TÉCNICO LISBOA

Citizen Science with Immediate Feedback

Guilherme André das Neves Eugénio

Thesis to obtain the Master of Science Degree in

Masters in Electrical and Computer Engineering

Supervisor(s): Prof. João Nuno De Oliveira e Silva
Dr. Pedro Pinho

Examination Committee

Chairperson: Prof. Teresa Maria Sá Ferreira Vazão Vasques
Supervisor: Prof. João Nuno De Oliveira e Silva
Member of the Committee: Prof. Pedro Abílio Duarte de Medeiros

January, 2021

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.

Acknowledgments

Queria demonstrar o meu agradecimento ao Prof. João Nuno Silva por me ter apresentado um tema tão interessante e por me ter orientado a criar não só uma ferramenta que contribui para a solução ao problema específico que foi proposto, mas também que poderá ser adaptada para um leque infinito de contextos científicos.

Obrigado ao Dr. Pedro Pinho por todo o acompanhamento desta dissertação, fazendo uma contextualização e esclarecimento de dúvidas sobre Biologia, Ecologia, Evolução e Alterações Ambientais alinhando-os com os atuais problemas no contexto científico.

Um profundo agradecimento aos meus pais por toda a preocupação, acompanhamento nos bons e maus momentos, permitindo-me alcançar sempre os meus sonhos.

Um especial agradecimento ao Miguel Leitão por ter sido não só o melhor colega de curso que pude pedir como também um grande amigo que teve inúmeras vezes presente, nas mais diversas situações. Um obrigado também ao Joaquim Silva por todo o trabalho e amizade, ajudando-me ao longo de todo o meu percurso universitário.

Resumo

Ciência do Cidadão é uma poderosa ferramenta de extração de informação, podendo ser aplicada num amplo leque de contextos científicos. Através de surveys, dados podem ser colectados de forma a contribuir para a informação já existente disponível para a comunidade científica. Mudanças Globais - incluindo alterações climáticas, poluição do ar e mudanças no uso dos solos, estão a colocar em risco a saúde da humanidade bem como o equilíbrio existente nos ecossistemas. Existe uma necessidade de mapear estes efeitos espacialmente mas a quantidade de informação não é suficiente. Esta falta de informação advém da falta de estações de monitorização disponíveis. Para além disso, os dados aí extraídos não são suficientemente esclarecedores para obter informação sobre os efeitos dos impulsionadores das mudanças globais. Líquenes, sendo sensíveis à poluição do ar e às alterações climáticas são frequentemente utilizados como indicadores ambientais. A extração de dados baseados nos líquenes pode ser feita de duas formas: analisando o seu conteúdo em poluentes para perceber a quantidade relativa de componentes poluentes presentes no ar e analisando a sua biodiversidade. O último método, relevante para este trabalho, permite extrair alguns índices baseados na diversidade funcional dos líquenes podendo ser usados para perceber o impacto relativo das diferentes alterações climáticas. Aliado com os princípios da Ciência do Cidadão, este método pode ser usado para fazer face à falta de informação existente fazendo uma extensão de uma ferramenta genérica de *Crowdsourcing*, com a possibilidade para ser modificada e melhorada, providenciando uma experiência de feedback imediato ao utilizador.

Palavras-chave: Ciência do Cidadão, Alterações Climáticas, Poluição do Ar, Sistema de Crowdsourcing Genérico, Indicadores Ambientais, Líquenes

Abstract

Citizen Science it's a powerful data sourcing tool that can be a driving force for a wide range of scientific scopes. Through surveys, information can be collected from the users to support the existing data available for the researchers community. Global change – including climate change, air pollution and land-use change, is affecting human health and ecosystem functioning. There is a need to map its effects over space, but the amount of information is low. This scarceness is derived from the insufficiency of monitoring stations available. In addiction, its data is not enlightening enough to obtain information about the effects of the global change drivers. Lichens, being sensitive to air pollution and climate change are frequently used as ecological indicators. Data can be extracted from lichens by two ways: analyzing their content in pollutants to understand the relative quantity of air pollutants and analyzing its biodiversity. The last method, relevant for this work, allows to extract some indexes based on the lichen functional diversity, understanding the relative impact of different environmental effects. Aligned with the principles of Citizen Science, this method can be used to tackle the existing scarceness of information by extending a Generic Crowdsourcing System with the ability to be modified and improved, providing immediate feedback to the user.

Keywords: Crowdsourcing, Air Pollution, Climate Change, Land-use Change, Lichens, Climate Proxies.

Contents

List of Figures

Nomenclature

API Application Programming Interface

APK Android Package

AVD Android Virtual Device

BSON Binary JSON

cE3c Centre for Ecology, Evolution and Environmental Changes

CSS Cascading Style Sheets

DB Database

DOM Document Object Model

FCUL Faculdade de Ciências da Universidade de Lisboa

GCS Geo Crowd Surveys

GIS Geographic Information System

GPS Global Positioning System

GUI Graphical User Interface

HTML HyperText Markup Language

HTTP HiperText Transfer Protocol

IFM Immediate Feedback Map

iLTER Networks of scientists engaged in long-term, site-based ecological and socio-ecological research

IMAP Internet Message Access Protocol

IPA iOS App Store Package

IPCC Intergovernmental Panel on Climate Change

IST Instituto Superior Técnico

JS JavaScript

JS JavaScript

JSON JavaScript Object Notation

MERN MongoDB, Express JS, React JS and Node JS

MIT Massachusetts Institute of Technology

NPM Node Package Manager

OS Operating System

PDF Portable Document Form

PoC Proof of Concept

REST Representational State Transfer

SPECO Sociedade Portuguesa de Ecologia

UGS Urban Green Spaces

UI User Interface

WHO World Health Organisation

WSGI Web Server Gateway Interface

Chapter 1

Introduction

Citizen science, a narrower subset of crowdsourcing, is emerging as a new form of interaction between scientists and citizens, allowing for social participation and involvement in scientific activities. Mapping and spatial data collection are two activities that shifted from a situation of researchers exclusiveness to public involvement due to the significant technological advances during the last decade [1]. Thus, crowdsourcing campaigns can be used in the framework of air pollution and climate change, collecting geolocated data that can be used to support the lichens mapping process.

Global Change, referring to the planetary-scale changes in the Earth system, include climate change, air pollutants and land-use change. Air pollution and climate change are closely related. A major source of air pollutants is the extraction and burning of fossil fuels, making it the main cause of CO2 emissions and driving climate change. At the atmosphere level, Ozone warms the climate, while different components of particulate matter can have either warming or cooling effects [2]. Air pollutants are increasing day by day, contributing to human health issues [3].Urban heat island is an event that occurs when a city experiences warmer temperatures than nearby rural areas. This difference has to do with how well the surfaces in each environment absorb and hold heat. In recent years, urban heat island effect has become a international issue for being the responsible for exacerbating the intense heat of the city. This climate issue seriously affects the urban ecological environment and the health of urban citizens [4]. Land use and changes in land cover can significantly contribute to overall impacts on local to global-scale weather and climate. Its impacts should not be underestimated, altering the flow of energy, water, and greenhouse gases between the land and the atmosphere. The impacts of climate change are ”global in scope and unprecedented in scale. Without drastic action today, adapting to these impacts in the future will be more difficult and costly” [5].

Lichens, considered to be the result of a symbiotic association of a fungus and/or an alga, are considerably sensitive to air pollution and climate change thus being frequently used as ecological indicators. Being a efficient ecological indicator for air pollution and climate change especially sulfur dioxide pollution, lichens derive their water and essential nutrients mainly from the atmosphere rather than from the soil [6]. It is proven that lichen functional groups have a strong association with the surrounding environmental variables [7]. Analyzing lichen diversity: species richness, lichen cover, and lichen com-

munity composition provides information about global change metrics. Thus, lichens can be used by researchers to complement the existing air pollution and climate change database and so, field campaigns are a possible but expensive and time consuming solution to the lack of mapping data.

The impacts of Global Change are unprecedented and a concerning topic for researchers, highlight the urge of information about the climate change drivers and their respective effects. This data reliability could allow a better analysis and possibly the formulation of conclusions and/or predictions. Existing monitoring stations provide accurate geolocated data about the global change drivers but they are not sufficiently numerous. Besides, monitoring stations don't provide information about the effects of the global change drivers. This scarceness of data sources are a concerning issue for researchers, highlighting the need to find accurate alternatives to tackle this problem.

Aligned with the principles of Citizen Science, a Generic Crowdsourcing System, can be developed, deployed and used. This surveying tool will provide a simple UI to extract user data based on the advanced computing capabilities of smartphones. This generic tool consists on features that are considered common on surveying based applications powered by a scientific issue. This features can be programmed in order to fulfill narrowed scientific requirements. This adaptability will allow to extend the Generic Crowdsourcing System into a powerful surveying tool for the scope of this dissertation.

An extended tool of the Generic Crowdsourcing System will provide a surveying mechanism that will supply data that can be used to tackle the existing scarceness of information in the context of air pollution and climate change. Lichen biodiversity will be used as a driving force for this narrowed system, contributing positively to obtain scientific data about global change drivers and their effects. Aligned with a simple UI, this application will provide Immediate Feedback to the user, inducing a responsive and user friendly surveying system.

1.1 Motivation

The impacts of global change are worldwide in scope and unprecedented in scale. Global climate change is expected to intensify the occurrence of urban heat island events, where air temperatures in cities rise disproportionately to surrounding areas affecting citizen's health, having economic and environmental impacts. Implications for local air quality, heat stress, morbidity, mortality [8] are some of the urban heat island effects, highlighting the importance of this matter.

Air pollution and climate change are among the more dangerous threats to Human-health. Asthma, rhinosinusitis, chronic obstructive pulmonary disease (COPD) and respiratory tract infections are examples of respiratory diseases that may be caused or aggravated with climate change [9]. By increasing temperature and the time of exposure, specially to vulnerable groups, cardiovascular disease mortality will increase and these increases will be intensified in the future decades. There's a strong positive association between maximum temperatures and mortality and the same, but weaker, for minimum temperatures [10]. Heatwaves, droughts, hurricanes intensification, sea level rise are some of the many climate change effects.

The pollution and climate change effects maps are an excellent way to understand the effects of

global change factors over space. On the other hand, the available maps are scarce in information, providing inaccurate and unreliable data for researchers. Global change effects could not be accurately represented in map due to their variations at high scale. For example, atmospheric pollutants, such as particulate matter and urban heat island effects can vary in different distance including short distances, such as $<500m$ [11]. This low map resolution represents a problem for local analysis that demands a spatially explicit sampling with high resolution.

One way to overcome this problem is to use ecological indicators, which reflect the long-term effect of air pollution and climate change and can be sampled in many locations [7]. Lichen growing on tree trunks are frequently used as ecological indicators of the effect of air pollution and climate.

Being a efficient proxy for air pollution and climate change, lichens field campaigns are a possible but expensive and time consuming solution to the lack of mapping data.

Citizen Science it's a powerful database contributor tool among many other used, being uses in many applications, for example environmental monitoring. By gathering data through surveying, it is a practice of public participation and collaboration in scientific research to increase scientific knowledge. Being a open tool, relying on a wide range of different people, it must consist on a simple tasks, without requiring a specific scientific knowledge.

1.2 Objectives

The Centre for Ecology, Evolution and Environmental Changes (cE3c) - FCUL, supported by iLTER, proposed to sample lichen diversity in a citizen science project, to map the effect of air pollution and climate, using an app-based interface. By returning immediate feedback to the user, processed surveying data is immediately provided. This will enhance user experience, allowing a better user engagement and involvement in scientific projects.

The existing lack of a generic surveying tool that could be adapted to any scientific scope represents a huge downside on the researchers community. Thus, a Generic Crowdsourcing System, with surveying features capable of being extended, represents the main objective of this work. This GCS will be structured by the principles of Citizen Science. A society collaborative work tool that could help solving this problem by offering simple and specific tasks to collaborators without the need of any scientific knowledge. This generic surveying tool would aggregate data, providing data fetching, data validation, storage, processing and feedback modules that could be programmed and activated.

There are several contributions that this system pretends to add in several Citizen Science based solutions and so, for example, in *Science*, occurs an improvement in the air quality and climate change mapping: migrate from a scarce to a high density map in space. Also, another defined objective is to contribute to the *Citizenship* awareness about air quality and climate change, being used the Citizen Science as a tool to make this approach. To *Society* in general, with a medium to long term concern to future generations, contributing to the SDG (Sustainable Development Goals) , namely the ones related to the atmospheric pollution and climate change (SDG11, SDG13 and SDG15)[12] that due to the increasingly humanity impact on Earth has been an extremely important topic of discussion in the

last decades.

As a proof of concept, this Generic Crowdsourcing System will be programmed and adapted fulfilling the narrowed requirements mentioned by cE3c - FCUL, supported by iLTER. To achieve this narrowed proposed objective, firstly is important to understand what are the current Global Change drivers that are relevant for this work, their respective impacts on society and their effects on ecosystems. Among many components of global change, air pollution, land-use change, climate change and their effects are the driving force for this work. These drivers will be represented through metrics: aridity, eutrophication and poleotolerance.

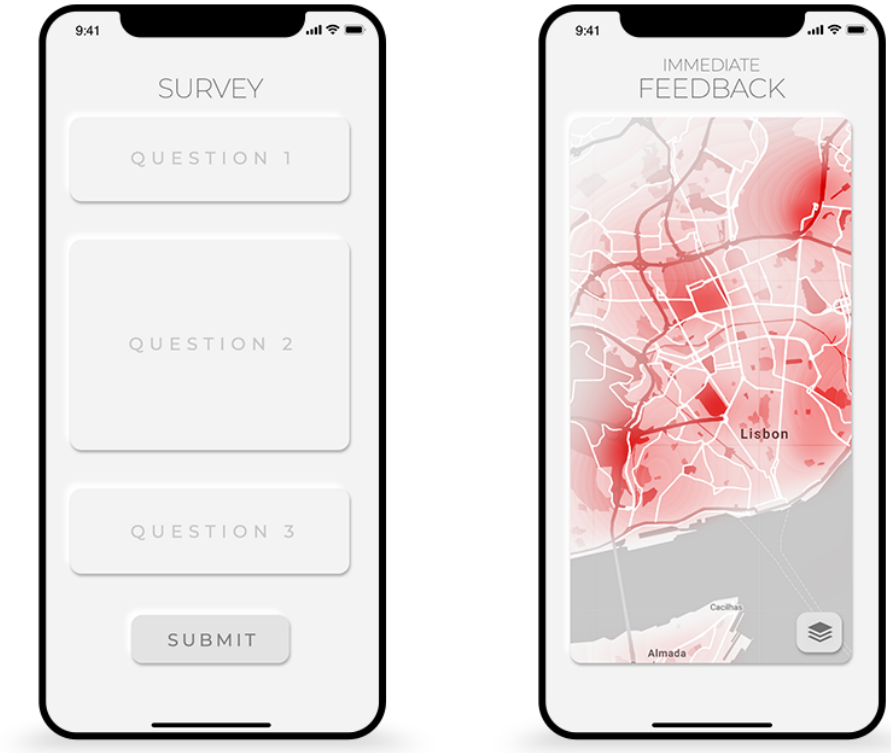


Figure 1.1: Draft mock-up of the mobile interface.

This particular proof of concept of the Generic Crowdsourcing System would be an mobile interface pretended to help to solve the problem presented by the Centre for Ecology, Evolution and Environmental Changes. It relies on the contribution of the regular user, by filling up several form fields. Following the principles of citizen science, the contributing citizen will submit survey data, contributing to the shared database, providing immediate feedback to the user. Thus, the app should be able to accept the inputs from its user, along with a timestamp and geo location shared by the user's device in order to provide a complete and structured user data. That data is then forwarded to a server and stored in a database. The processing component handles the input data, aggregating all the available system data and returning immediate feedback to the user. On this scientific scope, the metrics of the global change drivers are calculated based on indexes. These value indexes are relative, not absolute, so the usage of a heatmap represents the most efficient method of immediate feedback. This heatmap would provide a simple and

interactive interface, allowing to analyze the data gradients overlaid to the displayed map.

Aligned with the mobile interface, a complementary web application is another objective of this work. Data validation would be available on this complementary tool, allowing researchers to have control over the stored data. Data Download is another option that is useful in the scientific community. By having predefined geolocated formats, data could be used in external Geographic Information System applications.

The final long term objective is that, by extending the Generic Crowdsourcing System to several scientific scopes, Citizen Science would be used more often, providing simple UI to the users, involving citizens in scientific projects. Consequentially, it would bring awareness and knowledge to overall society while contributing positively to a less scarce data repository. This long term objective applies not only to the problem proposed by cE3c - FCUL, supported by iLTER, having the success of this system application as a proof of concept of the GCS tool extensibility.

1.3 Thesis Outline

Not only used to define the scope and focus of a dissertation, Thesis Outline also describes what to expect from the thesis structure. The software system that is here described provides features shared by some components. Thus, this thesis pretends to expose the growth and development of each feature, justifying structural and functional decisions made. This progress will be described in each of the following Chapters, developing a complete and efficient tool from scratch.

Background

Also called State of Art, it represents the highlights and conclusions based on the research made. It describes global change and their drivers relevant for this work, highlighting the chosen metrics. This section describes lichens, explaining their composition and their relation to the environment, justifying their usage as ecological indicators in a citizen science project. Explaining the crowdsourcing scope, this section will provide existing examples and their downsides.

Requirements

Having as a dissertation objective the development of a Generic Crowdsourcing System, the requirements of this tool will be detailed on this section. Afterwards, to fulfill the objectives of the narrowed scope of this thesis, the GCS will be adapted. Thus, the extension requirements are also described, detailing only the needed modifications to the generic tool.

Architecture

Using C4 Model, this section will describe the different architectural levels of the Generic Crowdsourcing System. This model describes the system with different abstraction levels. On a low level urged the need of dividing into the 3 system components: server, mobile and web application.

