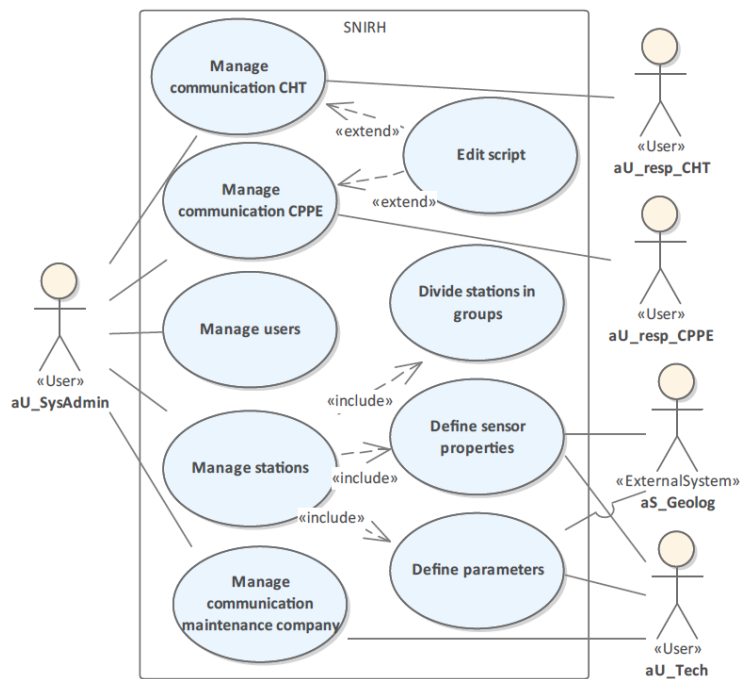


Figure 65: UML diagram of SNIRH UseCases: Manage package.

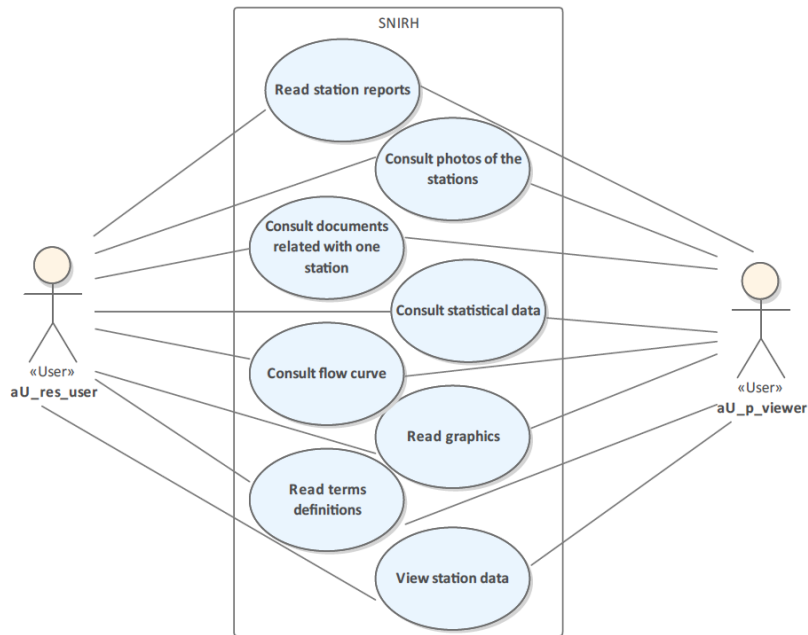


Figure 66: UML diagram of the SNIRH UseCases: Restricted user and Public viewer package.

SVARH

Term trm_station "Station" : Noun [applicableTo DataEntity synonym "monitoring point, site" description
 "Station is a place that allows to make certain observations/measurements through a set of sensors, which measure different parameters. The stations are composed by a datalogger, solar panel, sensors, battery, communication device (e.g.GSM modem). To define these stations the following data is necessary: Name, Station code, Basin, Location. Complementary information available: Responsible Entity, Type of equipment, Start and end functioning date, Functioning state, Teletransmission."]

Term trm_sensor "Sensors" : Noun [applicableTo DataEntity description
 "Instrument that measures a specific parameter or parameters, recording the data captured. E.g. Udometer (instrument that measures precipitation)."]

Term trm_s_parameter "Station parameters" : Noun [applicableTo DataEntity description
 "A variable and measurable property, which the value determines the systems' characteristics. Types of parameters: Flow (Downstream, Upstream, Instantaneous), Water level, Discharges, Volume, Evaporation, Precipitation, Radiation, Air temperature, Wind velocity and direction, Relative humidity."]

Term trm_monitoring "Monitoring" : Verb [applicableTo DataEntity description
 "Measurement and observation of natural phenomenon, which can be continuous or periodic, used several times for control and prevention."]

Term trm_HydroStation "Hydrometric station" : Noun [applicableTo DataEntity description
 "Station where data from rivers, lakes or reservoirs can be obtained. Namely, the water level, flow, transport and deposit of sediments, temperature and other physical, chemical and biological properties of the water."]

Term trm_MeteoStation "Meteorological station" : Noun [applicableTo DataEntity description
 "Station from which meteorological data can be obtained, such as, precipitation, temperature, air humidity, wind velocity and others. If the station only measures precipitation it is called udometric or udographic. If it also measures air and wind properties it is called climatological."]

Term trm_PReturn "Return period" : Noun [applicableTo DataEntity description
 "The average time range a long term or the number of years that separates an event with a known dimension from other with the same or higher dimension."]

Term trm_Telemetry "Telemetry" : Noun [applicableTo DataEntity description
 "Record of distance data through measurement instruments."]

Term trm_flood "Flood" : Noun [applicableTo DataEntity description
 "Significant rise of the level of water in a water course, lake, reservoir or coastal area."]

Specification 38: Partial RSL specification of SVARH glossary.


```

DataEntity e_Station "Station" : Master [
  attribute next_s_up "Next station name upstream" : String [isUnique]
  attribute next_s_down "Next station name downstream" : String [isUnique]
  attribute Name "Station name" : String (30) [isNotNull isUnique]
  attribute code "Code" : Regex [isNotNull isUnique]
  attribute State "State" : DataEnumeration de_State [isNotNull]
  attribute Muni_Name "Municipality name" : String [isNotNull isUnique]
  attribute Basin_Name "Basin name" : DataEnumeration de_Basins [isNotNull]
  attribute user_Description "User description" : String [isNotNull description "e.g. Intern Marta Gonzalez"]
  primaryKey (code)
  foreignKey e_Municipality (Muni_Name)
  foreignKey e_Basin (Basin_Name)
  foreignKey e_User (user_Description)]

DataEnumeration de_State "State" : Simple [values "OnLine", "OffLine"]

DataEntity e_StationType "Station type" : Reference []

DataEnumeration de_StationType "Station type" : Simple [values "Dam", "Hydrometric", "Meteorological", "Udometric"]

DataEnumeration de_InfoType "Information type" : Simple [values "Copy", "Original"]

DataEntity e_Sensor "Sensor" : Master [
  attribute station_code "Station code" : Regex [isNotNull isUnique]
  attribute type "Sensor type" : String [isNotNull description "Name of the sensor"]
  attribute State "Alarm state" : DataEnumeration de_AlarmColor [isNotNull]
  primaryKey (station_code, type)
  foreignKey e_Station (station_code)
  foreignKey e_AlarmSet (State)]

DataEntity e_HydroSensor "Hydrometric sensor" : Reference [
  isA e_Sensor
  attribute zero_scale "Zero level scale" : Decimal ]

DataEntity e_Sensor_S "Sensor settings" : Reference []

DataEntity e_Data "Data log": Document [
  attribute sensor_type "Sensor type" : String [isNotNull description "Name of the sensor"]
  attribute date "Date" : Date [isNotNull]
  attribute time "Time" : Time [isNotNull]
  attribute t_values "Total number of values" : Integer [isNotNull]
  primaryKey (sensor_type, date, time)
  foreignKey e_Sensor (sensor_type)]

DataEntity e_DamData "Dam data log" : Reference []

DataEntity e_HydrometricData "Hydrometric data log" : Reference []

DataEntity e_MeteorologicalData "Meteorological data log" : Reference []

```

Specification 39: RSL specification of SVARH DataEntities: Station.



Figure 67: UML diagram of SVARH DataEntities.