Implementation

This section describe the technologies, environments and languages used as well as libraries and modules. Functions, states, endpoints and system API calls are detailed to describe the Generic Crowdsourcing Implementation. It describes struggles and solutions that were taken throughout the development stage. It is organized by features and not through the system main components. This decision was made because there are features that are equal in different components or a single feature may use all the system components.

Demonstration

After implementation, the application is stored and deployed. Demonstration details the usage of the Generic Crowdsourcing System, detailing each system component, making this generic tool available to the programmed and adapted to a specific scientific scope. As a proof of concept, the GCS is extended to fulfil the specific requirements of the proposed issue of this dissertation. The implementation of eFlechten, the narrowed specific tool to tackle the information scarceness about global change drivers using lichens sampling, is also described on this Section. Another proof of concept, developed by Miguel Leitão for its MSc Thesis for Electrical and Computer Engineering about Urban Green Spaces is also detailed on the Demonstration.

Results

Applying a Google Forms to a sampling group of users, eFlechten, the narrowed crowdsourincg tool developed for this thesis, is evaluated with the System Usability Scale (SUS). Another relevant information about the feedback of this tool is also described and analyzed on this Section.

Conclusions

This section highlights the research objectives achieved, describing the overall conclusions that were made with this dissertation. It also shows the limitations of the developed work and possible upgrades that can be done in the future.

Chapter 2

Background

2.1 Global Change

Global change is composed of a series of separate problems – climate change, biodiversity loss, dwindling water resources, air pollution, land-use change and many more. Despite different, biological and physical processes interact to determine prevalent global environmental conditions. The ecosystems on Earth are not isolated, presenting dynamic inter exchanges. For example, Greenhouse gases and inferred temperature show regular synchronised variations over hundreds of thousands of years. In the framework of the global change is important to highlight to this work the climate change, the air pollution and the land-use changes and their environmental impacts.

Climate Change is the defining issue of our time and we are at a defining moment." [5] The impacts of climate change are "global in scope and unprecedented in scale. Without drastic action today, adapting to these impacts in the future will be more difficult and costly" [5]. Observations of the climate system are crucial to understand current climatic trends [13], whereas climate models are used to make environmental projections. Quantities like global mean air temperature and sea level are expected to respond to anthropogenic perturbations of the Earth's radiation budget.

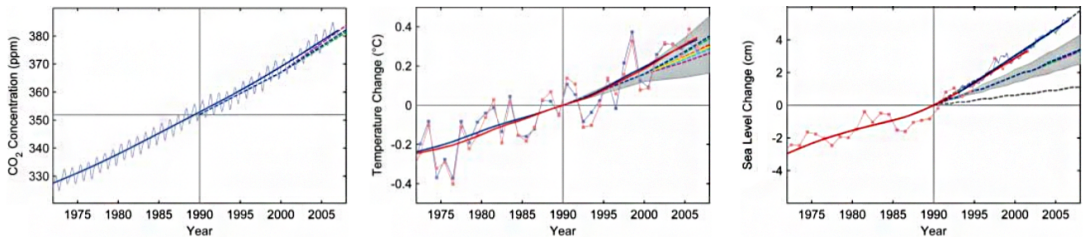


Figure 2.1: Changes in key global climate parameters since 1973, compared with the scenarios of the predictions from the IPCC (shown as dashed lines and gray ranges).

The IPCC, Intergovernmental Panel on Climate Change, started scenarios and projections in the year 1990. On the **left** image of Figure 2.1, is represented the monthly carbon dioxide concentration and its trend line at Mauna Loa, Hawaii, 2007, from Scripps in collaboration with NOAA. In the **middle**,

annual global mean land and ocean combined surface temperature from GISS (red) and the Hadley Centre/ Climatic Research Unit (blue) up to 2006, with their trends. On **right**, sea-level data based primarily on tide gauges (annual, red) and from satellite altimeter (blue).

To summarize, the data now available on Figure 2.1 shows that carbon dioxide concentration follows the projections almost exactly but it raise concerns, particularly regarding sea level, since it may be responding more quickly than climate models predicts [13]. The global increases in carbon dioxide concentration are due primarily to fossil fuel use and land-use change. Methane and nitrous oxide, also greenhouse gases, are primarily due to agriculture [14]. Cardiovascular mortality and respiratory illnesses due to heatwaves, altered transmission of infectious diseases and malnutrition from crop failures are some of the nefarious effects of climate change on human health [15]. "Changes in temperature and rainfall conditions also may influence transmission patterns for many diseases, including water-related diseases, such as diarrhoea, and vector-borne infections, including malaria " according to the World Health Organisation. WHO also claims that patterns of food production can be affected by climate change, having health impacts in terms of rates of malnutrition.

The source of the major public health concerns come from anthropogenic sources related to of air pollution. Motor vehicles, industrial facilities or to natural sources such as forest fires are some of the responsible for the emissions of air pollutants that contribute to human health issues such as particulate matter, benzo(a)pyrene, carbon monoxide, ozone, nitrogen dioxide, and sulfurdioxide [3]. Respiratory diseases and lung cancer are one of the many consequences of the exposure to poor air quality at different life stages, ranging from prenatal period to adult life making this a public health issue that must be taken seriously [16].

Human activities influence the surrounding environment by altering the distribution of ecosystems and their associated flows of energy. Water vapor, trace gases and particulates are emitted with the human presence activity on field. This management regime that humans impose on a site is called land-use. The transformation of the natural landscape, emphasizing the functional role of land for economic activities is called land-use change. Its impacts on climate change are notable. For example, when large areas of forests are cleared, transpiration is reduced. This changes in land-cover patterns can directly impact energy and mass fluxes, resulting on less cloud formation, less rainfall, and increased drying [17].

Urbanization and migration of people from rural areas to cities started in the 1950s and still continues today [18]. The rural exodus, the migratory pattern of people from rural areas into urban areas, changed the assessment of air pollution exposure in environmental epidemiology studies at the urban scale to be prioritized since the majority of the global population is expected to live in urban areas in the coming decades [19].

Among many consequences, a high number of premature deaths are one of the environmental problems of urbanization that will affect citizen's health, making a strong impact socially and economically [20]. By 2050, premature mortality is estimated to double because of the contribution of outdoor air pollution, reaching approximately 6.6 million premature deaths per year around the world [21] and, if no actions are taken, is is expected 250 000 annual deaths due to the urban heat island effect [22]. Thus,

it is crucial for the sake of environment and humanity future to take some action towards this matter.

Some nature-based solutions can be taken like, for example, the presence of urban green spaces [23][24], that despite being an overarching solution, the relative importance of specific vegetation structure, composition and management are somehow forgotten key factors that are proven to influence the ecosystem services of air purification and climate regulation [11]. The spatial placement of this urban green spaces must be done strategically, urging to enhance the environment quality of the most climate affected areas. For that, a good quality mapping analysis is required, deploying high resolution and diverse data that can be used to achieve accurate conclusions.

There are efforts from the scientific institutions and governmental agencies to regularly collect data on the economic, social or environmental status in all kind of forms, including maps. Nevertheless, scarce human and technical resources compromise gathering, managing , processing and delivering of scientific high quality data. Furthermore, sharing of raw data produced by scientists during field work is practically non-existent. According to the international journal Rural 21 [25], most of the national research institutions in West Africa have neither policies nor facilities to manage and share research data. Data sharing culture is almost non existent, being not limited to this region. On one hand, a sharing platform with high quality data would be an efficient way of tackle this existing scarceness. On the other hand, there's a lack of willingness to share time and efforts with the researchers' community.

Climate dimensions data can be directly measured from the monitoring stations, getting accurately retained locally. On the other hand the lack of existing monitoring stations are far from being enough to obtain high resolution data, for example, on climate change and air pollution maps. Therefore, biological indicators are considered to be a good reason to measure responses to climate change [26]. This happens because such responses may reflect the ability of organisms to cope with the effects of the global change drivers.

2.2 Lichens as Ecological Indicators

Lichens are considered the result of a symbiotic association of a fungus and/or an alga, a Cyanobacteriae or a Chlorophyceae. The fungus is usually an Ascomycetes, although on rare occasions it may be either a Basidiomycetes or a Phycomycetes. [27]. Lichens have no stomata or cuticle and so their thallus absorb gases and aerosols and quickly spread them over the photobiont. The photobiont is the first to be affected, with abnormalities in the thallus, such as chlorophyll bleaching and yellowish brown areas in the chloroplasts [28]. Unlike vascular plants lichens have free exchange of both gases and solutions, occurring across the cell surfaces. Almost all their nutrients from the atmosphere through uptake over their entire surface, rather than from the soil [6]. With this nutrient atmospheric absorbency, pollutants are also consumed. As a result, some species are more sensible to that pollutants absorbency, and others more tolerant. The same happens regarding the lichen surrounding climate, resulting in heterogeneous levels of tolerance towards aridity and humidity. This divergent species tolerance allows to aggregate lichen species in functional groups. Lichen sensitivity to various environmental factors made lichens to be widely used as indicators of climate changes [29] and as good surrogates of air pollution and climate

change during the last 30 years. Lichen's role in environmental and human health studies has gained importance in the last decade, when it was found a negative relationship between lichen biodiversity and human lung cancer mortality[30].

In order to get high spatial resolution maps of pollution and climate change, the existing monitoring stations don't provide sufficient data for the researchers community and so, ecological indicators are a powerful tool to overcome this problem. In general, bioindicators are organisms that can be used for the identification and qualitative determination of environmental factors generated by humans [31]. Besides the lack of monitoring data stations another problem is the collective effect of the pollutants in urban environments, making it difficult to unveil their effects [32] and so, since bioindicators integrate the results of air pollution over time, they are also a good option to analyze this collective pollutants effect [27].

Modern lichen-based environmental analyses provide low cost, high-resolution spatial tools for modelling and mapping the effects of climate change, used to complement the traditional networks of pollution or climate monitoring stations, allowing its detection, assessment and monitoring at the ecosystem level [33][34].

Metrics of lichen diversity such as species richness, lichen cover, and lichen community composition are some examples of metrics used to quantify the effects of atmospheric pollutants on ecosystems [35]. Lichen diversity is frequently used to monitor the effects of atmospheric pollution in urban areas. For example, Munzi et al. [36] carried out, in Rome, a study about lichen diversity between 1982 and 2003. Besides air pollution, the most important variable affecting the epiphytic lichen flora of Rome, is the influence of the Tyrrhenian Sea. Significant changes in the lichen flora have been noted over the past 20 years, with the lowest lichen diversity now being found in the urban centre and in the eastern and southern sectors, while the "lichen desert" area has decreased in parallel with decreasing concentrations of CO, NO_x and SO_2. The results of this paper show that the epiphytic lichen flora of Rome is composed mostly of species adapted to an anthropogenic environment, characterized by the complex interactions of contrasting biological and ecological variables. It is proven that lichen functional groups have a strong association with the surrouding environmental variables, for example microlichens to forest stand structure, nitrophytes to atmospheric NH_3 concentrations, photobiont type to potential solar radiation are some of the functional groups that stand out [7].

Another advantage, not to be used in this project context, is the lichen capacity to accumulate pollutants being used as a cost-effective method to assess pollutants deposition for decades [37][34]. As a result of this property of lichens, several papers have been published on heavy metal monitoring of lichens in different geographic areas [37]. Due to the reduction of SO_2 emission, NO_x and NH_3 are currently the main pollutants affecting lichens in Europe and North America [38]. This pollutants accumulation may, in turn, result in physiological damage either in the algae or in the fungal component, generating a loss of the lichen vitality [39], being therefore used as surrogates of the early impact signs of the air quality [37].

Lichens, as good surrogates of air pollution can be used as a high-resolution spatial tools for modelling and mapping the effects of climate change, when compared to the traditional networks of pollution

or climate monitoring stations, allowing its detection, assessment and monitoring at the ecosystem level [33][34].

Lichens in particular have been widely used as trace element atmospheric ecological indicators as they are widespread and capable of absorbing elements directly from the atmosphere and accumulating them in their tissues. On the other hand, using lichens as climate surrogates, like traditional networks of pollution or climate monitoring stations, are time consuming and so an efficient tool needs to be discussed in order to ease this problem, the Citizen Science.

2.3 Citizen Science

Citizen Science [1], an emerging new form of interaction between scientists and citizens, allows social participation and involvement in scientific activities. It can lead the participants to have a great social engagement with science through the participation in real science projects. Interacting directly with scientists, the participants are able to acquire a perception of the scientific activity by the educational and engaging value that these projects can have in particular circumstances.

There is a range of terms to describe the general subject area of citizen-derived geographic information and it is used variably over time. Similarly, there are a wide range of Internet sites that, in one way or another, use citizen-derived geographic information. Some terms like Geocollaboration, GeoWeb, Contributed Geographic Information or Participatory [1] sensing are, among many, examples of terms that, despite presenting small differences between them, are all used in discussion of what it has been collectively referred to as crowdsourced geographic information.

Citizen science can assume many forms, provided citizens are voluntarily involved in it, constituting a major change in the way research is conducted in several areas, while opening new possibilities. Many projects deal with great societal challenges and threats, such as climate change, biodiversity loss, pollution, habitat destruction, or health quality.

Mapping and spatial data collection are two activities that have dramatically changed in the last decades, shifting from a situation of researchers exclusiveness to public involvement due to the significant technological advances during the last decade [1]. The ease of creating content online more easily through Web 2.0 , the proliferation of advanced mobile devices equipped with a great amount of diverse and precise technologies that can provide different types of media and user inputs, open access to satellite imagery and online maps are some of the technological advances that made possible the public involvement.

Urban areas offer enormous potential for citizen science projects because there are already a enormous amount of possible volunteers. The ecology of urban areas is a growing field that still has a lot to be developed and investigated since the type and scope of information citizen scientists can provide is invaluable.

Data collection quantity is not the only benefit that can be taken from incorporating citizen science into ecological research. Volunteers gain experience in making observations, registering them, having an active participation in the scientific study, but most importantly, developing a great sense of maintenance

and protection over the populations or sites they are responsible for surveying or monitoring [40].

After data collection on citizen science projects, the results can be displayed and made accessible to the public. This immediate response is one of the main challenges in the crowdsourcing scope, allowing to return feedback to the user, right after submission. It can increase residents' knowledge and awareness about their community issues, ranging from a local to a global scale, normally inserted in a scientific project. This involvement can encourage citizens to participate more actively in this kind of scientific initiatives, supporting a more informed researchers team.

2.3.1 Mobile Crowdsourcing

Smartphones have now become the a new tool for data fetching, storage and communication. The development of the existing smartphone mechanisms and the appearance of new ones turned this kind of device an efficient and accurate tool to collect user data.

With this growing popularity of smartphones, there are emerging more and more mobile crowdsourcing platforms like gMission [41], Waze [42], ChinaCrowd [43]. The mobile crowdsourcing platforms are based on a wide range of volunteers, with all kinds of scientific knowledge and so it is more challenging to control quality, latency and cost for mobile platforms [44]. The interests and commitment of each individual varies substantially and so, the flexibility that smartphones give to Crowdsourcing allows to enhance task design and incentives to keep the user interested and motivated in the developed tool [45].

2.3.2 Incentives

In order to retain user's attention incentives could play a key role in the successful use of crowdsourcing. This incentives are important to ensure data reliability, reducing the error margin from the GCS. Monetary interests can be the key driver to user motivation but other incentives address social aspects, entertainment and altruism [46]. Another incentive that is proven to be a powerful way to retain user's attention and commitment is gamification, being more and more used in nowadays mobile apps. "Gamification is the concept to develop incentives aiming at entertainment and fun for the subjects" [47] and so, systems with user rankings, points attribution based on actions, points exchange or even user levels are some of the examples that could be used in this interactive incentive. This level of engagement could be an option for keeping the users flow in a Crowdsourcing platform, maintaining a certain level of reliability on input data. "For example, Eickhoff C [48] shows that gamification reduces fake ratings significantly by a factor of five and that innovative, creative tasks are less likely to invite cheating, increase data quality and efficiency"[47].

2.3.3 Survey Requirements

To ensure system reliability, the accuracy of the data produced is a key factor on making a successful Crowdsourcing platform. This success on crowdsourcing campaigns will typically be both attractive to potential participants but also fulfil sufficient data quality standards [49]. This results in a trade-off on the

crowdsourcing platforms, between maintaining a high reliability, data quality standards and keeping the platform's design simple, engaging and enjoyable at a user perspective [50]. For example, Tomnod [51] conclude that the simplicity and social currency of tasks makes them both appealing and straightforward for people who may not typically engage with online content. As the use of crowdsourcing spreads, the need to ensure the quality of data in crowdsourced platforms is magnified.

Borromeo, in a study made concerning the "Influence of Crowd Type and Task Complexity on Crowdsourced Work Quality" [52], analyzed some works done concerning task complexity. Accuracy on user outputs has been flagged as worse in tasks with complex UI's. For example, Finnerty et al. [53] after presenting a domain categorization task with either one or two questions in either simple or complex graphical user interfaces, reported that the number of questions in a task and a more complex crowdsourcing survey affected the users' performance. This concludes that, in general, results show that simple tasks are more likely to yield high-quality results than more complex ones.

2.3.4 Existing Software for Citizen Science Survey

Relating to the Citizen Science topic discussed in the Subsection 2.3, the participants surveying process is an important component of this powerful tool. Hence, this subsection intends to present some of the existing survey models that currently exists, their work method and ability to fulfil each particular problem needs.

There are already some existing models that are based on a Citizen Science project, intending to provide solutions to some problems in nowadays society for example, ruptures, accidents, or other events that are relevant to be reported. The contribute collection of data of this and other contexts pretends to create a database detailed and complex enough in order to provide a high spatially density mapping of a variety of users data, providing useful information that can be used by all parties, aiming to solve a particular problem context.

A great amount of this models resort to native and/or web applications being able to access directly the hardware of the participants device such as the GPS, camera, microphone, etc., making the user inputs extremely diverse, providing high fidelity data to the Citizen Science model, increasing the ease of the public involvement into scientific projects.

A joint initiative of the California Academy of Sciences and the National Geographic Society developed iNaturalist, an app in the scope of Citizen Science. This tools provides a place to record and organize nature findings, encouraging the participation of a wide variety of nature enthusiasts. This wide range allows iNaturalist to create extensive community awareness of local biodiversity and promote further exploration of local environments.

Despite being one of the most popular nature applications available, iNaturalist presents a system workflow that doesn't offer the required information to establish an effective correlation to the effects of the global change drivers. For example, on Figure 2.2, is represented the surveying of a lichen specie and other user results. It is just based on the lichen species identification, ignoring the abundances collection nor the species classification on functional groups. This represents a downside to this nature

findings application, only providing a geolocated marker of the identified specie.

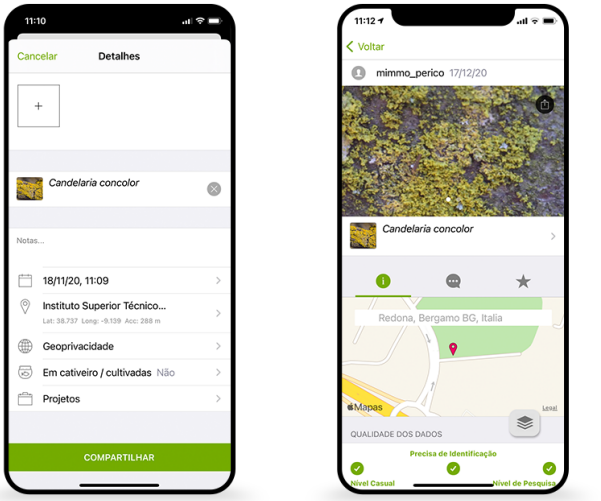


Figure 2.2: iNaturalist.

In Lisbon, in the context of the Biodiversity Cooperation Protocol, with the ambitious and pioneer objective of raising by 20% the potential of the biodiversity in the city of Lisbon until 2020 [54] it was launched a citizen science project, Figure 2.3, by the city municipality, the cE3c - Centre for Ecology, Evolution and Environmental Changes (FCUL) and the CESAM - Centre for Environmental and Marine Studies.

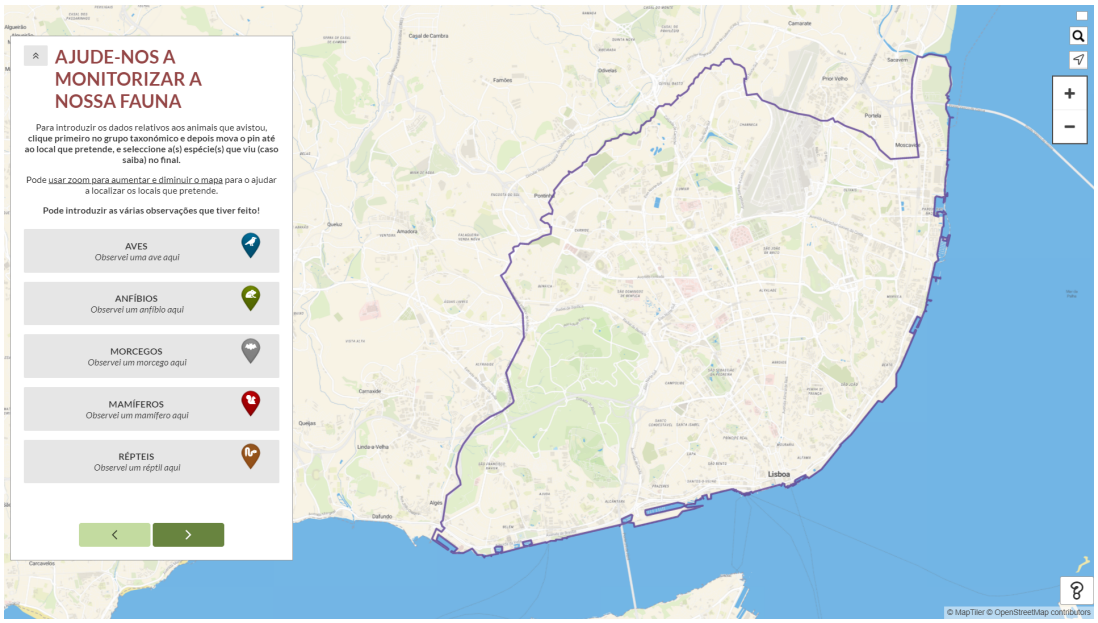


Figure 2.3: Public and online Survey in the context of Fauna Monitoring in Lisbon[55].

The goal was to understand what are the most common animal species spotted by the Lisbon citizens and in what areas were they spotted was the main motive for the online and public survey [55] launched for the fauna monitoring in the city of Lisbon. In this Citizen Science Survey, participants only need to

access the online form and identify the spotted species along with the location, using the list of species and the available map [55].

Surveying and monitoring bird populations [56] is also another area in which citizen scientists are widely interested. Several large-scale bird-monitoring projects, such as Christmas Bird Count (CBC) making use of a mobile app, Breeding Bird Survey (BBS) and Project FeederWatch (PFW) both using a developed web platform, relying on volunteers to collect data [57].

There are some other models that attend to solve specific problems where Citizen Science is the model basis to develop the pretended solution. Fulcrum [58], a mobile forms platform that enables users to build custom apps for collecting data in the field. Making usage of an intuitive drag-and-drop builder on the Fulcrum website, the user can afterwards deploy the developed survey to the field teams to complete it on their mobile devices. Data collected in the field is automatically geotagged and synced to the cloud for instant access from the office. It allows access to live data feed for real-time mapping, exportation of data and attachments in a variety of standard formats for further analysis. Fulcrum can be used as a standalone location-based data collection platform or integrated with existing services such as GIS and asset management systems.

Another models like Fulcrum are available, making usage of the Citizen Science philosophy of work. Thus, apps like Device Magic [59], Magpi [60] or FastField [61] act in a similarly way: customizable mobile forms software and data collection, even offline, using participants mobile devices. Delivering accurate data, in any pretended format, from the field to the office in real-time, making possible the later integration with another systems.

Some other apps, more narrowed than the previous ones are designed to solve more specific problems. For example there were studies made to analyze the current state of VGI as crowdsourced data for the use of geographic knowledge production in public administration in the Visegrád Group [62] and there were found apps like ZmapujTo [63], DejTip [64], T-mapy [65] among many others that, despite having slight differences in their final goals, offered people the possibility to report civic issues on municipalities via web applications and/or native apps. Out of the 23 identified applications 14 allow their users to report anonymously, while 9 applications require user registration either by filling out the registration form, or by logging-in to their Facebook or Google+ accounts. Another interesting finding is that almost one quarter of the identified applications in this study only run on mobile devices, not on websites. Each of the analyzed applications is available for Android, almost a half of the applications work on iOS, and only 7 applications are accessible on Windows Phone. One interesting finding is that only 6 of the 23 applications used web applications even with the known fact that the system would be more widespread if it was accessible through different types of OS, such as Android, iOS or Windows Phone and present in web applications.

This examples show the diversity of models and the applicability of the Citizen Science in several fields and areas. Despite all the examples previously presented, there are some negative aspects and disadvantages of the existing models that come up in to the context of the problem presented in this paper, being reported with more detail in the subsection 2.4.

2.3.5 Lichen Surveying Methodologies

Lichen diversity, proven to be a powerful ecological indicator of the global change driver effects, data collection must be done based on compatible standardized methodologies performed at regional and global scales [66]. Since the 1970s, many methodologies were considered to assess lichen diversity. Among many, one in Europe (EU) and another in the United States of America (US), two standardized methodologies for sampling lichen epiphytic diversity. "The methods differ largely in two main aspects: (i) the EU method registers all lichen species (macro and microlichens) occurring inside a size-standardized grid placed on a tree trunk in a fixed number of trees, resulting in metrics of frequency; (ii) the US method surveys all macrolichen species detected on any tree or shrub inside a large circular sampling plot, visually rating species abundance." [67]

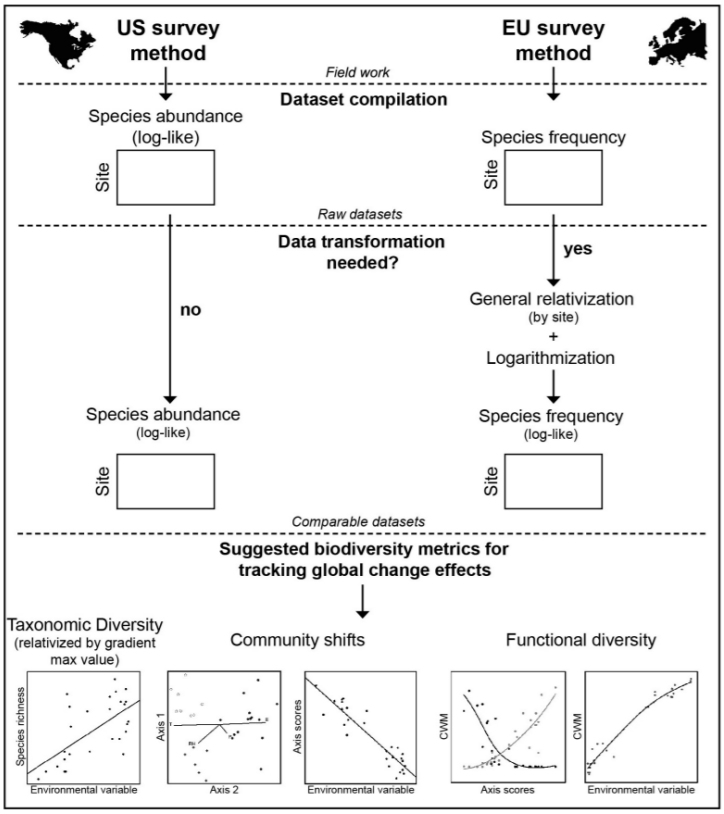


Figure 2.4: Conceptual framework for joint analysis of US and EU lichen diversity data sets for tracking global change. [67]

2.3.6 Lichen Sampling Method on a Citizen Science Scope

Similarly to the standard lichen sampling, the goal of this approach is to simplify the current methodologies to be executed by a simple citizen. This way, a simplified lichen diversity sampling methodology can be used to fulfil the surveying requirements of a crowdsourcing project.

Firstly, this simplified methodology requires a citizen to head to a tree on any location and quantify the amount of some lichen species in the main tree trunk. Each tree will be one observation, i.e. one survey.

The tree trunk must be vertical, with more than 20 cm of diameter, without brunches and between 1m and 2 m height. All tree species are welcome except those with very smooth bark or that peel easily (maples, eucalyptus, pines, poplars).

At each tree look for the aspect facing north, and looking at approximately 1.5m from the ground imagine a rectangle oriented vertically with approximately 20x30 cm (like an A4 paper). Then estimate the area occupied by each lichen species within that rectangle.

	Metrics		
Specie	Aridity	Poleotolerance	Eutrophication
Flavoparmelia Caperata	3	4	2
Evernia Prunastri	3	3	2
Xanthoria Parietina	4	4	3
Phaeophyscia Orbicularis	4	5	3
Hyperphyscia Adglutinata	4	5	3
Candelaria Concolor	4	5	3
Diploicia Canescens	3	4	2
Chrysothrix Candelaris	3	1	2
Parmotrema Hypoleucinum	2	2	1
Lecanora Allophana	4	3	2
Dendrographa Decolorans	2	3	2
Usnea Rubicunda	2	2	0
Ramalina Fastigiata	3	3	2
Physcia Adscendens	4	5	3

Table 2.1: Metric Indexes based on Lichen Species.

Thus, having set lichen sampling method and the metric indexes based on lichen species on Table 2.1 the existent variables can be processed with a set of equations in order to obtain the final Metric Index Model.

$$\alpha_i = \sum_{l=1}^{L} p(l)_i \quad , \qquad i = 1, 2, 3, 4, 5 \tag{2.1}$$

The equation 2.1 represents the sum of the percentages per metric value, where α represent each metric and i represents the metric indexes that can be found on table 2.1. After the lichen sampling method, each α will be the sum of the p occupation percentage of the found lichens, with L lichens available, that have the i metric index. Each metric index has a weight ω that is described on table 2.2.

i	ω_i
1	-5
2	-2
3	1
4	2
5	5

Table 2.2: Weight of each metric value.

$$\eta = \sum_{i=1}^{5} \alpha_i \omega_i \tag{2.2}$$

On the equation 2.2, η represents the weighted sum of the each metric α, where ω represents the weight of each i index.

$$m = \frac{\eta}{\sum_{i=1}^{5} \alpha_i} \tag{2.3}$$

On the equation 2.3, m represents the final metric index, having η representing the weighted sum and α the sum of the p occupation percentage of the found lichens that have the i metric index. This model equations can be applied to the 3 metrics highlighted on this project: aridity, poleotolerance, and eutrophication.

2.4 Discussion and Conclusion

Climate change and air pollution are a threat to public human health and will have a strong consequences in the future if no action is to be taken. Urban areas, more affected by the pollution caused by traffic roads, industrial areas and other anthropogenic sources, are becoming more and more populated and so a high density spatial mapping is necessary in order to better analyze each site, getting more detailed and accurate data from urban areas.

The lack of monitoring data stations contribute to a nowadays scarce mapping of climate change and air pollution. Lichens are good climate proxy because of their sensitivity to climate changes and so, this lack of information can be solved by using lichens as biomonitors. Some metrics like species richness, lichen cover, and lichen community com-position are some examples of metrics used to quantify the effects of atmospheric pollutants on ecosystems. The existing scarceness mapping problem can be overcome with lichens, using them as trustworthy sources of data.

Citizen Science is proven to be an effective tool that can help to increment the amount of useful data instead of having just a few stations as the only sources of data to be mapped. Using participants and their ease of access to mobile devices, they can be used as a source of data inputs, overcoming the lack of information problem and contributing to their scientific involvement.

There are some existing survey software models based on citizen science that are proven to be effective, contributing successfully to the specific problem. But this specificity proves to be a problem. The lack of extensibility and modularity becomes a disadvantage even in the drag-and-drop builder of mobile forms like Fulcrum, once this process is still made inside their developed application and so,

being a closed system, code manipulation is not possible to be done. The majority of this applications have different premium versions that enables some functionalities according to the chosen package but this still doesn't the solve the code manipulation problem because the available systems are never an open source code. The trial free versions are another negative aspect of the majority of this applications, representing cost and time penalties for the problem in context. The direct access and manipulation of the server and database is not possible in any of this case examples and so data is trusted in the original application server, putting safety and privacy of data at stake. Besides this all raised disadvantages, is important to highlight that the processing and the immediate feedback to the user that is required in the problem context proposed is not satisfied since the extensibility and modularity typical in generic systems are nonexistent in the current available models.

Therefore, based on the current needs of providing more information to the researchers in the many existent scientific scopes, a Generic Crowdsourcing System will be developed, presenting all the features required at the Chapter 3. This tool will overcome the issues raised by this thesis research phase, ensuring a generic system, capable of adapting not only to the narrowed case proposed for this thesis, but also to fulfill any Citizen Science project.

Chapter 3

Requirements

In the spectrum of the current problem that researchers are facing, a powerful generic surveying tool is needed in order to allow further developments. This framework, after being extended, will solve any problems that a specific scientific need might raise, ensuring an efficient and narrowed tool that researchers can deploy to citizens.

Based on some of the existing Software for Citizen Science Survey, present in section 2.3.4, it's possible to find common requirements that could be used in a Generic Crowdsourcing System. This section lists the requirements that are relevant for the framework, in order to develop a basic but complete and efficient GCS Tool. These requirements are going to sustain the GCS, providing adaptability that can be used to fulfil the need raised by any narrowed specific scientific need. This Chapter is divided into 4 main categories: the Functional Requirements, presenting System Entities and Framework Functionalities, Performance, Usability and Safety.

3.1 Functional Requirements

3.1.1 System Entities

As in any system, it is important to define roles and responsibilities in order to maintain an organized hierarchical structure, where permissions and privileges are managed to align among themselves in order to achieve the system final goal. There are four identified entities in this system: **Users**, **Researchers**, **Administrator** and **Programmer**, having their description and associated functionalities following described.

- **Users**

  Users are all the entities that use the Crowdsourcing Platform. Being able to register with a name, unique email and a password, users can afterwards perform the system login with the respective credentials. Having access to the Generic Crowdsourcing Menu, despite their scientific knowledge in the area in question, users can submit data through the Crowdsourcing Form that, according to the problem in context, present various combinations of surveying components. In the same main

menu, Users can access the Profile Section, presenting users' name and profile picture, being able to edit them. At last, Users can access to the Results Section still on the main menu, presenting the output of the Crowdsourcing Platform that will vary according to the problem in context. After the platform usage, users can them perform a logout action.

- **Researchers**

  Researchers are **Users** with advanced scientific knowledge in the field and so, naturally, with the possibility to insert data in the system. Responsible to maintain the outputs of the system reliable and trustworthy, researchers have an extra responsibility to doubt geolocated fluctuation since, being a crowdsourcing platform, surveys are submitted regardless of the user scientific knowledge. Thus, having access to all raw data used for processing is imperative, allowing researchers to delete and/or modify data if needed.

- **Administrator**

  Responsible for the system management, granting access and authorizations to third parties, defining types of data access and validation permissions. Having access to the full list of users, the administrator is responsible to select the users that are considered to be researchers, grating them all the permissions assigned to the role. All the platform change suggestions idealized by the **Users** must be sent directly to the Administrator to be considered and, if approved, sent to the **Programmer** to be handled.

- **Programmer**

  Has access to the source code and system data, granted by the **Administrator**, having visualization and/or editing permission status. The Administrator, the only entity that the programmer responds to, can ask for system modifications and so the programmer, making use of the generic extensible and modular source code, can add/remove functionalities and modify the existing ones. Responsible for bug detection and system safety must ensure data protection, preventing incorrect data injections and normalize malfunctioning variables.