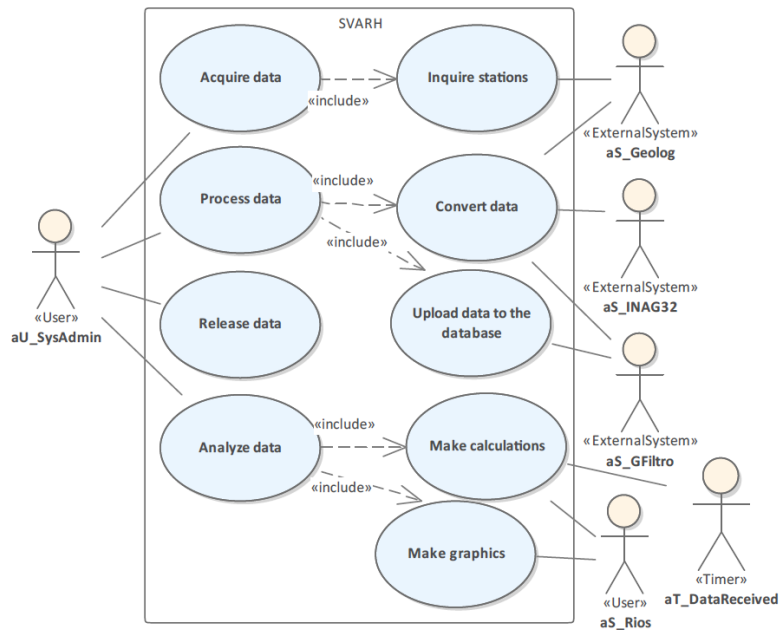


Figure 68: UML diagram of SVARH UseCases: Data package.

HiSTAV

Term trm_user "User" : Noun [applicableTo Stakeholder, Actor description
 "A user of the system is someone that has access to the tool and it is able to run simulation;
 There is a special type of user, the System administrator, that it is responsible for updating the executable file."
Term trm_sys_admin "System administrator" : Noun [applicableTo Stakeholder, Actor synonym "Power user, Technical administrator" description
 "User that belongs to the HiStav team, that has the responsibility to update the executable file, specifically the operations defined in it."

Term trm_HiStav "HiStav" : Noun [applicableTo Actor, Stakeholder description
 "Simulation tool applied, to check the consequences in land of a flood.
 It is based on the shallow-water equations, featuring dynamic bed geometries and sediment transport."
Term trm_Pre_processor "Pre processor" : Noun [applicableTo Actor, Stakeholder, System description
 "Responsible for processing the input data (e.g. DTM, Domain). It combines and converts all the input data to a new format.
 It contains a mesh maker tool that is responsible to produce the mesh."
Term trm_QGIS "QGIS" : Noun [applicableTo Actor, Stakeholder, System description
 "Geographic information system (free and open-sourced) that supports viewing, editing, and analysis of geospatial data."
Term trm_ParaView "ParaView" : Noun [applicableTo Actor, Stakeholder, System description
 "Software used to visualize the results obtain from the simulation tool (HiStav)."
Term trm_raster_data "Raster data" : Noun [applicableTo DataEntity description
 "Rasters are made up of a matrix of pixels, each containing a value that represents the conditions for the area covered by that cell.
 A raster dataset is composed of rows (running across) and columns (running down) of pixels (also know as cells)."
Term trm_DTM "Digital Terrain Model" : Noun [acronym "DTM" applicableTo DataEntity description
 "A DTM is a mathematical representation (model) of the ground surface, most often in the form of a regular grid,
 in which a unique elevation value is assigned to each pixel. DTMs represent the bare ground of the terrain."

Term trm_vector_data "Vector data" : Noun [applicableTo DataEntity description
 "Stored as a series of X, Y coordinate pairs inside the computer's memory.
 Vector data is uses points(e.g. towns), lines (e.g. rivers) and polygons (e.g. municipal boundaries) to represent data."

Specification 40: Partial RSL specification of HiStav glossary.

```
Actor aU_User "User" : User [stakeholder HiStav_Business.stk_user]
Actor aS_QGIS "QGIS" : ExternalSystem [stakeholder HiStav_Business.stk_QGIS]
```

Specification 41: Pre-processor actors in RSL.

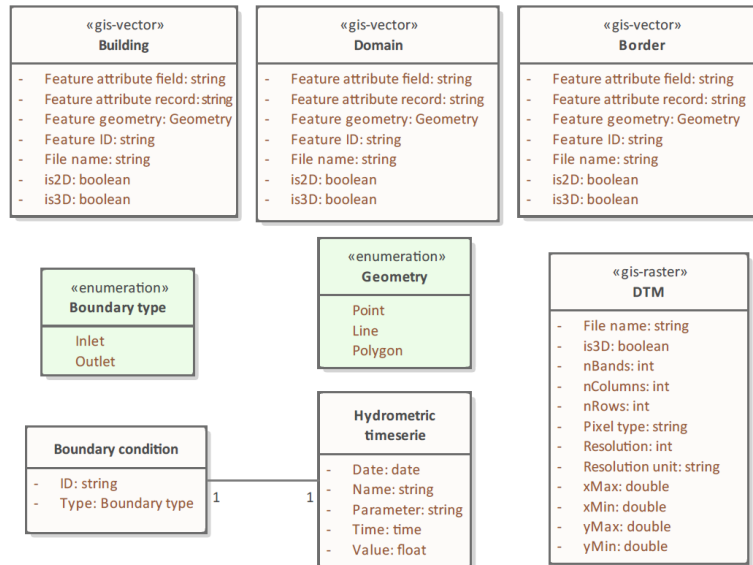


Figure 69: UML diagram of the pre-processor inputs.

```

DataEntity e_BoundaryCond "Boundary condition" : Master [
  attribute serie "Timeserie name" : String [isNotNull]
  attribute type "Type" : DataEnumeration de_BoundaryType [isNotNull]
  attribute ID "ID" : String [isUnique]
  primaryKey (ID)
  foreignKey e_Timeserie (serie) ]

DataEnumeration de_BoundaryType "Boundary type" : Simple [values "Inlet", "Outlet"]

DataEntity e_Timeserie "Hydrometric timeserie" : Document : Detail [
  attribute Name "Timeserie name" : String [isNotNull isUnique]
  attribute time "Time" : Time
  attribute Value "Value" : Decimal
  attribute parameter "Parameter" : String [isNotNull]
  primaryKey (Name)]

DataEntity e_InitialCond "Initial condition" : Master [
  attribute altName "Altimetry name" : String [isNotNull isUnique]
  attribute resName "Resistance name" : String [isNotNull isUnique]
  attribute AltValue "Altimetry value" : Decimal
  attribute ResValue "Resistance value" : Decimal
  primaryKey (altName, resName)]

DataEntity e_Mesh "Mesh" : Master [
  attribute Name "Name" : String [isNotNull isUnique]
  attribute is2D "is 2Dimensions" : Boolean
  attribute is3D "is 3Dimensions" : Boolean
  attribute nElements "nElements" : Integer [isNotNull]
  attribute nNodes "nNodes" : Integer [isNotNull]
  attribute BoundaryCond "Boundary condition ID" : String [isNotNull isUnique]
  attribute altName "Altimetry name" : String [isNotNull isUnique]
  attribute resName "Resistance name" : String [isNotNull isUnique]
  primaryKey (Name)
  foreignKey e_InitialCond (altName, resName)
  foreignKey e_BoundaryCond (BoundaryCond)]

DataEntity e_Spatial "Spatial reference" : Reference[
  attribute CRS "Coordinate reference system" : Text [isNotNull]
  attribute Name "Mesh name" : String [isNotNull isUnique]
  primaryKey (CRS, Name)
  foreignKey e_Mesh (Name)]
  
```

Specification 42: RSL specification of HiStav pre-processor DataEntities: Outputs.