3.1.2 Framework Functionalities

Based on the current needs for a Crowdsourcing Campaign, there are some framework functionalities that, despite the subjectivity of the matter, are considered transverse and relevant for the success of any Citizen Science Tool. The following modules are considered to be Generic Requirements for a GCS but, nevertheless, is important to highlight the extensibility that they can provide if pretended.

Registration and Login

This module, having a crucial role at defining System Entities 3.1.1 in the Generic Crowdsourcing System, it allows the management of the system users, grating them different accesses according to the respective role. The Registration process corresponds to the submission of the user credentials: name,

email and password. This data will be saved on a database and then, in the Login Process, access will be granted after validation of the submitted email and password credentials. In order to limit the login time of the user, its email is going to be inserted in a general-purpose distributed memory-caching system, normally used for small chunks of arbitrary data, eliminating user credentials after a login time limit is reached, blocking any actions from user, forcing the user to re-login if needed.

In order to reduce the user resistance to register or login this module needs to offer, besides the manual method, the possibility to authenticate through other digital platforms like Google or Facebook. This authentication would have no impact of the platform usage, grating the same benefits as the manual method users. This authentication would provide the user email to the platform, using it as the unique user identifier just like the register manual method.

- **R1 - The framework should allow the registration of users.**
- **R2 - The framework should allow the login of users.**
- **R3 - The user authentication should be made natively or using third party applications.**

Profile

To consolidate the relation between crowdsourcing system and user, a Generic Profile module could be one of the key factors to establish a greater commitment. The email, being a unique identifier of the user, is not allowed to be changed in this module but non less important information could be modified. User photo and name are the ones considered to be generic fields that are relevant to be displayed and modified in this module. This module accepts also extensibility, being able to handle a Ranking Mechanism or a Profile Extension. This last one, not used in this work, pretends to create a custom profile section that, after being loaded to the project folder, it will be appended bellow the Generic Profile Section. This extensibility makes this module a complete and adjustable tool to handle users' personal information.

- **R4 - The framework should store a Profile of the user.**
- **R5 - The framework should allow the extensibility of the Profile.**

Ranking System

In order to retain users' attention and commitment the Ranking Extension is a module that is based on a user points system. Based on their actions, in a crowdsourcing framework - the form submission, users receive a certain amount of points associated with their profile. The amount of user points is then loaded and displayed on the Profile, Subsection 3.1.2. The user ranking level, also displayed on the Profile Section, is a mechanism that according to the amount of user points, hooks a user to a certain level. This pretends to introduce a basic gamification to the GCS with the aim of improving the user experience in the surveying process and to bring new users to the platform.

- **R6 - The framework should allow the implementation of a Ranking system.**

Surveys

This module is considered to be one of the most important of the Generic Crowdsourcing System for having the responsibility to fetch data from the users' input, submitting it for later handling. According to the researched needs on the Citizen Science projects there are two types of surveys distinguished and considered to be an important requirement for the generic tool: static and dynamic surveys. The first, contrasting to the dynamic survey is independent from the users inputs. This means that the dynamic surveys progressively adapt to the survey answer submissions, changing constantly to the user scope. Regardless of the type survey chosen to a specific scientific scope, a module is needed to handle this form configuration file. To make this module adaptable to any scientific crowdsourcing need, a file can be uploaded to the system, containing all the structure and information needed to render a simple UI survey to be presented to the user, allowing corrections of the input file to be made without negative impacts on the system data flow.

Due to the importance of this simple workflow between an input file containing the form structure and its render into a simple UI, a library must be developed and deployed. The existing downloadable libraries don't present extensibility and are very narrowed to be handled in a specific way. Therefore this library would be used by the Generic Crowdsourcing System and both would be available to be used by the general public. This represents a great value to the research community, contributing to a positive impact on Science.

The rendered Survey, regardless of their type and the chosen components of its constitution, aggregates the user inputs into a data object to be sent to the server. This object should be handled and stored on a database.

- **R7 - The framework should allow a surveying mechanism.**
- **R8 - The framework should provide static or dynamic surveys.**
- **R9 - The framework should use a external library that renders surveys and handles data, while providing extensibility.**

Data Validation

Due to the responsibility given to the citizens, regardless of their scientific knowledge, data supervision is one aspect that must be taken in account. Error rate, specially taking account a crowdsourcing campaign, is a non disposable aspect that must be taken in consideration to make the system as accurate and trustworthy as possible. Thus, researchers, having the scientific knowledge, capableness and responsibility towards the system accurateness and reliability, should have a platform for data validation, providing raw inputs visualization, with a filtering mechanism, being able to delete data considered to be incorrect.

- **R10 - The framework should allow data validation.**

Data Download

Researchers use data to be computed, processed, stored or handled by other external frameworks. The Results module, displaying simplified information to the user, can be not enough to the scientific community analysis. Thus, GCS must provide a Data Download component that allows the extraction of detailed information from the framework. This component must be extendable, allowing to define what data fields are relevant for the researchers community according to the scientific scope.

- **R11 - The framework should allow data download.**

Results

This module consists on the results presented by the Generic Crowdsourcing System. This results vary according to the crowdsourcing context but are always based on the processing of the data surveyed on the platform. The feedback can be simple or complex. The last one is fetched from the server after the data processing, being displayed to the user in the best way possible. The Results Module is relevant for the survey tool because it provides information directly related to the user inputs, contributing to the users' knowledge and commitment to the scientific process.

The Results Component is the feature of the Generic Crowdsourcing System responsible for displaying the required Feedback for the client interface. There are two types of feedback: the immediate, in its definition, and the differentiated. The first requirement is for the system to have a simple response from the server, not demanding data process. The second requirement is the differentiated feedback, giving the possibility of the GCS to return processed data from the server based on the Survey user inputs. The processing phase of this feedback mechanism runs independently on the server, not demanding the client application to wait for the end. It runs modular differentiated components, handling data sequentially, storing it for further purposes.

- **R12 - The framework should allow immediate and differentiated feedback.**
- **R13 - The framework should allow display the processing results to the user.**

Help Guide

Using a horizontal scroll mechanism, this module pretends to provide user support at any time of the platform usage. Despite the simplicity of the extended tool, the user surveying process must be done correctly, following the method standard lichen sampling method considered in Section 2.3.6. This method, although being simple enough process to a non scientifically informed user, must be explained carefully and accurately, justifying the urge of this module existence having as the main objective the error rate decrease.

- **R14 - The framework should allow a Help Guide Presentation.**

3.2 Performance

This requirement is related to the efficiency and quality of the execution of the computing technology and the availability of adequate hardware resources.

Thus, is important to reduce the size of the transmitted data, avoiding unnecessary reloads, overloading the server with too many requests.

Caching service on the server side enhances the system performance speeding up loading and minimizing system resources needed to load data.

Distributed architecture application is also a performance requirement having task distribution on several machines, balancing the system load.

Due to the Citizen Science scope of this work, a simplification of the software design is required. The simpler the system architecture and implementation from the very beginning the easier it will be for programmers to extend it and to maintain it.

One of the major performance requirement from the Generic Crowdsourcing System is the singular code base of this tool. This means that by extending and adapting a collection of source code it is possible to build a particular software system, application, or software component.

Another performance requirement is after-deployment monitoring. Performance testing is an important part of product life cycle. This also includes bugs and feedback reported from users, helping to maintain the system stable and efficient.

The amount of loading time must be short, providing a fast initialization of the application itself and other components or hardware sensors required at usage time.

Being deployed to any user, the storage available in each smartphone depends. Thus, the application must be low weight, not requiring much mobile resources.

- **R15 - The framework should allow database and caching service.**
- **R16 - The framework should be a singular source code.**
- **R17 - The framework should be tested and monitored.**
- **R18 - The framework should be low weight, simple, fast and understandable for the programmer.**

3.3 Usability

In order to achieve objectives with effectiveness, efficiency, and satisfaction in a quantified context of use, this sections pretends to highlight the usability of the Generic Crowdsourcing System. The designed User Interface must be simple and intuitive to the users. It also must provide support in its usage, avoiding misinterpretations or hesitations. Citizen Science, consisting on wide range of target users where every citizen can have a role, is an important factor considering the crowdsourcing component of the future developed tool. Thus, the mobile devices of the citizens are an important drive force of the GCS. It must be developed for both iOS and Android, making it available for both platforms. The

developed tool must be equal for both platforms, allowing the same usage of the same tool features. In the usability scope, smartphone dimensions and other relevant specifications, adapting the Generic Surveying Tool to provide a dynamic interface that shapes itself to the user mobile device.

- **R19 - The framework should be user friendly.**
- **R20 - The framework should be compatible with Android and iOS.**

3.4 Security

More than ever, privacy and safety are a key factor in software systems. It is important to develop a system where user credentials are safe and internal connections in the system are not affected. Password encryption is one effective way of handling this safety. Native libraries from different language technologies provide efficient tools of encryption. Furthermore, there are some technology environments that provide better choices when it comes to safety. Thus, the Generic Crowdsourcing System must be able to use the best available technology choices, ensuring that data is not compromised or users feel safe upon the GCS usage.

- **R21 - The framework should be prepared for external attacks.**
- **R22 - The framework should preserve the user privacy.**

Chapter 4

Architecture

There are some existing crowdsourcing systems that present efficient and diverse functionalities but their model structure, regardless of their effectiveness, is very specific to each problem. This section based on the Requirements, presented on Chapter 3, intends to expose the structure of the system based on the system needs while hiding the implementation details. It refers the fundamental structures of a software system, each comprising software elements, relations among them, and properties of both elements and relations.

The Architecture of the GCS was done independently from the scientific scope of this thesis, being able to fulfil the generic requirements for this framework. Nevertheless, a good proof of concept of the extensibility of this framework is the success of a developed tool that tackles the problem proposed by the Centre for Ecology, Evolution and Environmental Changes (cE3c) - FCUL, supported by iLTER. The future extended system, an app-based interface named **eFlechten**, will allow lichen diversity sampling campaigns aligned with citizen science conducts, processing user data in a remote server, providing immediate feedback from this data handling and, ultimately, its presentation to the user as a map, portraying the effects of air pollution and climate change. Thus, having established the bridge between the generic and the extension requirements for this work, the architectural model must adapt to this bound considering the extensibility mechanism intended.

Driven by the C4 Model[68] approach, used by somewhere over 10,000 people in more than 30 countries, the software system architecture is diagrammed in an ”abstraction-first” model, based upon abstractions that reflect how software architects and developers think about and how software is built. This model considers the static structures of a software system in 4 distinct levels: Context , Container, Component and Code diagrams. The last level, the Coding diagram will be represented on the Implementation, on Chapter 5.

4.1 System Context Diagram

This Crowdsourcing Platform Architecture will adapt to the common needs of the different scientific fields ensuring a multi functional, extensible system that can be implemented. On Figure 4.1 is represented the system architecture in the highest level of software abstraction, the first level of the C4 Model, highlighting system entities (actors, roles, personas, etc), software systems and their relations rather than technologies, protocols and other low-level details.

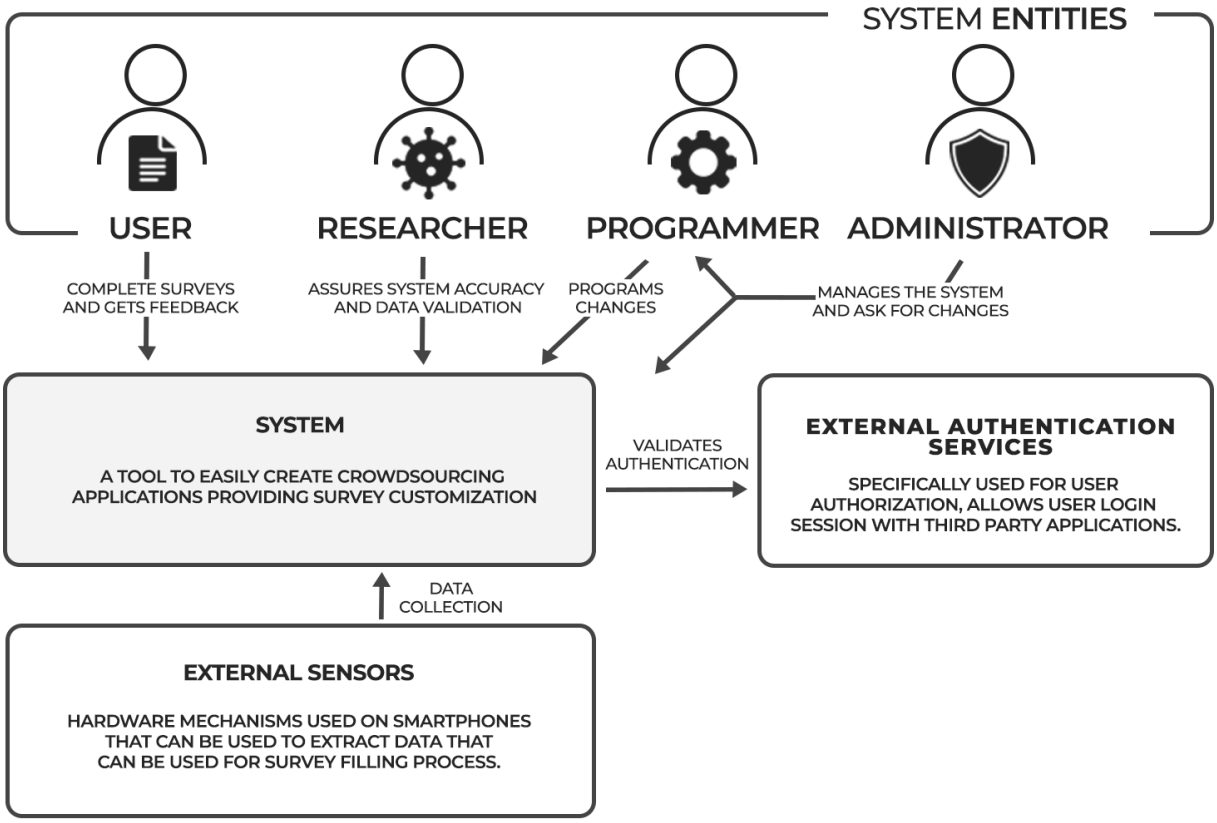


Figure 4.1: Level 1: System Context Diagram.

Four system entities are identified: Users, Researchers, Administrator and the Programmer. Users are the ones responsible for completing surveys, receiving the feedback generated based on their inputs. Researchers, that can use the platform like normal Users, are responsible for ensuring data reliability, identifying possible problems or errors and proceeding to their adjustment or repair. Administrator manages the system entities, defining which users are considered to be researchers, and the system performance, demanding the Programmer to proceed to system software amendments.

The System, providing generic crowdsourcing features embraces several containers to be detailed in the following level. It is composed by several components and modules that are going to be inherited to any extended system application, allowing further modifications. It is a cohesive and efficient framework that is not isolated, allowing API calls to be made.

The Context diagram also defines external systems that are directly connected to the software system in scope. External Authentication Services and External Sensors are represented in the diagram as well as the relations with the main software system.

The External Authentication Services allows GCS users to authenticate with external third party applications. This service safely validates the user credentials, simplifying the process of registration or login into the system.

The External Sensors are composed by all the hardware mechanisms used on users smartphones or other devices. This allows external data extraction from the wide range of technologies available, constituting another source of information that can be attached to the data flow in the system.

4.2 Container Diagram

The next level in the C4 Model is the Container Diagram, where a container is considered to be a separately runnable/deployable unit that executes code or stores data, like a server or a mobile application. It also shows the major technology choices and how the containers communicate with one another. With a lower degree of abstraction, the Container Diagram is represented the Container Diagram on Figure 4.2. This diagram shows the high-level shape of the software architecture and how responsibilities are distributed across it.

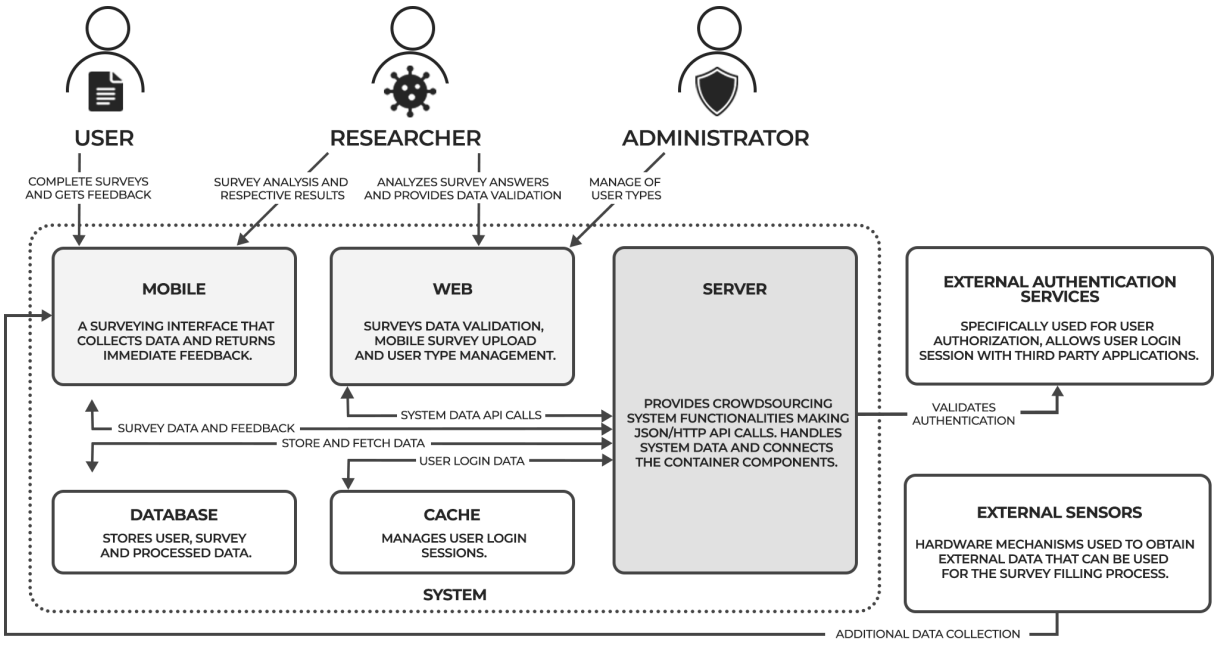


Figure 4.2: Level 2: Container Diagram.