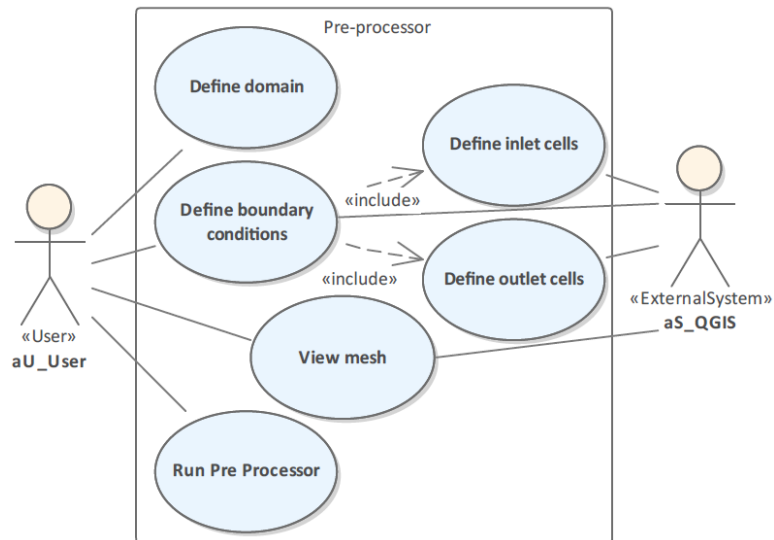


Figure 70: UML diagram of the pre-processor UseCases.

```

UseCase uc_1_MakeSimulation "Make simulation" : EntityCreate [
    actorInitiates aU_User
    dataEntity ec_Results
    actions aCreate
    extensionPoints EPDefineDistCond
    includes uc_2_DefinePhysical, uc_3_DefineNumeric, uc_4_DefineTime
    description "Run the HiStav to obtain the simulation results."]

UseCase uc_2_DefinePhysical "Define physical constants" : EntityCreate []

UseCase uc_3_DefineNumeric "Define numeric information" : EntityCreate []

UseCase uc_4_DefineTime "Define time information" : EntityCreate []

UseCase uc_1_1_DefineDistCond "Define distributed conditions" : EntityCreate []

UseCase uc_5_RunParaView "Run ParaView" : EntityRead [
    actorInitiates aU_User
    actorParticipates aS_ParaView
    dataEntity ec_Results
    actions aRead
    extensionPoints EPVisualizeResults
    description "Run the ParaView to visualize the results."]

UseCase uc_5_1_VisualizeResults "Visualize results" : EntitiesMapInteract [
    actorInitiates aU_User
    actorParticipates aS_ParaView
    dataEntity ec_Results
    actions aRead
    includes uc_6_ReadMatrixResults
    extends uc_5_RunParaView onExtensionPoint EPVisualizeResults]

UseCase uc_6_ReadMatrixResults "Read results matrix" : EntityRead [
    actorInitiates aS_ParaView
    dataEntity ec_Results
    precondition "Results matrix must be created previously."
    actions aRead]
    
```

Specification 43: RSL specification of HiStav UseCases.

RiverCure

Users Permissions

Table 4: Permissions of RCP user roles.

Permissions		User roles							
		SuperAdmin	Manager	Technician					Citizen
Main roles				Spatial technician	Numeric technician	Udometric technician	Hydrometric technician	Radar technician	
Secondary roles									
Manage users		X							
Manage updates		X							
Manage information		X	X						
Manage alarms	View	X	X			X	X		
	Edit and delete	X	X			X*	X*		
Monitoring points	View real-time data	X	X			X	X		
	View historical data	X	X	X	X	X	X	X	X
Manage data		X	X			X	X		
Radar observations	View data	X	X	X	X	X	X	X	X
	Manage	X	X					X	
Events	View open events	X	X			X	X		
	View historical events	X	X	X	X	X	X	X	X
Manage events		X	X						
Simulations	View simulations	X	X	X	X	X	X	X	X
	Create scenarios	X	X	X	X	X	X	X	X
	Create simulations	X	X	X	X				
	Upload files	X	X	X	X				
	Manage files owned	X	X	X	X	X	X	X	X
Manage files (not owned)		X	X						
Interactive map	View data	X	X	X	X	X	X	X	X
	Edit basemap	X	X	X	X	X	X	X	X
	Edit definitions	X	X						
Photos	Upload	X	X	X	X	X	X	X	X
	View photo and data	X	X	X	X	X	X	X	X
	View information history	X	X	X					X
	Manage photos owned	X	X	X	X	X	X	X	X
Manage photos (not owned)		X	X	X					X
About	View	X	X	X	X	X	X	X	X
	Edit	X							

X Has permission
X* Has permission with restrictions

Glossary

Term trm_boundary_cond "Boundary condition" : Noun [applicableTo DataEntity description
 "Condition to be satisfied by a dependent variable of a differential equation along the boundary of a model domain."

Term trm_initial_cond "Initial condition" : Noun [applicableTo DataEntity description
 "Description of the discharge, depth of flow or other dynamic condition at the beginning of a simulation period for unsteady flow models. Over the domain/mesh (all or just a region)."

Term trm_user "User" : Noun [applicableTo Stakeholder, Actor description
 "A user of the system is someone that has a user account;
 Each user profile has different permissions in terms of editing, viewing, accessing tools and setting alarms.
 There are the following types of user profiles:
 - Citizen, responsible for uploading photos, managing photos owned and creating new simulation scenarios;
 - Technician, responsible for setting alarms, managing data and simulations.
 - Manager, responsible for configuring technical features;
 - SuperAdmin, responsible for the portal, namely for managing users accounts."

Term trm_citizen "Citizen" : Noun [applicableTo Stakeholder, Actor description
 "A citizen is a user that is able to define its own settings, view data from different sources, make simulations scenarios, upload photos, manage photos owned, access the interactive map, view events records."

Term trm_technician "Technician" : Noun [applicableTo Stakeholder, Actor description
 "A user that is able to view data from different sources in real time, run new simulations, upload photos, view open events, access the interactive map and define alarms.
 It is responsible for setting alarms, configuring its own settings, managing data (stations, photos, radar, events) and simulations. Certain functionalities depend on the secondary role, which can be:
 radar, numeric, udometric, spatial, crowdsourcing or hydrometric technician."

Term trm_manager "Manager" : Noun [applicableTo Stakeholder, Actor description
 "User that is responsible for configuring the system (e.g. export configuration parameters), setting alarms, configuring account settings, managing data (stations, photos, radar) and simulations."

Term trm_SuperAdmin "SuperAdmin" : Noun [applicableTo Stakeholder, Actor description
 "Is a user that has total access to the portal, this means that has access to all the information and tools.
 It is responsible for managing the users accounts."

Specification 44: Partial RSL specification of RCP glossary.

Data Entities

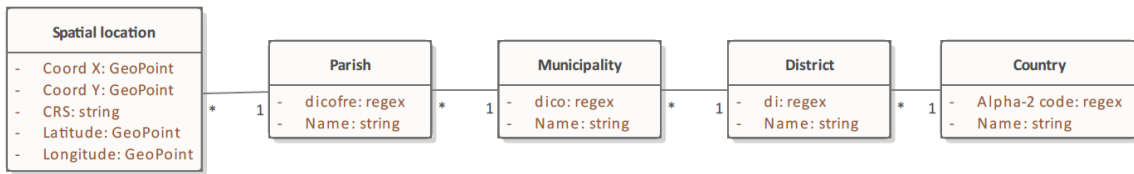


Figure 71: UML diagram of the RCP DataEntities: Spatial package.

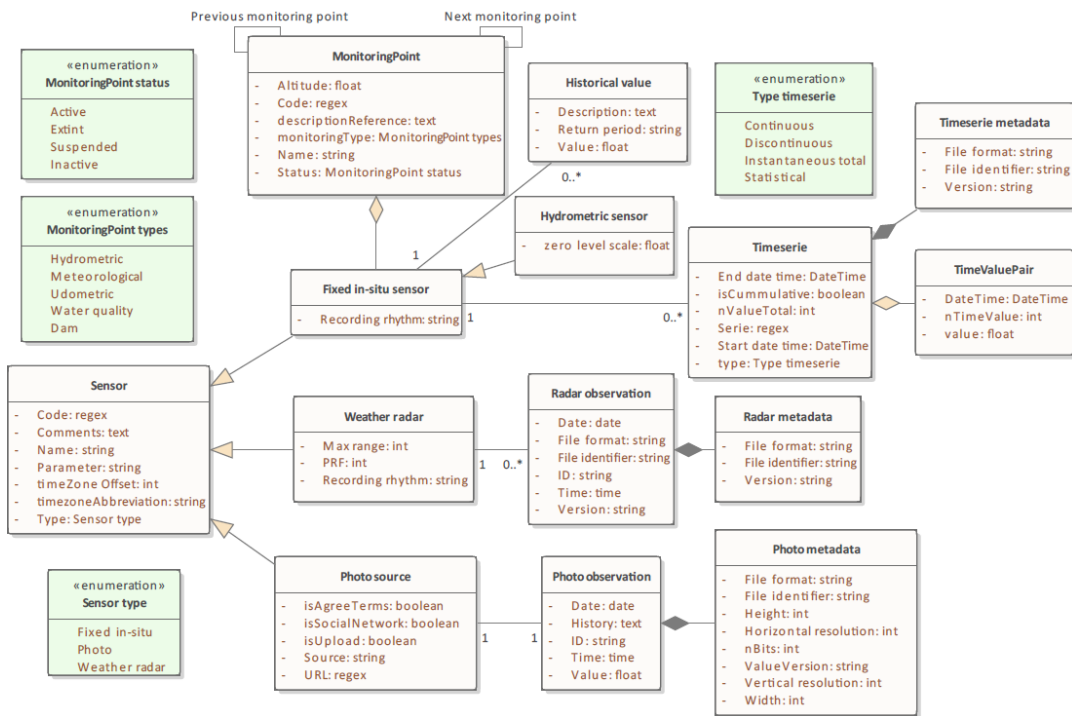


Figure 72: UML diagram of the RCP DataEntities: Sensor package.