Focusing on the single software system it is possible to detail different containers on Figure 4.2. The client interfaces are represented by the mobile container and the web container. The Server, its data services and the external services that connect to the system are also represented.

The Mobile container is extremely important once it connects the user actions with the system. It

allows interactions between components, collecting data with surveys and sending it to the server. Receiving data from the server, the mobile application handles the received information, rendering it to the user in a simplified and interactive way.

The Web container is a complementary interface built to enhance system performance and maintenance. It allows the upload of surveys to be rendered on the mobile application, provides the management of the user types and allows to delete data, answers or even users. For further complex analysis, a data download mechanism is provided, allowing researchers to use it in external applications.

The Server container provides the GCS processing features, allowing client interfaces to correctly display information on rendered screens. It handles all the API calls, retrieving the correct response, even if an error occurs. It also manages data, sending it and fetching it with data storage services.

The Container diagram also contains data storage services: database, a repository that supports an application's back-end data, and cache, that acts as an adjacent data access layer. Both communicate with the Server, bidirectionally exchanging data and user credentials.

The User, the most important role on a Citizen Science project only interacts with the mobile application since it is the interface chosen to fulfill the surveying process. It is responsible for the field work, submitting information that is useful for researcher analysis. It will also receive the feedback from the system, having access to the results screen that will render processed data.

The Researcher can submit surveys and visualize the results, having access to the mobile application provided with the exact same features presented to a normal user. Being scientifically experienced in the area, Researchers can analyze minutely the results, tracking possible inaccuracies. Thus, having an extra responsibility to the reliability of the system, they can access to the web application. In this interface, researchers can interact with the data validation and download, excluding invalid user inputs and having the possibility to extract a file containing all the survey answers.

The Administrator is responsible for ensuring all the system containers are responding correctly, maintaining the efficiency of the GCS. It can also manage user types, having the power to grant or remove privileges from users toggling between research or normal user type. If a user is considered to be disposable, it can be removed by the Administrator. It is also responsible to demand the programmer to perform system coding modifications of new features or to fix bugs.

Lastly, the Programmer, that can be an external entity, is not represented in this diagram but, nevertheless, it can provide its service to any container accordingly to the instructions of the Administrator.

As in the previous level of the C4 model, the External Sensors and the External Authentication Services are also represented in this diagram to represent the relations with the displayed containers.

4.3 Component Diagram

In the third level of the C4 model, each container presented in the previous level is decomposed, revealing the major structural building blocks and their interactions. Mobile Application, Server and Web Application will be detailed in the following subsections, each presenting a component diagram, showing the most relevant ”components” of a container, what each of those components are, their responsibilities.

The Generic Crowdsourcing System development had the contribution of another student of Masters in Electrical and Computer Engineering, Miguel Leitão, developing its Msc thesis regarding Urban Green Spaces, in order to map UGS while assessing urban dewellers preferences within these spaces. After development, the Generic Crowdsourcing System will be used in both thesis in order to prove the extensibility and the adaptability of the developed tool to any crowdsourcing context.

Thus, having different crowdsourcing needs to each thesis, the next subsections will present the 3 Component Diagrams where is highlighted in green the component blocks developed to this thesis and in red the ones developed by Miguel Leitão. The generic system developed will thereupon satisfy both thesis, proving the system extensibility by the efficiency of the final tools.

4.3.1 Mobile Component Diagram

The mobile component is a surveying interface tool that interacts with the user providing a feedback mechanism. Figure 4.3 represents the Component Diagram of the Mobile application detailing the architectural blocks that represent the functionality components of this interface.

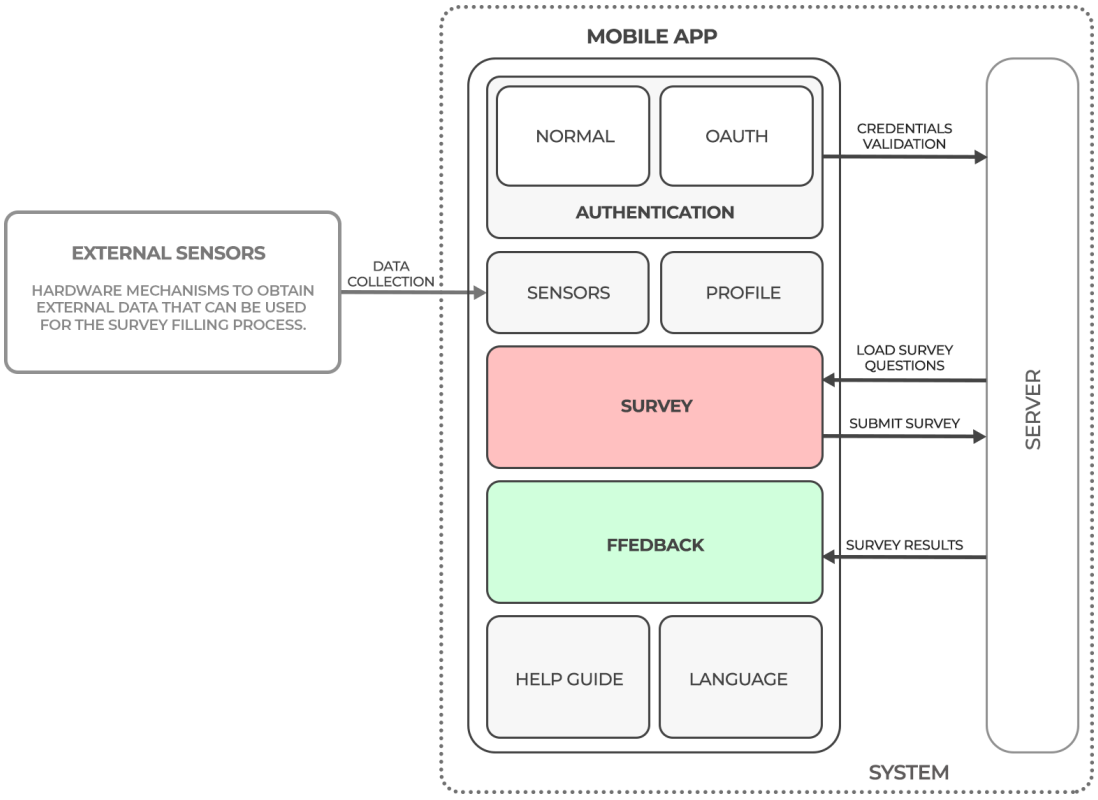


Figure 4.3: Level 3: Component Diagram - Mobile Application.

The Authentication component provides a normal mode and a third party authentication mode, OAuth. This component offers a register service, allowing any person to use the crowdsourcing system. After registration, the user can login and use the mobile client interface. Both registration and login methods send user credentials through a JSON file using HTTP requests using a REST API to the server, to be validated and stored. Besides the components to be detailed in this subsection, the user can access its profile page.

The Survey component loads questions from the Server that can be dynamic or static. The first option refers to the survey questions that are dependent from the previous user survey answers. On the other hand, the second option displays the same questions to all users regardless of their previous inputs defining a static surveying system. The survey questions are sent from the server to the client in JSON files using HTTP calls using REST requests. This JSON files are then used to render a simple UI Survey, to be displayed in the mobile application to the user. After the survey filling process, the user inputs are then submitted and sent to the Server with REST requests using a JSON format. This user data will then be handled and stored by the server for later processing or validation.

After processing, the Server returns survey processed or unprocessed data back to the mobile application, the Feedback module. This component receives a JSON file as a server response to its request, containing the survey feedback. This file is then handled by the mobile application, rendering a simple UI that can display the received feedback to the user.

The Language Component that despite being isolated from the server, offers system advantages in terms of user adaptability. It configures the system to display to the user a certain requested language, not affecting the system behavior.

Another isolated module that offers user adaptability, the Help Guide component is also present in this Mobile Application Component Diagram. It displays to the user a tutorial with a set of information relevant to the system usage, avoiding dubiety and misapprehension.

4.3.2 Server Component Diagram

The Server provides Crowdsourcing functionalities unifying the different components of this tool in order to make a cohesive system. It establishes the bridge between the user and data storing services through REST API calls. Data can be processed with this system component, generating survey feedback that can be stored and/or sent to the client interface. Figure 4.4 represents the Component Diagram of the Server, detailing the architectural inside blocks that represent the functionality components of this interface.

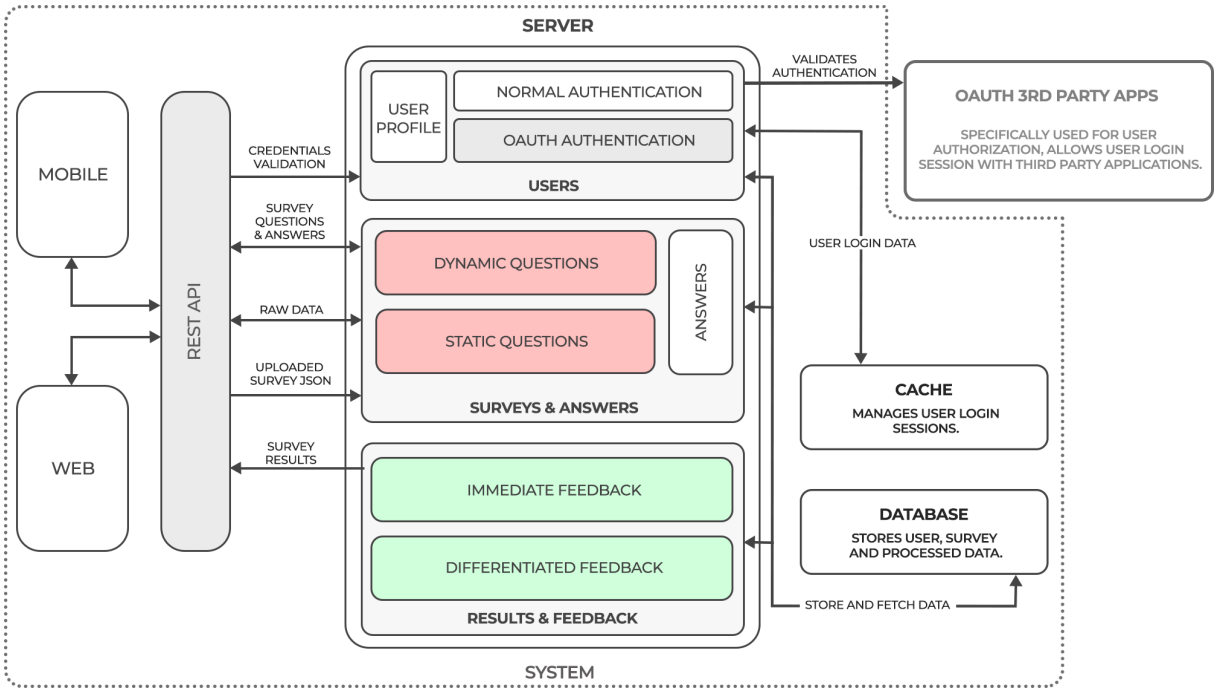


Figure 4.4: Level 3: Component Diagram - Server.

Both client interfaces, Mobile and Web, require authentication, establishing a HTTP request using a REST API to communicate with the Server. In this call credentials are sent or handled, having the possibility of activating the normal authentication or using OAuth. This last one allows user to login simply with a third party app making a request to an external OAuth Server, easing the registration/login methods without affecting the system behavior.

Surveys and Answers are the Server component responsible for sending the survey to be presented on the mobile application and handle the user inputs after submission. The Surveys to be presented on the mobile application can be static or dynamic. The first option displays the same questions to all users regardless of their previous inputs. On the other hand, the second option refers to the survey questions that are dependent from the previous user survey answers, defining a dynamic surveying system. This questions are generated by the JSON file, fetched from the database, that was initially sent from the web application in the Survey Submission module. Surveys and Answers are also the component responsible for handling the survey answers, storing user inputs. After the data storage process, Data Validation from

the web application can request it and perform any necessary modifications directly. After validation, data is able to be processed, generating survey results and providing feedback to the user.

Results and Feedback are the Server component that returns survey processed data in the form of immediate or differentiated feedback. The first is a simple feedback object that is sent as a response to the REST request made by the client. On the other hand, the differentiated feedback is a mechanism to handle Survey user inputs in a sequence of processing modules. These processing modules will run asynchronously, storing data throughout the process, being sent to the mobile application to be displayed to the user.

4.3.3 Web Component Diagram

The Web Application complements the system, adding an extra UI layer to the Researchers and the Administrator. Figure 4.5 represents the Component Diagram of the Web Application, detailing the architectural inside blocks that represent the functionality components of this interface.

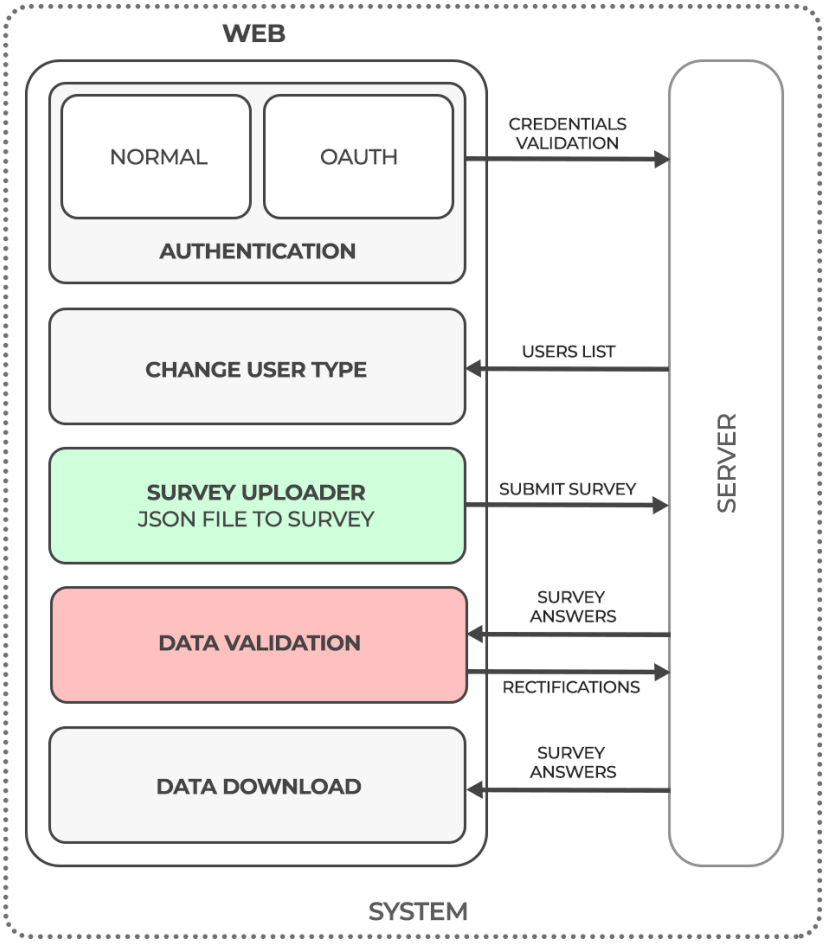


Figure 4.5: Level 3: Component Diagram - Web Application.

The Authentication component, just like the mobile application provides a normal mode and a third party authentication mode, OAuth. This component offers a register service, allowing any person to use the crowdsourcing system. After registration, the user can login and use the mobile client interface. Both

registration and login methods send user credentials through a JSON file using HTTP requests using a REST API to the server, to be validated and stored. Besides the components to be detailed in this subsection, the user can access its profile page.

If the Administrator is logged in, the web application, through an HTTP request with a REST API, will receive a user list. For each user on that list, the Administrator can toggle user types between normal users or researchers. This allows to grant or remove privileges from already registered users. Furthermore, the Administrator can delete users from the system. Thus, with the Web Application, the administrator has some user management control, granting permissions to whom is selected or deleting users if considered to be disposable.

If the Researcher is logged in, with Survey Uploader module, there is the possibility to upload a JSON file that will generate a survey to be displayed in the Mobile Application. This JSON file will be sent using a HTTP request to the Server that will handle the received data. This component allows survey extensibility to the crowdsourcing system, attributing that responsibility to the Researcher. Furthermore Researchers, with the Data Validation component, can visualize all the user survey inputs. This list can be analyzed and validated by this system entity. The modifications are applied and saved by a REST call to the Server, handling data modifications. This module ensures data reliability to the crowdsourcing system, attributing that responsibility to the Researcher. Lastly the Researcher can extract the user survey answers with the Data Download module from the Web application. This feature allows researchers to obtain certain data fields generated by the GCS in order to fulfil the external analysis requirements.

Chapter 5

Implementation

Having established the Architecture of the Generic Crowdsourcing System , the fundamental or high level structure is outlined. It merges technical and operational requirements, structuring the components considering application quality, maintenance, performance and overall success.

Afterwards, the Generic Crowdsourcing Implementation pretends to detail how the several components described on the previous chapter are going to be implemented, relating technologies with requirements, defining components, modules, functions, endpoints and API calls.

Since the architectured framework compel the different system main components: server, mobile and web application to establish several bidirectional communications and share equally structured features, the best approach found to describe this chapter is by describing features. This approach also eases the correlation between this chapter and the Requirements, Chapter 3.

All the implemented features that will be described in this chapter belong to the Generic Crowdsourcing System. Most of these features can be extended and adapted to any crowdsourcing system application, inheriting the core implementation that will be detailed on Section 5.2.

5.1 Development Environment

One of the first crucial aspects of the Implementation phase of a software system is the decision of the development environment to be used. According to Brijendra Singh & SP Kannojia [69] there are some factors that are relevant to the success of a software project like cost, cycle time and predictability. They concluded that all characteristics of quality software are not applicable to all programming languages. Real time languages, program structures, ability to execute several actions simultaneously are crucial aspects that vary according to the existing programming language, concluding that is critical to ”choose the right programming language for a particular intended use to achieve the quality of software product”[69].

The primary scripting language for web browsers is JavaScript, highlighting its importance for modern applications. According to StackOverflow, JavaScript is the top leader of the programming languages worldwide, having 69% of professional developers prefer JS for development [70]. Being interpreted

as a lightweight programming language with object-oriented capabilities, it uses prototype objects to model inheritance [71]. JavaScript, whose implementation allows a client-side script to interact with a user and to make dynamic pages can also be used on the server side, easing the interoperability of the developed system. Considering all the mentioned characteristics, JS is a minimalist and easy programming language that is often used in libraries and frameworks. From the long list of JS libraries and frameworks there are some that deserve special highlight for the system do be developed: React.js and React Native, by Facebook, Node.js and Express.js.

5.1.1 React Native

React Native combines the best parts of native development with React, a best-in-class JavaScript library for building user interfaces. Developed by Facebook, React Native is a JavaScript framework for writing real, natively rendering mobile applications for iOS and Android. It's based on React, Facebook's JavaScript library for building user interfaces, but instead of web applications, it targets mobile platforms.

Cross-platform development is developing custom software that is meant to work on multiple platforms and software environments React components wrap existing native code and interact with native APIs via React's declarative UI paradigm and JavaScript. This enables native app development for whole new teams of developers, and can let existing native teams work much faster.

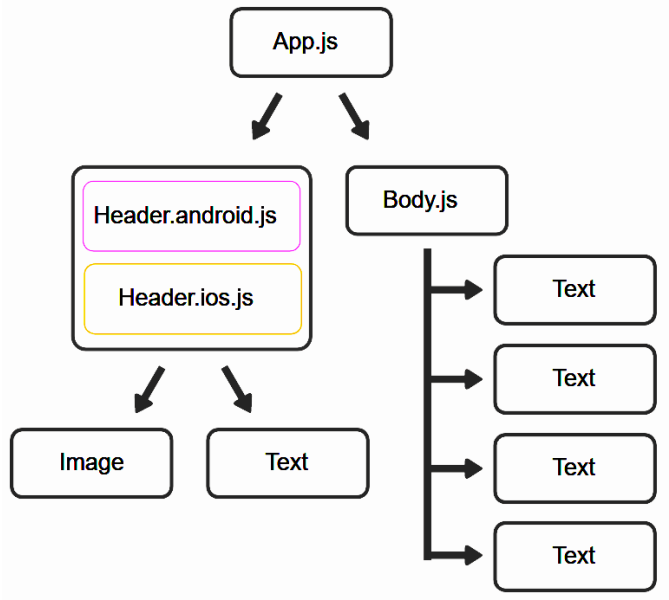


Figure 5.1: React Native Seamless Cross-Platform Scheme [72]

This cross-platform feature ensures stability to the app and is ideal for a transposable architecture. It also ensures the app performs optimally without having costs in time and performance.

The accelerated React Native mobile development is partially derived from the pre-built components present in React Native. It provides a core set of platform agnostic native components like View, Text, and Image that map directly to the platform's native UI building blocks. Besides the range of React Native's Core Components, this powerful framework allows nesting these components inside each other to

create new components. These nestable, reusable components are at the heart of the React paradigm.

React Native also provides a fast refresh that reloads the app in the React Native app development stage itself. Changes made by the developer can be seen through live-tracking. Real-time data is fetched and an updated UI also will be generated. This framework is built to prevent waiting time on native builds, improving the development time by reloading the application automatically as the code changes.

For the benefit of security, most of the existing frameworks are not allowed to use third-Party plugins. This could be a great advantage but in some cases this represents a flaw in the framework potential of functionality or usability. React Native, on the other hand, empowers developers to append the Third-Party Component Libraries and UI toolkit as well. This is possible due to the flexible platform that React Native offers, with pragmatic interfaces powered with customization options.

Lastly, by targetting iOS and Android users at the same time providing code reusability, React Native allows a widens reach, proving to be a very helpful framework because it enables companies to access a larger audience. Furthermore, being an open source tool, creates a larger native community, free access to documentation and individual experiences which is an asset for developers.

5.1.2 React.js

Also known as ReactJS, is an open-source, front end JavaScript framework for building user interfaces or UI components on web applications. In terms of websites and web applications, UIs are the collection of on-screen menus, search bars, buttons, and anything else someone interacts with in a website application. It allows developers to create large web applications that can change data, without reloading the page. The main purpose of React is to be fast, scalable, and simple in the development of web applications.

Firstly, the React framework is structured to describe User Interfaces using components. This feature allows web development to use components just like functions, inserting states and proprieties as inputs and having the UI's as outputs. This feature also allows components reuse in multiple User Interfaces as well as the usage of components that contain other components.

With React is possible to represent HTML using JavaScript. This is made by generating HTML that depends on data rather than enhancing HTML to handle that data. This JavaScript feature makes it easy for React to keep a virtual representation of HTML in memory (which is commonly known as The Virtual DOM). React uses DOM to virtually render an HTML tree and so, instead of writing a whole new tree with modifications, it only writes the differences of the new tree with every state change. This feature allows ReactJS to always have a virtual representation of views in memory. Plus, when the state of a component changes React makes the User Interface to represent that changes aswell. All this described features makes React a good choice for web front-end development.

5.1.3 Node.js

Node.js, an asynchronous event-driven JavaScript runtime environment, is designed to build scalable network applications. Normally used by developers for backend work, Node.js is an open-source cross-platform framework that handles data updates from the front end and builds scalable network applications able to process many simultaneous user requests.

In the context of backend, synchronous processing assumes that code is executed in a sequence. JavaScript execution in Node.js is single threaded, making use of the event loop and callbacks for I/O operations. Any code that is expected to run in a concurrent manner must allow the event loop to continue running as non-JavaScript operations, like I/O, are occurring. Asynchronous processing allows requests to be processed without blocking (non-blocking I/O) the thread. So after a request is processed, non-blocking asynchronous operations can be adopted, allowing other requests to be handled. This is a significant difference in capacity just by choosing to use non-blocking methods instead of blocking methods. Thus, this concurrency strategy helps Node.js make the most of single threading, resulting in short response time and concurrent processing [73]. Node's scalability is then achieved by the load balancing and the capability to handle a huge number of concurrent connections.

There are some backend technologies that can use JSON format for communication but Node.js does it without converting between binary models, using JavaScript. This is especially handy when using RESTful APIs for database support, like MongoDB. This seamless communication with one of the main data transfer standards is another advantage of the Node.js environment.

Being one of the most prominent open-source runtime environments brought about an exponential growth in the usage of frameworks built specifically for the JavaScript community to facilitate quick prototyping. A Node.js framework is just some abstract design, built out of Node.js, that embodies the control flow of the given framework's design.

Express.js

Since it is lightweight, Express.js is minimalist web framework for Node.js that is incredibly fast, behaving like a middleware to help manage system servers and routes.

The asynchronous nature of Node.js makes this framework a light-weight application that can process more than a single request seamlessly. The primary usage of it is creating Restful API's that, using JSON as a transport data format eases and enhances the JavaScript environment performance. Using a REST API using Express.js allows users to configure routes to send/receive requests between the front-end and the database (acting as a HTTP server framework). In addition, this framework comes with security features and error handling provisions, so developers can easily use it for crafting enterprise-level or browser based applications.

Axios

Axios is a Promised-based JavaScript library that is used to send HTTP requests. This modern library is usually used to send HTTP requests and handle their responses using JavaScript's promises.

In order to avoid old practices like having to rely on callbacks or deprecated modules like request(), Axios has been an elegant and modern way to handle HTTP requests in Node.js.

In opposition to fetch() that needs two steps to handle JSON, one for the actual request and the second for the response using json(), Axios only needs one. It performs automatic transforms of JSON data, being a more efficient choice for Node.js.

5.1.4 MongoDB

NoSQL databases, that means ”not only SQL”, are non tabular, storing data differently than relational tables. MongoDB is consistently ranked as the world's most popular NoSQL database. MongoDB is a general purpose, document-based, distributed database built for modern application developers and for the cloud era [74].

Cloud computing also rose in popularity, and developers began using public clouds to host their applications and data. The ability to distribute data across multiple servers and regions to make their applications resilient, to scale-out instead of scale-up, and to intelligent geo-place data are some capabilities provided by MongoDB.

MongoDB also has flexible document schemas that allows virtually any kind of data structure to be modeled and manipulated easily. MongoDB stores data in BSON format both internally, and over the network. BSON, standing for “Binary JSON,” is a binary structure that results from the encoding of JSON-like documents, allowing MongoDB parse it more efficiently.

MongoDB supports creating explicit schemas and validating data so it doesn't get out of control, but this flexibility is an incredible asset when handling real-world data, and handling changes in requirements or environment[74].

5.1.5 Memcached

Memcached is a free and open source, high-performance, distributed memory object caching system, generic in nature, but intended for use in speeding up dynamic web applications by alleviating database load [75]. This caching system is an in-memory key-value store for small chunks of arbitrary data from results of database or API calls.

Memcached is a popular choice among application developers, designed to provide the sub-millisecond latency and scale required to manage session data such as user credentials and session state.

5.1.6 MERN

MERN is the acronym for MongoDB, Express JS, React and Node JS. Is a combination of the above technologies, all based on JavaScript, used to build advanced applications. It is an open source full stack development framework i.e. it provides entire front-end to back-end development components.

With MongoDB as a database for higher scalability, Express JS for speed enhancements, JavaScript as its primary language for end-to-end development, MERN is one of the best full stack development.

React JS is the best when it comes to UI layer abstraction. It provides the best-in-class tools for faster code development. While React is only a library, it gives freedom to build the application and code organization, by providing the necessary tools.

MongoDB was designed to store JSON data natively and it works extremely well with Node.js, storing, manipulating, easing the JSON data representation at every tier of the application. The whole system, including the front-end, the back-end and the database, uses the REST API, which acts as a 'middleware' and is reusable for any other application. Having a RESTful system, all the MERN components are based on HTTP imitating web communication styles, making them very advantageous to use in the MERN stack.

5.1.7 Expo

Expo is an open-source platform for making universal native apps for Android, iOS, and the web with JavaScript and React. It is a set of tools and services built around React Native and native platforms that help you develop, build, deploy, and quickly iterate on iOS, Android, and web apps from the same JavaScript/TypeScript codebase [76].

It allows the creation of new projects, developing the architectured app: running the project server, viewing logs allowing to open the app in development on a simulator. To preserve the React Native hot reload Expo client is a perfect build and production environment once the code changes are reflected instantly in the application simulator.

Regardless of the mobile operating system, iOS or Android, Expo app provides progress monitoring of applications and test new features while the apps are being build. Developers can publish new versions of the app which then will be available on their own devices through the Expo app in run time.

Lastly it allows publishing the developed JavaScript app as well as new versions that may occur when fixing bugs or adding system features.

5.2 Generic Crowdsourcing System Implementation

This section intends to describe the Generic Crowdsourcing Implementation, defining how the system should be built, ensuring that the information system is operational and meets quality standard. The GCS Implementation describes the execution of the Architectural model, described on Chapter 4, that falls within the Development Environment scope, described on the Section 5.1, having as a main goal to meet the Generic Requirements described on Chapter 3.

The following subsections will describe system features, describing the role of each system main component: server, mobile and web application, in each feature represented. Communications between system components are also described, detailing technologies, endpoints and message body contents. It's important to highlight that the key used as a validation object in the scope of communication is the user email, being the unique identifier of each user, grating permission to actions to be executed.

Another important factor regarding the Implementation of the GCS on the context of the development languages is the React Components to be chosen. This is important to be highlighted in the introduction of this section because of its widespread use in the features of the GCS.

React Components

React Components represent the part of the user interface and are reusable. There are two mainly React Components: the functional, also known as Stateless Components, and the Class Component also known as Stateful.

The functional component is a simple plain JavaScript function which takes props as an argument and returns a HTML UI. It has no state nor lifecycle so lifecycle-hooks are unavailable but is more perceptible to read and test than the Class Component. On the other hand React Hooks can be used inside a functional component: useEffect() hook can be used to replicate lifecycle behaviour, and useState can be used to store state.

Thus, having the need of the intervention of the programmer for maintenance or extension, a simpler code, being more perceptible will be preferable.

Having efficient solutions for their downside, functional components will be used across the implemented features for their simplicity towards the Class Components. This section will only cover the implemented features, endpoints and system flows of the Generic Crowdsourcing System. The extended applications of this generic tool, despite not being detailed in this section, will inherit all of the implemented components to be described bellow.

5.2.1 Authentication

One architectural component of the system is the User Authentication. Regarding this process, user interacts with the mobile app or the web application. This interface connects with the Server, accessing a database for user's storage and a general-purpose distributed memory-caching system for user's login session, limited by a predefined time out.

To fulfill the requirements of a Generic Crowdsourcing System, the Authentication component must be available on both mobile and web applications. On Figure 5.4 is represented the implementation scheme of this component, using the technologies referred on Section 5.1.

This Generic Crowdsourcing System allows normal or third party authentication, OAuth. Firstly, the highlight goes to the normal authentication, not accessing an external component for that matter.

For registration, system entities must use the client interface, web or mobile. There are 3 Text Inputs that are presented to the user to be filled: the name of the user, the email and the password, with caret hidden. These input fields are then aggregated in a JSON structure to be sent to the Server in a POST request, using in the header 'Content-Type:application/json' for describing body content. The Server register endpoint will receive the HTTP request, inserting the new user in the MongoDB if its email is not yet registered. Either way, Server will inform the client interface with a response to its HTTP request.

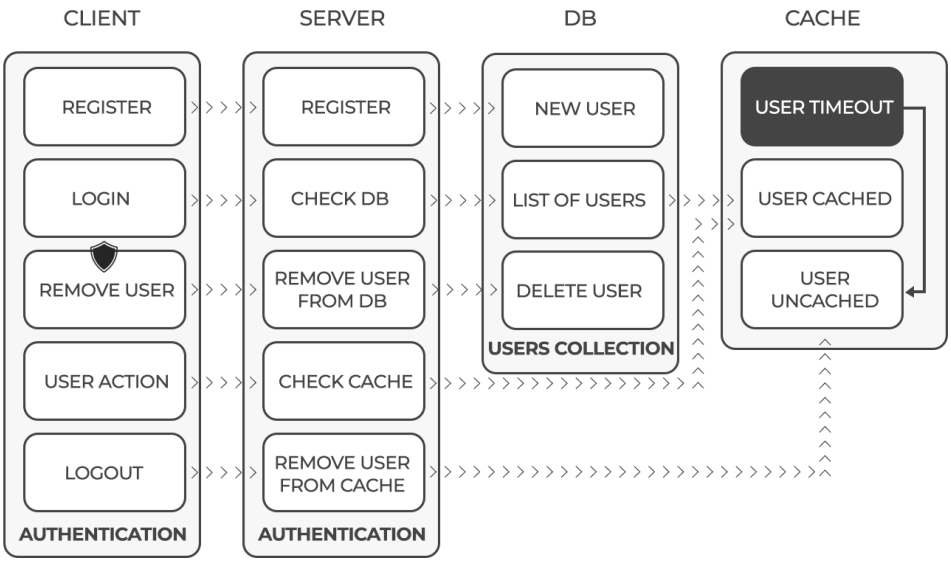


Figure 5.2: Authentication Module - Implementation Scheme

For login, system entities must use the client interface, web or mobile. There are 2 Text Inputs that are presented to the user to be filled: the email and the password, with caret hidden. These input fields are then aggregated in a JSON format to be sent to the Server in a POST request. The Server login endpoint will receive the HTTP request, checking and comparing the received credentials with the Users collection extracted from the database. If a match is verified, the user email will be used as a key in the Memcached with a predefined timeout. Either way, Server will inform the client with a response to its HTTP request, informing about the result of the validation. After the user timeout has been reached, the user is removed from the caching system.

Every HTTP request client-server will make a caching validation, checking if the user login session state is still active. Upon Logout or Login Timeout reach, user is removed from the caching service, blocking any further actions.

All this mechanisms regarding database and caching service are also applied when concerning the third party authentication, OAuth. This alternative gives the user the exact same application features, presenting the same usage as the normal method but simplifying the authentication step.

OAuth 2.0

Recently, integrating social functionalities into applications has become very common, easing the process of registration and, consequently, improving the user experience. Giving third parties outright access to user's personal information would be highly unsafe and undesirable, so protocols like OAuth, OAuth2 and OpenID are used by providers. OAuth 2, delegating more security to the HTTPS protocol than the other protocols, is an authorization framework that enables applications to obtain limited access to user accounts on an web service. By delegating user authentication to the service that hosts the user account, OAuth 2 provides authorization flows for web applications and mobile devices.

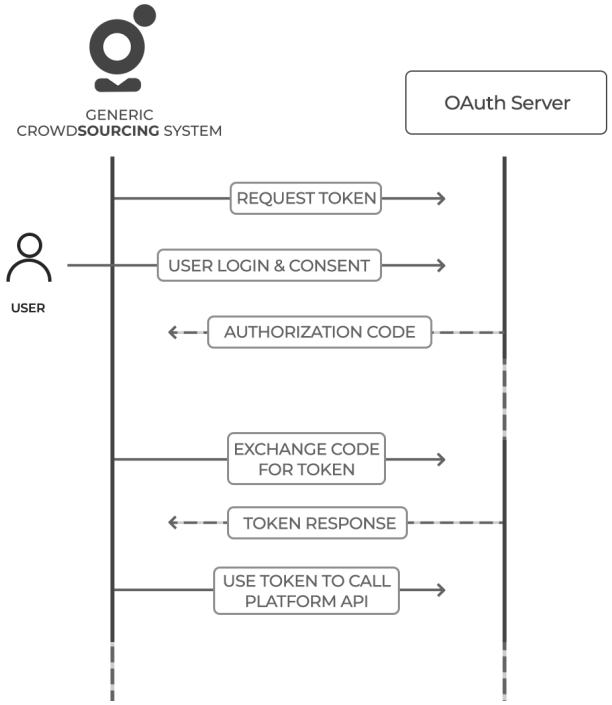


Figure 5.3: OAuth Authentication Scheme.