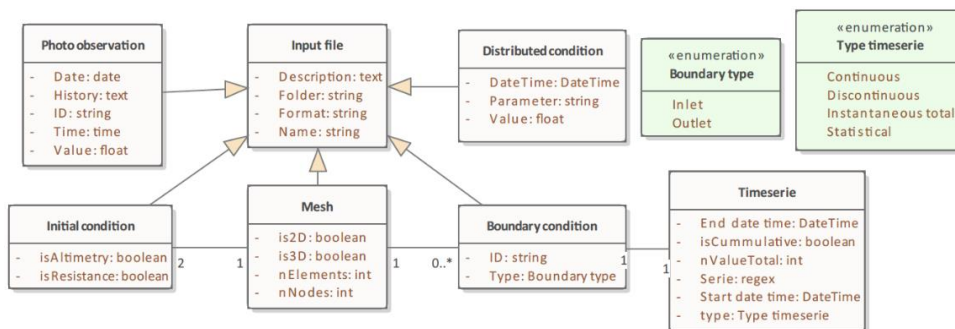


Figure 73: UML diagram of the RCP DataEntities: Simulation input package.

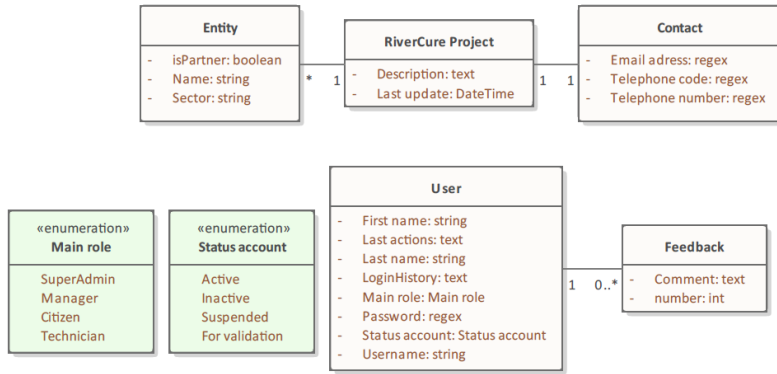


Figure 74: UML diagram of the RCP DataEntities: About package.

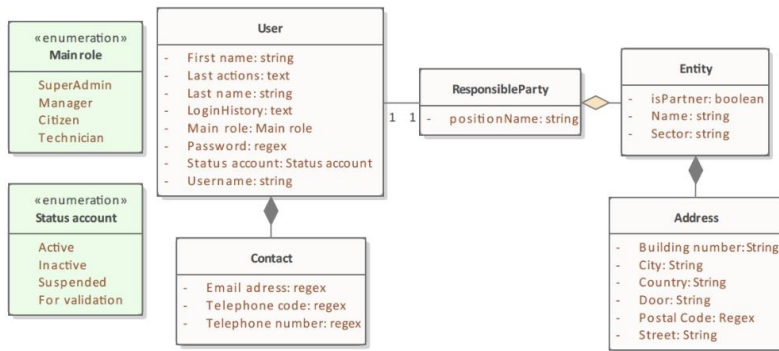


Figure 75: UML diagram of the RCP DataEntities: User main package.

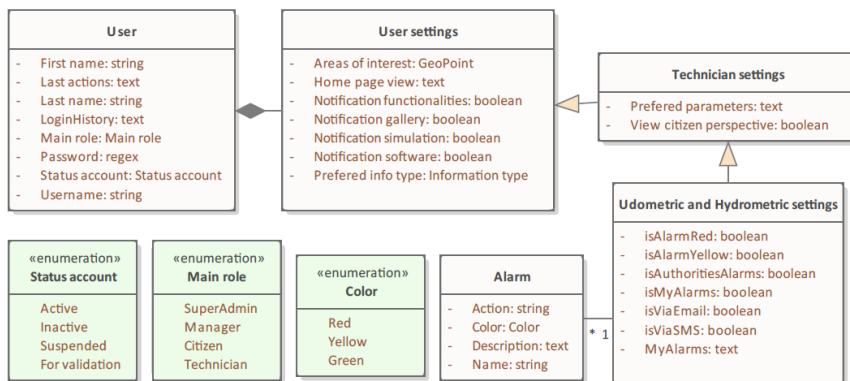


Figure 76: UML diagram of the RCP DataEntities: User settings package.


```

DataEntity e_HydroFeature "Hydrometric feature" : Master [
  attribute code "Code" : Regex [isNotNull isUnique]
  attribute Name "Name" : String [isNotNull isUnique]
  attribute type "Type" : DataEnumeration de_HydroType [isNotNull]
  attribute shape "Shape" : DataEnumeration de_Geometry [isNotNull]
  attribute dico "Municipality code" : Regex [isNotNull isUnique]
  primaryKey (code)
  foreignKey e_Municipality (dico)]

DataEnumeration de_HydroType "Hydrometric type" : Simple [values
"Dam", "River basin", "Drainage basin", "River", "Estuary", "Lake"]

DataEnumeration de_Geometry "Geometry" : Simple [values "Point", "Line", "Polygon"]

DataEntity e_RiverBasin "River basin" : Reference [
  isA e_HydroFeature
  attribute area "Area" : Decimal
  attribute AlarmName "Alarm name" : String [isNotNull isUnique]
  attribute basinCode "Basin code" : Regex [isNotNull isUnique]
  foreignKey e_Alarm (AlarmName)]

DataEntity e_River "River" : Reference [
  isA e_HydroFeature
  attribute lenght "Lenght" : Decimal [isNotNull]
  attribute hierarchy "Hierarchy" : String [isNotNull]
  attribute basinCode "Basin code" : Regex [isNotNull isUnique]
  foreignKey e_HydroFeature (basinCode)]

```

Specification 45: RSL specification of RCP DataEntities: Hydro feature.

```

DataEntity e_Sensor "Sensor" : Master [
  attribute code "code" : Regex [isNotNull isUnique]
  attribute Name "Name" : String [isNotNull isUnique]
  attribute comment "Comment" : Text
  attribute parameter "Parameter" : String [isNotNull]
  attribute type "Type" : DataEnumeration de_SensorType [isNotNull]
  attribute TimeZoneAbbreviation "TimeZone abbreviation" : String
  attribute TimeZoneOffset "TimeZone offset" : Integer
  attribute username "Username responsible" : String [isNotNull isUnique]
  primaryKey (code, parameter)
  foreignKey e_User (username)]

DataEnumeration de_SensorType "Sensor type" : Simple [values "Fixed in-situ", "Photo", "Weather"]

```

Specification 46: RSL specification of RCP DataEntity: Sensor.

```

DataEntity e_FixedInSitu "Fixed in-situ sensor" : Master [
  isA e_Sensor
  attribute recrhythm "Recording rhythm" : String [isNotNull]
  attribute SensorCode "Sensor code" : Regex [isNotNull isUnique]
  attribute latitude "Latitude" : GeoPoint [isNotNull]
  attribute longitude "Longitude" : GeoPoint [isNotNull]
  foreignKey e_Spatial_Location (latitude, longitude)]

DataEntity e_HydrometricSensor "Hydrometric sensor" : Parameter []

DataEntity e_Timeserie "Timeserie" : Document [
  attribute endDateTime "End date time" : Datetime [isNotNull]
  attribute isCumulative "is Cumulative" : Boolean [defaultValue "False"]
  attribute nValueTotal "Total number of values" : Integer [isNotNull]
  attribute serie "Serie number" : Regex [isNotNull isUnique]
  attribute startDateTime "Start date time" : Datetime [isNotNull]
  attribute type "Type" : DataEnumeration de_TimeserieType
  attribute SensorCode "Sensor code" : Regex [isNotNull isUnique]
  primaryKey (serie, SensorCode)
  foreignKey e_FixedInSitu (SensorCode)]

DataEnumeration de_TimeserieType "Timeserie type" : Simple [values
"Continuous", "Discontinuous", "Instantaneous total", "Statistical"]

DataEntity e_TimeserieMetadata "Timeserie metadata" : Document : Detail []

DataEntity e_TimeValuePair "Time value pair" : Document : Detail []

DataEntity e_HistoricalValue "Historical value" : Parameter []

DataEntity e_MonitoringPoint "Monitoring point" : Master [
  attribute code "Code" : Regex [isNotNull isUnique]
  attribute descriptionReference "Description reference" : Text
  attribute monitoringType "Monitoring type" : DataEnumeration de_MonitoringPoint_Types [isNotNull]
  attribute Name "Name" : String [isNotNull isUnique]
  attribute NextPoint "Next monitoring point" : String [isNotNull isUnique]
  attribute PreviousPoint "Previous monitoring point" : String [isNotNull isUnique]
  attribute Status "Status" : DataEnumeration de_MonitoringPoint_Status [isNotNull]
  attribute altitude "Altitude" : Decimal [isNotNull]
  attribute SensorCode "Sensor code" : Regex [isNotNull isUnique]
  primaryKey (code)
  foreignKey e_Sensor (SensorCode)]

DataEnumeration de_MonitoringPoint_Types "Monitoring point types" : Simple []

DataEnumeration de_MonitoringPoint_Status "Monitoring point status" : Simple [values []]

```

Specification 47: RSL specification of RCP DataEntities: Fixed in-situ.

```

DataEntity e_PhotoSource "Photo source" : Master [
  isA e_Sensor
  attribute isSocialNetwork "is from a social network" : Boolean [defaultValue "False" isNotNull]
  attribute isUpload "is from an upload" : Boolean [isNotNull]
  attribute Source "Source" : String [isNotNull]
  attribute url "URL" : Regex
  attribute isAgreeTerms "is agree with terms and conditions" : Boolean [defaultValue "False"]
  attribute SensorCode "Sensor code" : Regex [isNotNull isUnique]]

DataEntity e_PhotoObservation "Photo observation" : Document [
  isA e_InputFile
  attribute date "Date" : Date [isNotNull]
  attribute time "Time" : Time
  attribute Value "Value" : Decimal
  attribute ID "Photo ID" : String [isNotNull isUnique]
  attribute history "History" : Text
  attribute SensorCode "Sensor code" : Regex [isNotNull isUnique]
  attribute latitude "Latitude" : GeoPoint [isNotNull]
  attribute longitude "Longitude" : GeoPoint [isNotNull]
  attribute HydroFeatureCode "Hydrometric feature code" : Regex [isUnique]
  primaryKey (ID)
  foreignKey e_PhotoSource (SensorCode)
  foreignKey e_Spatial_Location (latitude, longitude)
  foreignKey e_HydroFeature (HydroFeatureCode)]

DataEntity e_PhotoMetadata "Photo metadata" : Document : Detail [ ]

```

Specification 48: RSL specification of RCP DataEntities: Photo.

```

DataEntity e_InteractiveMap "Interactive map" : Master [
  attribute basemap "Basemap" : String [isNotNull]
  attribute dateInterval "Date interval" : Date [isUnique]
  attribute isDateInterval "is date interval" : Boolean [defaultValue "True"]
  attribute nLastDays "Number of last day(s)" : Integer [defaultValue "7"]
  attribute isLastDays "is last days" : Boolean [defaultValue "False"]
  primaryKey (basemap)]

DataEntity e_Layer "Layer" : Master [
  attribute symbol "Symbol" : Image [isNotNull]
  attribute symbolColor "Symbol color" : String [isNotNull isUnique]
  attribute type "Information type" : DataEnumeration de_InformationType [isNotNull]
  attribute dateInterval "Date interval" : Date [isUnique]
  primaryKey (type)
  foreignKey e_InteractiveMap (dateInterval)]

DataEntity e_EventLayer "Event layer" : Master [
  isA e_Layer
  attribute isOpen "is open" : Boolean
  attribute is48h "is last 48h" : Boolean
  attribute is72h "is last 72h" : Boolean
  attribute EventCode "Event code" : Regex [isNotNull isUnique]
  foreignKey e_Event (EventCode)]

DataEntity e_AlarmLayer "Alarm layer" : Master [ ]

DataEntity e_SensorLayer "Sensor layer" : Master [ ]

DataEntity e_SimulationLayer "Simulation layer" : Master [ ]

```

Specification 49: RSL specification of RCP DataEntities: Map.

```

DataEntity e_Alarm "Alarm" : Master []

DataEnumeration de_Color "Color" : Simple [values "Red", "Yellow", "Green"]

DataEntity e_AlarmSetting "Alarm setting" : Master : Detail [
  attribute lowthreshold "Low threshold" : Decimal [isNotNull]
  attribute upthreshold "Up threshold" : Decimal [isNotNull]
  attribute AlarmName "Alarm name" : String [isNotNull isUnique]
  attribute SensorCode "Sensor code" : Regex [isNotNull isUnique]
  primaryKey (AlarmName)
  foreignKey e_FixedInSitu (SensorCode)
  foreignKey e_Alarm (AlarmName)]

DataEntity e_Event "Event" : Master [
  attribute code "Code" : Regex [isNotNull isUnique]
  attribute Description "Description" : Text [isNotNull]
  attribute EndDateTime "End date and time" : Datetime
  attribute StartDateTime "Start date and time" : Datetime [isNotNull]
  attribute isBelowGreenLowThreshold "is below green low threshold" : Boolean [defaultValue "False"]
  attribute lastUpdate "Last update" : Datetime [isReadOnly]
  attribute State "State" : DataEnumeration de_EventState [isNotNull]
  attribute Type "Type" : DataEnumeration de_EventType [isNotNull]
  attribute AlarmName "Alarm name" : String [isNotNull isUnique]
  attribute SensorFixedCode "Fixed in-situ sensor code" : Regex [isNotNull isUnique]
  attribute PhotoID "Photo observation ID" : String [isNotNull isUnique]
  attribute username "Username responsible" : String [isNotNull isUnique]
  primaryKey (code)
  foreignKey e_User (username)
  foreignKey e_Alarm (AlarmName)
  foreignKey e_FixedInSitu (SensorFixedCode)
  foreignKey e_PhotoObservation (PhotoID)]

DataEnumeration de_EventType "Event type" : Simple [values []]

DataEnumeration de_EventState "Event state" : Simple [values "Occuring", "Concluded"]

```

Specification 50: RSL specification of RCP DataEntities: Event and Alarm.

```

DataEntity e_Simulation "Simulation" : Master [
  attribute Description "Description" : Text
  attribute folder "Folder" : String [isNotNull]
  attribute isShared "is shared" : Boolean [defaultValue "False"]
  attribute lastUpdate "Last update" : Datetime [isNotNull isReadOnly]
  attribute Name "Name" : String [isNotNull isUnique]
  attribute Version "Version" : String [isNotNull isUnique]
  attribute type "Type" : DataEnumeration de_SimulationType [isNotNull]
  attribute NameInput "Input file name" : String [isNotNull isUnique]
  attribute PhysicalConstant "Physical constant name" : String [isNotNull isUnique]
  attribute startDateTime "Start datetime" : Datetime [isNotNull]
  attribute writeInterval "Write interval" : Integer [isNotNull]
  attribute username "Username" : String [isNotNull isUnique]
  attribute CodeDistrict "District code" : Regex [isNotNull isUnique]
  primaryKey (Name, Version)
  foreignKey e_InputFile (NameInput)
  foreignKey e_PhysicalConstant (PhysicalConstant)
  foreignKey e_TimeInfo (startDateTime)
  foreignKey e_NumericInfo (writeInterval)
  foreignKey e_User (username)
  foreignKey e_District (CodeDistrict)]

DataEnumeration de_SimulationType "Simulation type" : Simple [values "Simulation", "Scenario"]

DataEntity e_SimulationScenario "Simulation scenario" : Parameter [
  isA e_Simulation
  attribute Version "Version" : String [isNotNull isUnique]
  attribute startDateTime "Start datetime" : Datetime [isNotNull]
  attribute endDateTime "End datetime" : Datetime
  foreignKey e_TimeInfo (startDateTime, endDateTime)]

DataEntity e_TimeInfo "Time information" : Parameter []

DataEntity e_NumericInfo "Numeric information" : Parameter []

DataEntity e_PhysicalConstant "Physical constant" : Reference [
  attribute Name "Name" : String [isNotNull isUnique]
  attribute Unit "Unit" : String [isNotNull]
  attribute Value "Value" : Decimal [isNotNull]
  primaryKey (Name)]

DataEntity e_OutputFile "Output file" : Master [
  attribute folder "Folder" : String [isNotNull]
  attribute format "Format" : String [isNotNull]
  attribute Name "Name" : String [isNotNull isUnique]
  attribute nPoints "Number of points" : Integer [isNotNull]
  attribute SimulationName "Simulation name" : String [isNotNull isUnique]
  attribute Version "Simulation version" : String [isNotNull isUnique]
  primaryKey (Name)
  foreignKey e_Simulation (SimulationName, Version)]

DataEntity e_InputFile "Input file": Master []