The OAuth process starts by requesting an authorization code to access service resources. If this request is authorized, the application receives an authorization grant code. Afterwards, the application requests an access token from the authorization server API by presenting the user identity and the authorization grant in order to be exchanged. If the application identity is authenticated and the authorization grant code is valid, the authorization server sends an access token to the application as a response. From this moment, having an access token, the application can request the pretended resource from the resource server by presenting the given access token to be authenticated. If the access token is valid, the resource server will serve the resource requested by the application.

Just like the normal authentication method, OAuth also registers new users on the database if not already registered. The credentials received from the resources server are name and email that is also stored along with a platform key, indicating the third party used. It also caches the login state of the users, removing upon logout or timeout, using this caching system to validate each client-server request.

Remove User & Change Type

Only for the web application, this feature allows the administrator, upon login, to remove users or change their types.

Firstly, a list of users is loaded to the client interface. This is made by a POST request done with Axios, sending the administrator email for caching validation and receiving a JSON with the list of users as a response from Server.

This list of users is then rendered to the client interface by selecting the fields with interest: name, email and user type. The last field can only be user or researcher. Each user is then presented to the administrator as well as two icons to each user: a delete button and a toggle button that are able to perform two tasks, remove a user or change its user type, respectively.

The first sends a POST request with the administrator email for caching validation and the email of the user to be deleted. This request will be sent to a narrowed endpoint, making the Server to delete the user by requesting it to the database, using the user email as key for identification.

The second task sends a POST request with the administrator email for caching validation, the new user type and its email. This request will be sent to a narrowed endpoint, making the Server to change the user type by requesting it to the database, using the user email as key for identification.

5.2.2 Language

The Generic Crowdsourcing System has a Language picker that allows changing the mobile application language. This is made by declaring the pretended languages to be available. After this activation, every screen that will be rendered must fetch the language state of the mobile application. Then, having a dictionary imported, each screen will replace the UI according to the language selected. The language picker UI is displayed in the mobile application landing screen with TouchableOpacity, a react native wrapper for making views respond properly to touches. The images associated with the languages are flags that are loaded from images in the assets folder of the native application.

5.2.3 Profile

The profile component is a generic component that belongs to the GCS. Its implementation in the React Native mobile application intends to strengthen the relation of the crowdsourcing system with the user, introducing a certain level of customization and gamification.

It consists on a user identification card providing an uploading photo system, allowing the user to upload a photo from its camera roll or smartphone gallery. This photo will be stored in a React Native state with Base64, a group of binary-to-text encoding schemes that represent binary data that include the ability to embed image files.

Profile component also displays the user name, providing user the ability to update it. A text input block pre-built component is used to then save the new information on a React Native state as well.

The profile component imports a Ranking Handler, implementing the gamification path chosen to the GCS. This raking component displays the user current ranking points and a Ranking Level ['Amateur

Explorer', 'Good Explorer', 'Pro Explorer', 'Explorer Premium'] that is activated when a certain amout of ranking points is reached. The Ranking Level is displayed along with a respective image, present in the assets folder of React Native.

To submit the new user name or photo, there is an Update button that sends this information to the server.The credentials are then updated/inserted on the database, using the user email as the identifier. Since the image is stored in the Base64 format, the size of information is to big to be sent in the HTTP POST request with the 'Content-Type:application/json', JSON with Ref 1.2 format. The header of this type of requests usually is done using 'Content-Type: multipart/form-data', normally used to send images since with this type of content, data is sent in chunks. Thus the profile updated values must be stored and sent in a FormData, an object that allows compile a set of key/value pairs and even files.

Multer

The POST request will be received in the server using Multer [77], a middleware for Express and Node.js that makes it easy to handle multipart/form-data when users upload files.

Multer middleware also allows size modifications, alocating the space required for the FormData to be received in the server side. Thus, this middleware will be used to fulfill the Profile Requirements, being able to send simple objects or files with a bigger size, like the user profile image.

5.2.4 Surveys

Surveys are a key component of the Generic Crowdsourcing System. The main objective is to implement a module fully customizable, providing a simple UI to the client, ensuring extensibility to the GCS. This surveying process, declared endpoints and the necessary mobile-server data flow was developed by Miguel Leitão, developing its Masters in Electrical and Computer Engineering thesis regarding Urban Green Spaces.

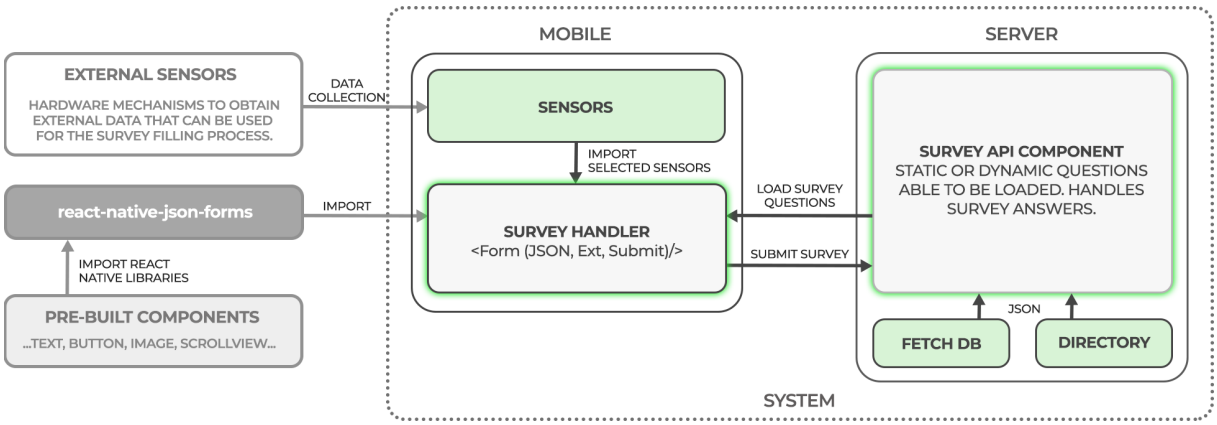


Figure 5.4: Surveys Module - Implementation Scheme

The pre-built components available in the React Native library provide a simple but powerful tool that can be used to create diverse surveys, allowing to extract diverse type of data with different survey

component tools. The researcher is responsible to choose the components that considers to best fit the scientific surveying needs.

There are two types of surveys that can be presented to the user: static or dynamic. The mobile application makes a HTTP POST Request to the Server in the Surveys component. The Server, having the surveys available on a local directory or stored in the database, will fetch the requested JSON files. The client POST request is connected to a server endpoint that retrieves JSON files as a response with a dynamic, static or both, according to what the programmer has established.

Then the JSON file received in the mobile application must be handled. Therefore, a library package, detailed on sub section 5.2.4, was developed for this thesis with the collaboration of Miguel Leitão. The necessity of a package was raised for the amount of files needed to cover all the diverse components that are considered fit to a survey.

The package is imported to the Surveys screen on the mobile application, allowing to upload as 'props' extension components to it, incorporating them on the UI render, and a submit function. This uploaded function will allow the imported package to send the survey output data, in a JSON format, to a server endpoint. The POST request will contain the user email and the JSON survey answer in its body, demanding the Server to store the received data along with a timestamp in the database, in a raw 'answers' collection.

After receiving this JSON file, the mobile application must handle the received data, rendering to the user the survey to be presented.

react-native-json-forms

The advantages of JSON turns this file format into a powerful and simple tool to be used in the Generic Crowdsourcing System, seamlessly communicating between JavaScript, Node.js, React, React Native and MongoDB environments and, lastly, for extension and customization. Based on the principles of Citizen Science, regardless of the users scientific knowledge, a well explained, simple UI structure form to be presented to the user, is the most efficient and objective way to obtain accurate data.

The form intended for a Generic Crowdsourcing purpose must be, regardless of its simplicity, able to tackle a large variation of subjects, diverse survey components, able to be customized, gathering all the user inputs into an object to be used for the processing phase.

Thus, using JSON as an input file of a Crowdsourcing Platform, a form with a simple UI structure could be generated based on the components activated by the JSON. There are some websites that provide form generation, online or with code snippets to be incorporated in larger programming modules, for example, SurveyJS [78], SurveyMonkey [79] or Typeform [80]. Despite all of their diversity in terms of functionalities and services, most of this applications are expensive and very limited in terms of customization. Expecting a specific format of a JSON file, the previous mentioned websites and similar ones, don't offer a lot of extensibility, not allowing alternative options of form components.

From the list of websites analyzed previously, the most suitable to the pretended objective was the SurveyJS [78]. This particular website had a wide range of form elements along with a simple UI online platform that helps users to build a form, allowing to see the parallel construction of the JSON file.

Thus, in order to solve the mentioned need, a tool was developed to create forms using an uploaded JSON file, containing all the pretended form structure. This tool was developed to be compatible with forms created using, for example, SurveyJS [78], having some extra developed features. The extension problem mentioned previously is covered with this tool by allowing the developer to upload a JavaScript custom extension component to the project, activating it through the JSON File. This way the Form element will have the following proprieties:

- **json**: Passes a JSON file containing the description of the form structure and details.
- **extension**: Passes a JavaScript file, if available, containing the description of the extension elements that the user wants to implement.
- **onSubmit**: Passes a function that receives an JavaScript object as argument containing the answer to the form.
- **showSubmitButton (Optional)**: Boolean that when false hides the submit button, useful when extension elements do not require/cannot have submit button. If any of the core elements is used the button automatically appear even when showSubmitButton is set to false.
- **submitText (Optional)**: String that allows to customize the submit button text.

In order to associate an answer with its respective question an optional **id** field can be added to each element in the survey JSON. By using the ID feature in the JSON file, the answer to a specific question will also contain an id field with the same value as the question's id. This library feature proves to be very useful when it comes to dynamic surveying.

Another field that can be added is the **required** field to any of the form's elements its submission is blocked until those element's questions are answered. It is a boolean field that its default to false. Elements such as expression, image, HTML or any custom elements where the final answer may be an empty string may not be compatible with this feature.

The developed open source tool has its source code in a GitHub Repository [81] but also, to simplify the package install, it is available too in npm [82], the package manager for JavaScript. Tool details can be found in both links along with the Contributors information, License and Copyright provided by MIT License and some Coming Soon Features.

5.2.5 Data Validation

Only for the web application, this feature allows the researcher to increase data reliability to the system.

Firstly, a list of answers is loaded to the client interface. This is made by a POST request done with Axios, sending the researcher email for caching validation and receiving a JSON with the list of survey raw inputs a response from Server. This list is then rendered to the client interface by mapping the JSON file received, selecting the fields with interest to the researcher, implemented previously by the

programmer. Each answer has an ID associated that is unique and is useful for the features provided by the web application.

Each user answer is then presented to the researcher as well as a delete icon per item, able to delete it from the system. This is done based on the scientific knowledge of the researcher, that upon error detection, is capable of deleting the respective item that is proven to be faulty or inaccurate.

This removal feature is done by sending a POST request with the researcher email for caching validation and the answer ID of the answer to be deleted. This request will be sent to a narrowed endpoint, making the Server to delete the user input from the 'answers' collection on database, using the answer ID as key for identification.

5.2.6 Data Download

Another generic crowdsourcing system feature is the download of the survey raw user inputs. This is done by using a library 'react-download-link' that provides a simple mechanic easy to understand by the programmer. This system entity is responsible to define what fields from the JSON file, received as a response from the server, are going to be used in the file to be downloaded.

This library is imported in a Generic component, 'Download', that receives the necessary information to define the structure of the file to be downloaded. If the information is not provided, the whole JSON file with the raw user inputs is provided for download.

5.2.7 Help Presentation Guide

There is also a Help Presentation Guide, a responsive React Native Horizontal Carousel. This generic component consists on screens displayed horizontally, allowing the user to scroll between them, presenting ordered information pages to the user. It is located on the Menu Screen on the The main objective of this component is to explain to the user with a simple UI the basics of the crowdsourcing application.

5.2.8 Feedback

Citizen Science with Immediate Feedback, a thesis proposed by the Centre for Ecology, Evolution and Environmental Changes (cE3c) - FCUL, supported by iLTER, requires a Feedback mechanism capable of returning Immediate Feedback to the user.

This was used as a drive force to develop a generic Feedback component that represents an important role of the Generic Crowdsourcing System. This component, being inherited to any extended application, will be formatted to fit in the scientific framework that is at issue. This extension procedure is made by the programmer, leaving up to researchers administration to understand the scientific needs of the extended tool. The implementation of the Feedback Module is represented on the Figure 5.5, detailing the Server and the Mobile system components, relevant for the module to be issued in this subsection.

Upon Survey submission, user data is fetched and stored by the Server and the Feedback endpoint is called through a HTTP POST request. This API call sends the user email for caching validation and any content that is found appropriate to be used on the Feedback module. The Feedback server endpoint is connected to the feedback module that has two segments: the immediate and the differentiated feedback.

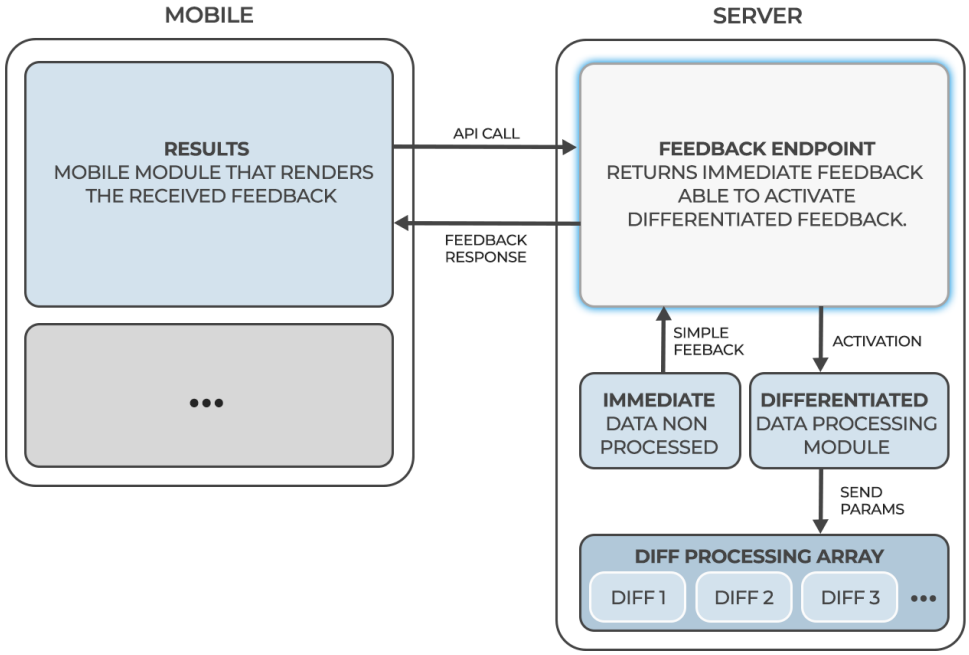


Figure 5.5: Implementation of the Feedback Module

The first is a module that returns a simple feedback to the user, forming the body content of the response of the POST request made to the Feedback server endpoint. This immediate feedback can be a simple string or a simple object that does not require data processing. This module sends the output back to the mobile application as a request response. The client interface, waiting for the server response, will receive it and use it for display or processing, according to what was programmed.

The other segment, the Differentiated Module depends of the user will of activation, being optional and customizable. It is important to highlight that this module is executed somehow asynchronously, not demanding user to wait for the processing to finish. As a endpoint, a message body containing information to be processed can be sent from the mobile application. This is sent as a parameter to a mapping function that runs each differentiated component sequentially. Each block, *Diff1*, *Diff2*, *Diff3*... represented on Figure 5.5, is composed by a set of functions and components defined by the programmer. Each of this differentiated blocks process data and saves it to be handled by the next block, and so on. This allows data to be saved and handled progressively, allowing a better error handling or processing modifications. Afterwards, the mobile application, with the Results Section may request the output of the Differentiated Feedback Module. Data will be sent to the client interface when ready, being adapted and displayed in the respective section of the mobile application. Despite the name chosen to this module, Differentiated Feedback, it is also a form of instantaneous feedback since the user only have to wait until the processing is complete.

5.3 REST API

REST, Representational State Transfer, is one of the most used web service technology. Sharing data between two or more systems has always been a fundamental requirement of software development. Thus, a REST API is a way for two system components to communicate over HTTP in a similar way to web browsers and servers. The following subsections represent the main routes of the server, detailing for each route a specific endpoint that performs a specific system feature.