```

Specification 51: RSL specification of RCP DataEntities: Simulation main.

```

DataEntity e_UserRole "User role" : Master : Detail [
  attribute Name "Name" : String [isNotNull isUnique]
  attribute type "Type" : DataEnumeration de_UserRoleType [isNotNull]
  primaryKey (Name)]

DataEnumeration de_UserRoleType "User role type" : Simple [values "Main", "Secondary"]

DataEntity e_Technician "Technician" : Parameter [
  isA e_UserRole
  attribute secondaryRole "Secondary role" : DataEnumeration de_SecondaryRole [isNotNull]]

DataEnumeration de_SecondaryRole "Secondary role" : Simple [values
"Spacial", "Numeric", "Hydrometric", "Udometric", "Radar", "Crowdsourcing"]

DataEntity e_User_permission "User permission" : Parameter [
  attribute alarms "Alarm permission" : DataEnumeration de_AlarmPermission [isNotNull]
  attribute about "About permissions" : Text [isNotNull]
  attribute Events "Events permissions" : Text [isNotNull]
  attribute InteractiveMap "Interactive map permissions" : Text [isNotNull]
  attribute isManageInformation "is manage information" : Boolean
  attribute ManageInformation "Manage information" : Text [isNotNull]
  attribute isManageUpdates "is manage updates" : Boolean
  attribute ManageUpdates "Manage updates" : Text [isNotNull]
  attribute isManageUsers "is manage users" : Boolean
  attribute ManageUsers "Manage users" : Text [isNotNull]
  attribute MonitoringPoints "Monitoring points permissions" : Text [isNotNull]
  attribute Photos "Photos permissions" : Text [isNotNull]
  attribute RadarObs "Radar observations permissions" : Text [isNotNull]
  attribute Simulations "Simulations permissions" : Text [isNotNull]
  attribute role "Role name" : String [isNotNull isUnique]
  attribute basinCode "Basin code" : Regex [isNotNull isUnique]
  primaryKey (role)
  foreignKey e_UserRole (role)
  foreignKey e_RiverBasin (basinCode)]

DataEnumeration de_AlarmPermission "Alarm permission" : Simple [values
"Yes", "Yes (only authorities alarms)", "No", "Depend on secondary role"]

```

Specification 52: RSL specification of RCP DataEntities: User roles.

Requirements

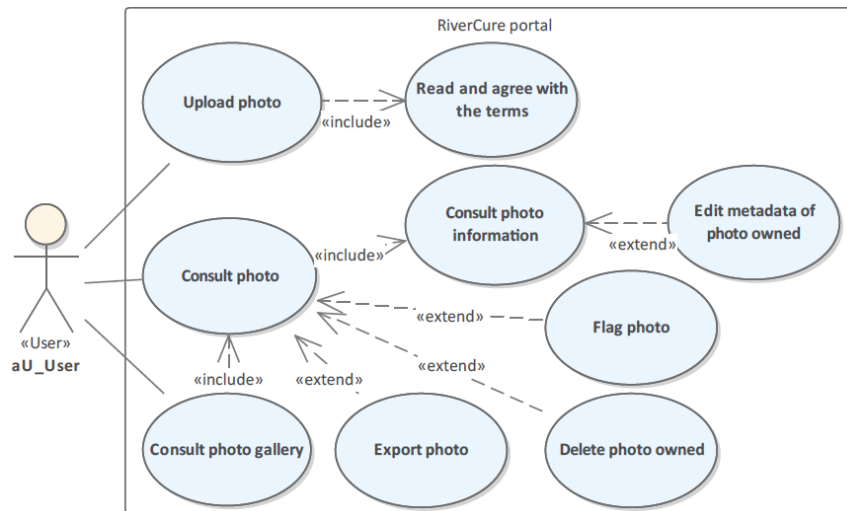


Figure 77: UML diagram of the RCP UseCases: User photos package.

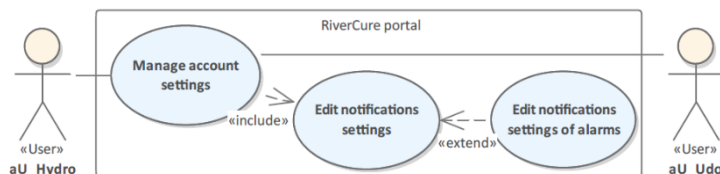


Figure 78: UML diagram of the RCP UseCases: Hydrometric and Udometric settings package.

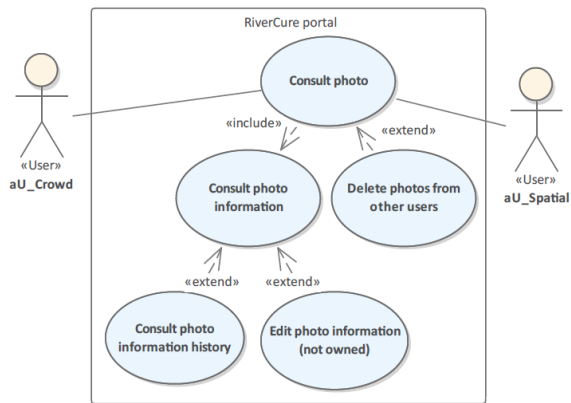


Figure 79: UML diagram of the RCP UseCases: Photos package.

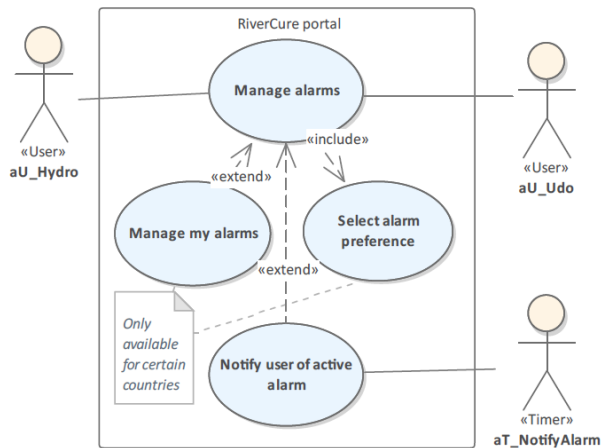


Figure 80: UML diagram of the RCP UseCases: Alarm package.

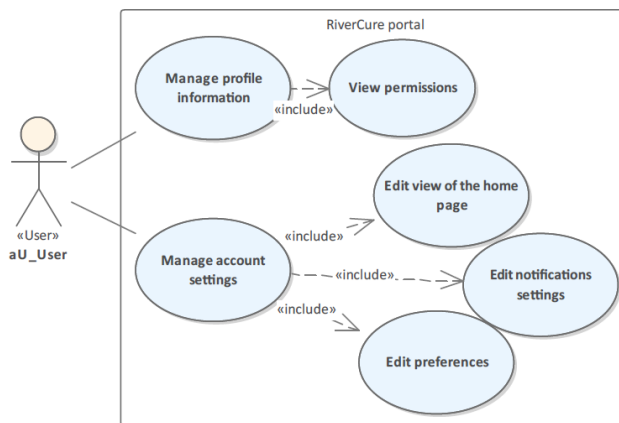


Figure 81: UML diagram of the RCP UseCases: Settings package.

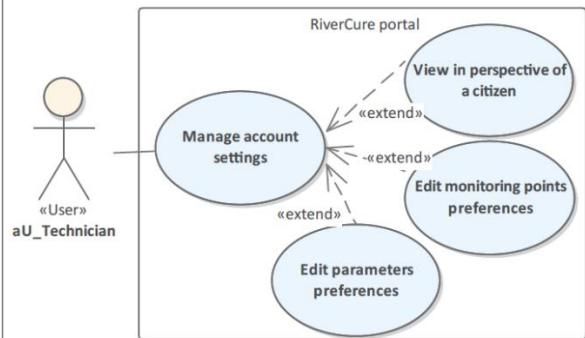


Figure 82: UML diagram of the RCP UseCases: Technician settings package.

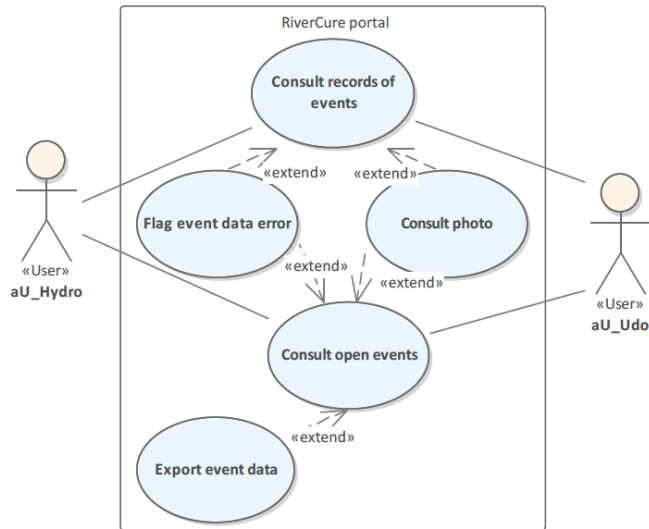


Figure 83: UML diagram of the RCP UseCases: Events package.



Figure 84: UML diagram of the RCP UseCases: SuperAdmin package.

```

UseCase uc_1_ConsumeMonitoringPointsData "Consult monitoring points data" : EntitiesBrowse []
UseCase uc_1_1_ViewDataBasin "View data by basin" : EntitiesBrowse []
UseCase uc_1_2_ViewDataDistrict "View data by district" : EntitiesBrowse []
UseCase uc_1_3_ConsumeMonitoringPointSchema "Consult monitoring point schema" : EntitiesBrowse []
UseCase uc_1_4_ViewDataMonitoringPoints "View data by monitoring points" : EntitiesBrowse []
UseCase uc_1_4_1_ReadDataRecords "Read data records" : EntitiesBrowse [
  actorInitiates aU_User
  dataEntity ec_Fixed_Insitu
  actions aRead, aFilter]
UseCase uc_1_4_2_ReadRealTimeData "Read real-time data" : EntitiesBrowse [
  actorInitiates aU_Hydro
  actorParticipates aU_Udo
  dataEntity ec_Fixed_Insitu
  actions aRead, aFilter]
UseCase uc_1_5_CreateGraphic "Create graphic" : EntityCreate [
  actorInitiates aU_User
  dataEntity ec_Fixed_Insitu
  actions aRead, aSelect, aExport, aChangeColor
  extends uc_1_ConsumeMonitoringPointsData onExtensionPoint EPCreateGraphic]
UseCase uc_2_FlagMonitoringPoint "Flag monitoring point or its data" : EntityOther [
  actorInitiates aU_Hydro
  actorParticipates aU_Udo
  dataEntity ec_Fixed_Insitu
  actions aFlag]

```

Specification 53: RSL specification of RCP UseCases: Monitoring points.

```

UseCase uc_23_ManageInformation "Manage information" : EntitiesManage [
  actorInitiates aU_Manager
  dataEntity ec_UserInformation
  actions aCreate, aDelete, aRead, aEdit, aSelect, aView
  extensionPoints EPManageMonitoringPoints, EPManageRadarObservations, EPManageEvents, EPManageSimulations,
  EPViewPhotosFlagged, EPEditMapDefinitions, EPViewPortalAnalytics, EPAnalyseUserAccountsLogs]
UseCase uc_23_1_ManageMonitoringPoints "Manage monitoring points" : EntitiesManage []
UseCase uc_23_1_1_AddMonitoringPoint "Add monitoring point" : EntityCreate []
UseCase uc_23_1_2_DeleteMonitoringPoint "Delete monitoring point or its data" : EntityDelete []
UseCase uc_23_1_3_EditMonitoringPoint "Edit monitoring point or its data" : EntityUpdate []
UseCase uc_23_2_ManageRadarObservations "Manage radar observations" : EntitiesManage []
UseCase uc_16_2_ViewRecordsFlagged "View events records flagged" : EntitiesBrowse []
UseCase uc_23_3_ManageSimulations "Manage simulations and input files (not owned)" : EntitiesManage []
UseCase uc_23_4_ViewPhotosFlagged "View list of photos flagged" : EntitiesBrowse []
UseCase uc_23_5_EditMapDefinitions "Edit interactive map definitions" : EntityUpdate []
UseCase uc_23_6_ViewPortalAnalytics "View portal analytics" : EntitiesDashboard []
UseCase uc_23_7_AnalyseUserAccountsLogs "Analyse user accounts logs" : EntitiesDashboard []

```

Specification 54: RSL specification of RCP UseCases: Manage information.

Prototype

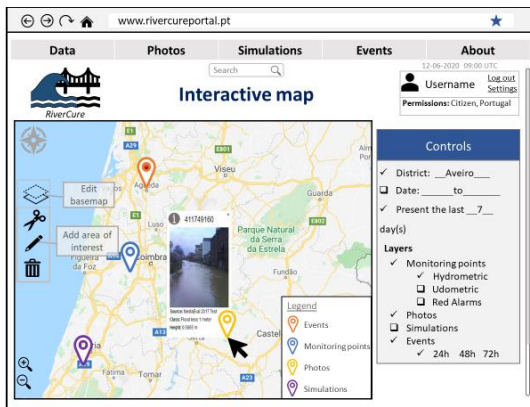


Figure 85: Interactive map (photos) mock-up of the RCP.

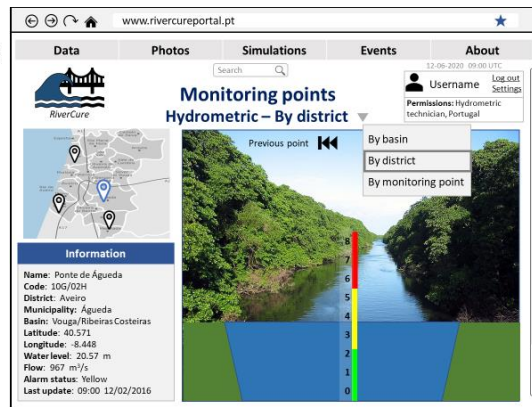


Figure 86: Monitoring points (By district) mock-up of the RCP.

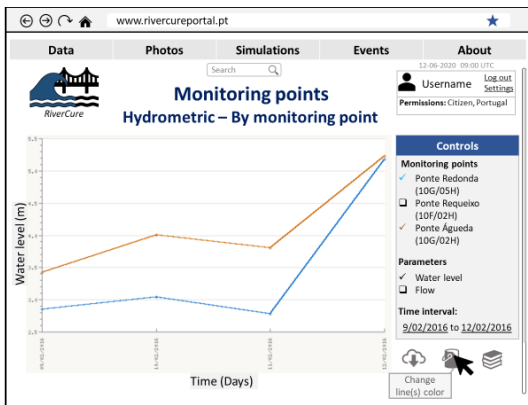


Figure 87: Monitoring points (Graphic) mock-up of the RCP.

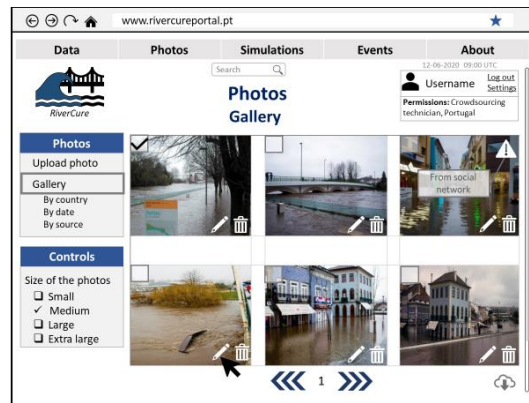


Figure 88: Photos gallery (Crowdsourcing technician) mock-up of the RCP.

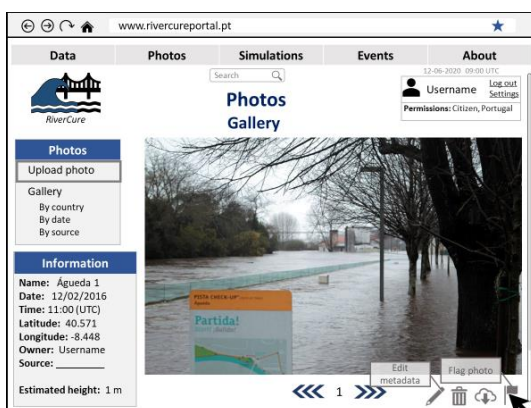


Figure 89: Photos gallery mock-up of the RCP.

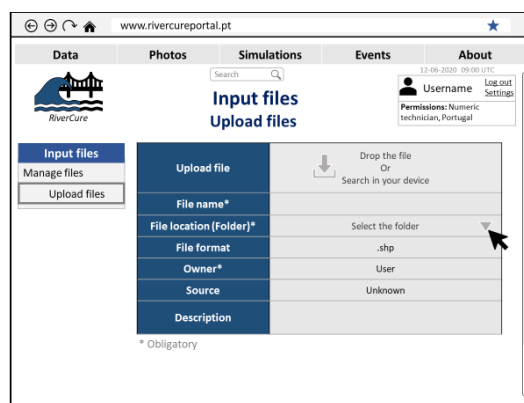


Figure 90: Simulations input files (Upload files) mock-up of the RCP.

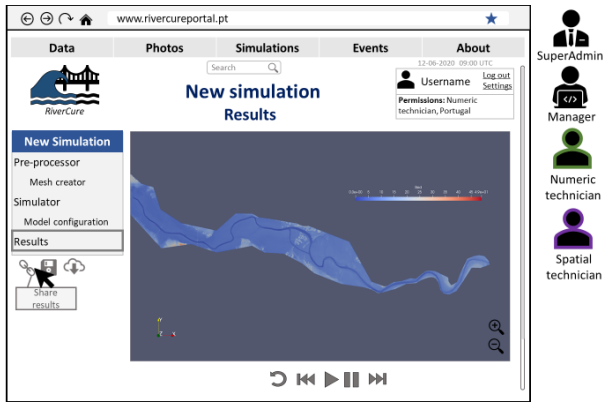


Figure 91: New simulation (Results) mock-up of the RCP.

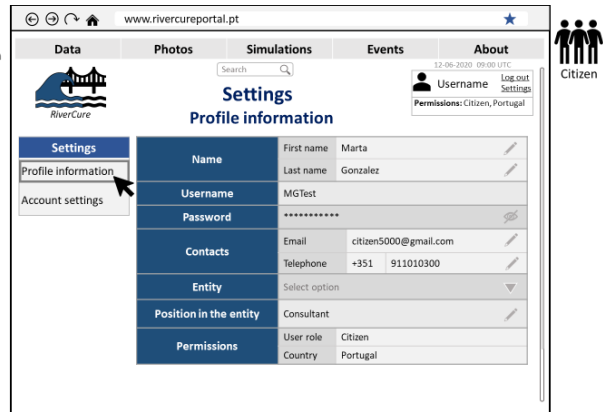


Figure 92: Settings (Profile information) mock-up of the RCP.

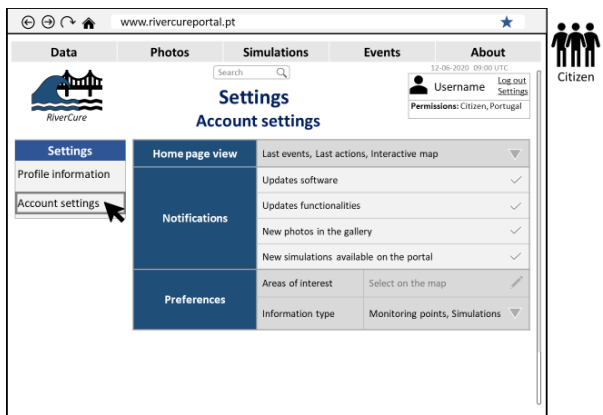


Figure 93: Settings (Account) mock-up of RCP.

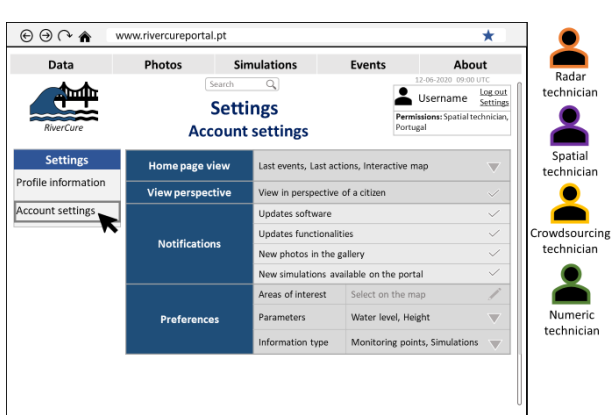


Figure 94: Settings (Account) mock-up of the RCP.

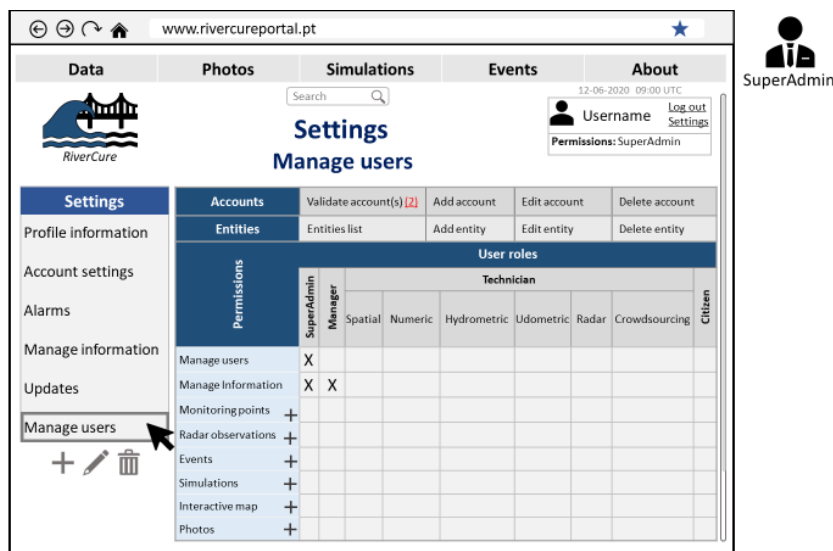


Figure 95: Settings (Manage users) mock-up of the RCP.

Meetings

APA (with Dra. Manuela Saramago)

- 8/01/2019 – Introduction of the RiverCure project;
- 12/04/2019 – Introduction to the SNIRH and SVARH systems (how data is collected, who are the stakeholders involved);
- 21/06/2019 – Meeting with the RiverCure team about the use of the SVARH's Silverlight version, the modelling tool used by APA (MIKE) and certain details regarding the systems;
- 4/07/2019 – Demonstration of the SVARH system (This meeting had the purpose to introduce the topics to the new members of the team);
- 25/10/2019 – Validation of the models (specifically use cases and data entities) regarding SNIRH and SVARH;
- 22/11/2019 – Visualization of the SVARH's Gestor and Editor.

RiverCure

- 18/06/2018 – Project introduction by Professor Alberto;
- 14/09/2018 - Situation point with the RiverCure team;
- 4/10/2018 – Team meeting for planning the correlation of the students' master thesis with the RiverCure project;
- 25/10/2018 - Situation point with team and presentation regarding WaterML;
- 18/09/2019 – Meeting with Professor Rui Ferreira regarding the RCP requirements;
- 22/10/2019 – Meeting with Jorge Pereira regarding the social flood app, developed by him;
- 6/11/2019 – Meeting with Professor Alberto and Professor Rui Ferreira, where the following topics were approached: Situation point regarding SVARH, SNIRH and HiSTAV; Decisions regarding the portal were made; Materials needed;
- 20/11/2019 - Situation point with Professor Alberto;
- 20/12/2019 – Meeting with Professor Rui and other members of the team;
- 8/1/2020 – Meeting with the RiverCure team regarding the situation point and future tasks;
- 21/1/2020 – Meeting with Jorge Marques, Ivo, Professor Rui, Professor Alberto and Professor Bruno to align some requirements and present the first draft of the RCP;
- 1/04/2020 – Meeting regarding the RCP and its users with Jorge Marques, Ivo, Rui Granja, Professor Jacinto, Professor Alberto.

HiSTAV

- 17/10/2019 – Online meeting with Daniel Conde, Marcos Dionisi and Professor Rui Ferreira regarding specific doubts related to the treatment of the data from the case study;
- 18/10/2019 – Meeting with Marco Dionisi regarding the system functioning;
- 18/11/2019 – Meeting with Professor Rui regarding the information's flux of the system;
- 20/12/2019 – Meeting with Marco Dionisi regarding some doubts related to the tool.