5.3.1 Users

Contains all the endpoints that handle requests related to the users. Handles users authentication, registration and login. It allows to get and update the generic fields of the user Profile. Administrator features are also represented on this subsection like changing the user type or removing users. All the endpoints represented are preceded with '*/api/users/*'.

```
1 router.post('/', async (req, res)); // Get User List
2 router.post('/register', async (req, res)); // Register User
3 router.post('/login', async (req, res)); // Login User
4 router.post('/logout', async (req, res)); // Logout User
5 router.post('/edit', upload.single(), async (req, res)); // Edit User Profile
6 router.post('/get', upload.single(), async (req, res)); // Obtain User Profile info
7 router.post('/changeType', async (req, res)); // Change User Type (Administrator Only)
8 router.post('/remove', async (req, res)); // Delete User (Administrator Only)
```

5.3.2 Profile

Contains all the endpoints that handle requests related to the user's profile in the mobile app. It allows to get and update the extended fields of the user Profile. It also allows the ranking points update. All the endpoints represented are preceded with '*/api/profile/*'.

```
1 router.post('/', async (req, res)); // Get user profile
2 router.post('/editRequest', async (req, res)); // Request to edit user's profile
3 router.post('/edit', async (req, res)); // Edit user's profile (submit changes)
4 router.post('/editRanking', async (req, res)); // Update user's ranking points
```

5.3.3 Results

Contains all the endpoints that handle requests related to the results screen in the mobile application. All the endpoints represented are preceded with '*/api/results/*'.

```
1 router.post('/getData', async (req, res)); // Get results
```

5.3.4 OAuth

Contains all the endpoints that handle requests related to OAuth authentication. All the endpoints represented are preceded with '*/api/oauth/*'.

```
1 router.post('/login', async (req, res)); // Login user with OAuth
2 router.post('/register', async (req, res)); // Register user with OAuth
```

5.3.5 Surveys

Contains all the endpoints that handle requests related to the surveys. Allows to get surveys, answer them, get feedback, submit new surveys and get additional info from the system. All the endpoints represented are preceded with '*/api/surveys/*'.

```
1 router.post('/', async (req, res)); // Get a survey
2 router.post('/submit', async (req, res)); // Submit a new survey to the server
3 router.post('/answer', async (req, res)); // Submit survey answer
4 router.post('/answerImage', upload.single(), async (req, res)); // Submit survey image
5 router.post('/feedback', async (req, res)); // Get feedback of an answer
6 router.post('/getInfo', async (req, res)); // Get info to help answer survey
```

5.3.6 Researcher

Contains all the endpoints that handle requests related to researcher's console. Allow to get, edit, remove or validate system data. All the endpoints represented are preceded with '*/api/researcher/*'.

```
1 // Get some data for the researcher's console
2 router.post('/getData', async (req, res));
3 // Edit some data from the researcher's console
4 router.post('/editData', async (req, res));
5 // Remove some system data from the researcher's console
6 router.post('/removeData', async (req, res));
7 // Validate som data from the researcher's console
8 router.post('/validateData', async (req, res));
```

5.4 Usage

The Generic Crowdsourcing System is available on the GitHub Page [83] 'generic-crowdsourcing-sys' with the contributors: Guilherme Eugénio (guilherme.eugenio@tecnico.ulisboa.pt) and Miguel Leitão (miguel.s.leitao@tecnico.ulisboa.pt), INESC-ID, under the MIT Licence. This package has 3 main folders: server, mobile and client (web). All the information to run each system component is described on the Wiki section of the GitHub Page [83], requiring the installation of development environments and dependencies. For the installation and the initiation of each system component npm was used, a package manager for JavaScript environment.

It's important to highlight that despite this section is related to the Usage of the Generic Crowdsourcing System, extension folders will be mentioned throughout this section because there are some default connections and imports that are already declared to leave this generic tool ready to be extended.

Regarding the folders present on the GCS, the extension of each system must be done in the respective *extension* folder. This folder is on the main directory of each system component with an exception to the *client* that has it inside the *src* directory. In each *extension* folder there is a *config.js* file that is important to set up credentials, URL's, timeouts and other important information that could be useful throughout the extended application.

5.4.1 Server

The Server has a set of endpoints that is provided in the generic part of the server covering many features essential to every crowdsourcing system.

This endpoints are detailed on the Routes directory each one dedicated to one part of the system and its functionalities. This endpoints import files from the extension folder, dedicated to hold the files containing the specific system features, in other words, the features implemented by each programmer so the application can solve a specific problem. The changes to the program must be done in this files, maintaining the Routes directory unchanged as well as the structure of the modules folder, maintaining the default structure of the Generic Crowdsourcing System.

The extension directory has 2 folders inside: feedback and public. The first folder contain the files to be used for data processing, the immediate and differentiated feedback to be executed sequentially. The second folder, public, contains a set of images that can accessed and fetched using an URL containing the file name. This extension directory also This folder is composed of a series of files that are yet to be developed by each programmer in order to implement the desired features. These files have functions and components that may and should suffer changes in order to integrate the specific features that the programmer needs to implement.

The modules directory is composed by files that represent to data storage, like database or caching service, and feedback containing the functions to perform immediate and differentiated feedback. There is also the main server files, including metadata in the main directory of the server.

5.4.2 Mobile

This section is dedicated to the mobile application of the system. After the installation of the Expo development interface and all packages are installed it's time to dive into the structure of the mobile GCS directory structure.

The mobile app follows the schema of the rest of the system, it has a generic part that implement the basic functionalities and makes the integration with the remaining parts of the system but it also has an extension folder. This folder is composed of a series of components and functions that are yet to be developed by each programmer in order to implement the desired features. The extension folder must also include the components and functions provided in the template in order to connect with the generic part. These function and components may and should suffer changes in order to integrate the specific features that the programmer needs to implement. The extension directory has 2 folders inside: presentation and sensors. The first contain ordered files, each containing one screen to be presented in the Help Guide, having to be imported to the config file. The second contains the sensors that could be loaded with the JSON form to be used for the SurveyScreen.

There are 6 screens that compose the mobile application. The landing screen, MainScreen presents the login form and a navigation to the RegisterScreen in the case user is not registered. After login, MenuScreen is presented, allowing navigation to ProfileScreen, ResultsScreen and SurveyScreen.

The navigation folder has a stack navigator, MainNavigator, which works stacking screens on top of each other as the user visits them. It loads the screens developed for this mobile interface, setting up a navigation environment that connects each one and enables user to navigate between them.

The components folder are modules that represent objects to be used across the Generic Crowdsourcing System like, for example, a generic designed button that has multiple params. This folder was designed to contain reusable functions, objects or components across the mobile interface.

The data folder contains JSON files that can be useful to imports done in run time or just to keep a copy of a JSON file that is stored on the database.

The constants folder contain the Stylesheet files for React Native, an abstraction similar to CSS StyleSheets, that can be imported across the mobile interface.

The config folder contains the activationConfig file that imports files from the extension folder: activationHandler, that receives the activation mode from the activationJSON file and activates the notifications accordingly. This service runs a function registered with Expo Task Manager every time the user location matches one of the areas selected. These files must not be delete from the extension even if the programmer does not want to implement these feature, in this case the json file must contain the null mode and the activationHandler must contain the activationHandler function but without any code inside of it.

Lastly there's the assets folder, containing the images that can be used across the mobile application. There is also core files in the main mobile directory that sets up the application to be executed, having metadata available in this directory as well.

5.4.3 Client

The client folder, representing the web application, was built using the React framework, making it available for any browser. The web interface follows the schema of the rest of the system, it has a generic part that implement the basic functionalities and makes the integration with the remaining parts of the system and it also has a specific part. All this source files are inside the src directory that also includes the extension folder.

The extension folder, containing components and functions provided in the template in order to connect with the generic part, has files that must be programmed, being already imported to the files in the pages directory. These function and components may and should suffer changes in order to integrate the specific features that the programmer needs to implement.

The pages folder contain the generic screens that can be displayed to the client interface. The LandingPage that enables the user to navigate to the LoginPage or the RegisterPage. After login, there is the AdminPage or the ResearcherPage according to the user type. If the user pretends to logout it has a LogoutPage that performs that action, redirecting to the LandingPage again. Since the caching service is active as well for the web application there is, lastly, a ExpiredSessionPage that redirects to the LoginPage.

On components folder there is a Header file inside the layout directory. This file allows the web application to provide a navigation bar for each screen using 'react-router-dom'. This allows to build a single-page web application with navigation without the page refreshing as the user navigates. Inside the components folder is also modules that represent objects to be used across the Generic Crowdsourcing System like, for example, a generic designed button that has multiple params. This folder was designed to contain reusable functions, objects or components across the web interface.

Chapter 6

Demonstration

A powerful proof of concept of the Generic Crowdsourcing extensibility is the parallel development of the eFlechten, an extended application in the context of this thesis and described on Section 6.1, developed and the Urban Green Spaces, described on Section 6.2. Both applications are an extension of the Generic Crowdsourcing System that was developed with the collaboration of Miguel Leitão, an Electrical and Computer Engineering master's student within the scope of his dissertation who also developed the mentioned Urban Green Spaces System.

The existence of this two system applications is relevant to highlight the multiple genericity of the system: the possibility to extend the generic system into a variety of applications. By extending modules and creating different versions of the same core system it is possible to develop divergent successful applications with little programming effort.

6.1 eFlechten

Driven by the thesis proposal from the Centre for Ecology, Evolution and Environmental Changes (cE3c) - FCUL, supported by iLTER, a Generic Crowdsourcing System was developed able to tackle different scientific scopes including the narrowed one proposed specifically for this work. So, to fulfil the scientific requirements for this thesis, the Generic Crowdsourcing System was extended into a narrowed application to tackle the raised issue.

'Flechten', in German, means lichens, a powerful ecological indicator that can be used to tackle the existing scarceness of mapping data in the scope of climate change and air pollution. Able to survey lichen diversity in a citizen science project, eFlechten allows to analyze the effects of the global change drivers, using an app-based interface. By returning immediate feedback to the user, processed surveying data is immediately provided. The following subsections describe the GCS extended adjustments made to the available generic features, turning eFlechten into a successful proof of concept of this system extensibility.

6.1.1 Authentication Extension

eFlechten uses the Generic Crowdsourcing System Authentication, having just a personalized logo and a different welcoming text. This text is loaded from the dictionaryExtension file, stored in the extension directory and varies according to the set up language. On Figure 6.1 is represented the Login and the Register Screen. After the user is logged, the Menu Screen, on the right, is presented, containing all the crowdsourcing features represented through buttons, inherited from the GCS.

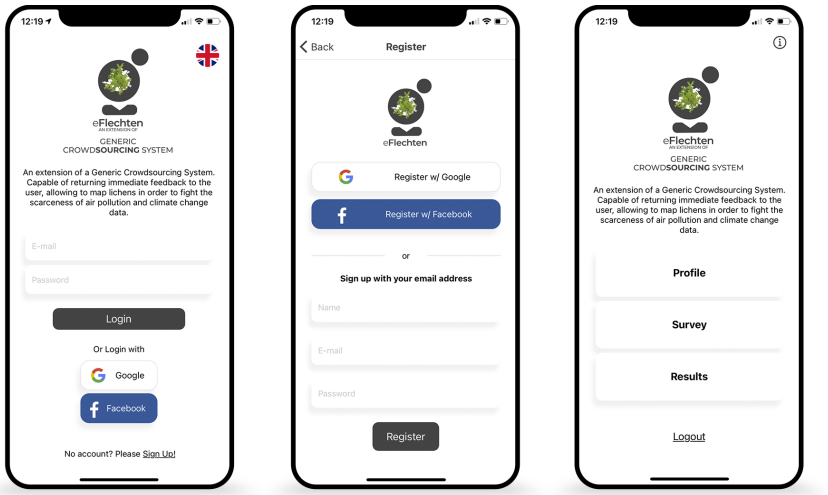


Figure 6.1: eFlechten Authentication and Menu Screens.

6.1.2 Profile and Help Guide Extension

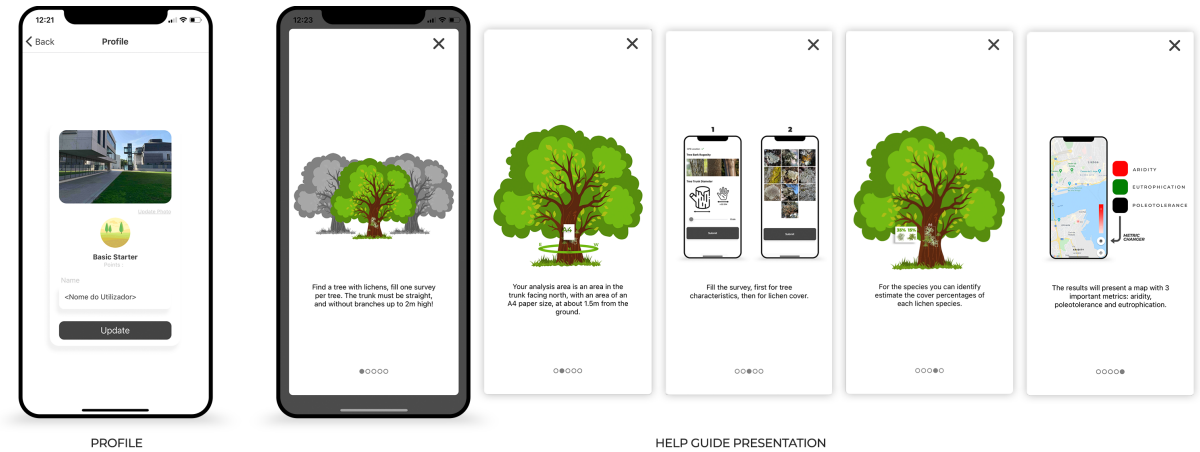


Figure 6.2: eFlechten Profile and Help Guide Screens.

Since there is no special user information that is relevant besides the one declared on the Generic Crowdsourcing System, the Profile Extension is left in blank. So, the Profile Section on eFlechten implements the generic profile upload, user name change box and a Ranking System, with ranking points and a ranking level associated.

There is also a Help Presentation Guide for eFlechten, a component that consists on five screens displayed horizontally, allowing the user to scroll between them, presenting ordered information pages to the user. These files are defined on the presentation folder, inside the extension directory. The main objective of this component is to better understand the Lichen Sampling Method, applied to this Citizen Science Scope, described on Subsection 2.3.6. It also provides information about the Survey filling process and the Results dynamic, providing UI support.

6.1.3 Activation Extension

The lichen surveying process don't depend on external factors nor dependencies. Since the lichen sampling method could be done anywhere, urban or rural areas are considered. This means that no matter the location, time, date or other external triggers are not relevant for this work. Thus, the Activation Extension was not considered necessary, not requiring notifications system on the mobile interface.

6.1.4 Surveys Extension

In the context of the problem proposed by cE3c - Centre for Ecology, Evolution and Environmental Changes, concerning the information scarceness in the framework of air pollution and climate change maps. The Survey module will be extended in order to present a efficient surveying tool that tackles the problem presented in this work with the lichens mapping method.

This surveying GCS feature will be used to fulfil the mapping lichens sampling method, described on the Section 2.3.6. The input file must be structured in a way that responds efficiently to the need of surveying lichens species and percentages of occupation while providing geolocated information about the sampling tree trunk.

The Surveys Extension represents one of the main extensibility features of the Generic Crowdsourcing System. It allows static surveys or, if questions are influenced by user inputs, dynamic surveys. For this project scope there's no advantage of having dynamic surveys since the Lichen Sampling Method to be used is an observe and register process.

Static Survey

Loaded to the SurveyScreenExtension file in the mobile extensions directory, the JSON fetched from the database "surveys" collection is imported to the mobile as a response to the POST request made. This JSON file, containing a static survey will be handled in a useState() hook, being rendered with the 'react-native-json-forms' package, described on Subsection 5.2.4. The JSON file contains an array with two components, being each one a survey. Each survey calls the pretended components to best fit the surveying needs.

```
1 [{"pages": [ {"name": "page1",
2     "elements": [
3         {
4             "type": "ext:sensor",
```

```
            "subtype": "geolocation",
            "name": "Geolocation Sensor",
            "required": true
        },
        {
            "type": "imagepicker",
            "name": "Tree Bark Rugosity",
            "required": true,
            "numberPerLine": 3,
            "imageSize": "big",
            "singleChoice": true,
            "choices": [
             {  "value": "Rugose",
                "imageLink": "http://146.193.41.162/lif/server/public/rugoso.jpg"},
             {  "value": "Intermediate",
                "imageLink": "http://146.193.41.162/lif/server/public/intermedio.jpg"},
             {  "value": "Smooth",
                "imageLink": "http://146.193.41.162/lif/server/public/liso.jpg"}
            ]
        },
        {
            "type": "TrunkSlider",
            "name": "Tree Trunk Diameter",
            "required": true,
            "min" : 0,
            "max" : 200,
            "link": "http://146.193.41.162/lif/server/public/trunk.jpg"
        }
]}]}]
```

The first survey to be rendered corresponds to the tree trunk identification having three elements: **Geolocation Sensor**, **Tree Bark Rugosity** and the **Tree Trunk Diameter**. The first element, stored natively in the sensors directory, is sent to the 'react-native-json-forms' library with a generic type 'ext:sensor' so that it can be handled appropriately. It provides the location of the user, demanded by the programmer to have the highest accuracy, creating geolocated data. The second element, intends to describe the Tree Bark Rugosity, being used the Image Picker element from the 'react-native-json-forms', displaying 3 rugosity levels: Rugose, Intermediate, Smooth. Each level have a server stored image associated, displaying a single choice image picker. The third element, Tree Trunk Diameter, is a component created specifically for this project. It is stored in the extension directory, the TrunkSlider, and is loaded to the FormExtension file just like the Geolocation Sensor. This file is then passed as the extension props to the 'react-native-json-forms'. This third element displays a slider from 0 cm to 200 cm with steps of 10 cm. It also displays an natively stored explaining image to provide support to the user. On the previously represented code section is the representation of the first survey JSON file, being handled by the 'react-native-json-forms' library.

The Second Survey only has one element: the **LichensImagePicker**. The main objective of this element is to, in coherence to the Lichen Sampling Method described on Subsection 2.3.6, obtain the lichens present and their respective percentage on a study site. Since the Image Picker element from the package is quite simple and mattering the fact that lichens percentage is needed, a extended image picker must be created, the LichensImagePicker. Just like TrunkSlider, this is another natively stored element to be used as the extension props to the 'react-native-json-forms'. It displays natively stored images of lichens, associated with keys, allowing a multiple choice mechanism. The user upon lichen selection, is faced with a pop up box, containing the lichen in question zoomed up and a 5% step slider, with a 0 to 100% coverage percentage. The inserted percentages are displayed to the user after the abundanceAnalyzer function. Since it is not possible to have more than 100% lichen coverage of the study area, this function readjusts the percentages if the sum surpasses the maximum. Thus, the last percentage inserted maintains equal and the others are adjusted to a minimum of 5%, decreasing the last input only if needed.

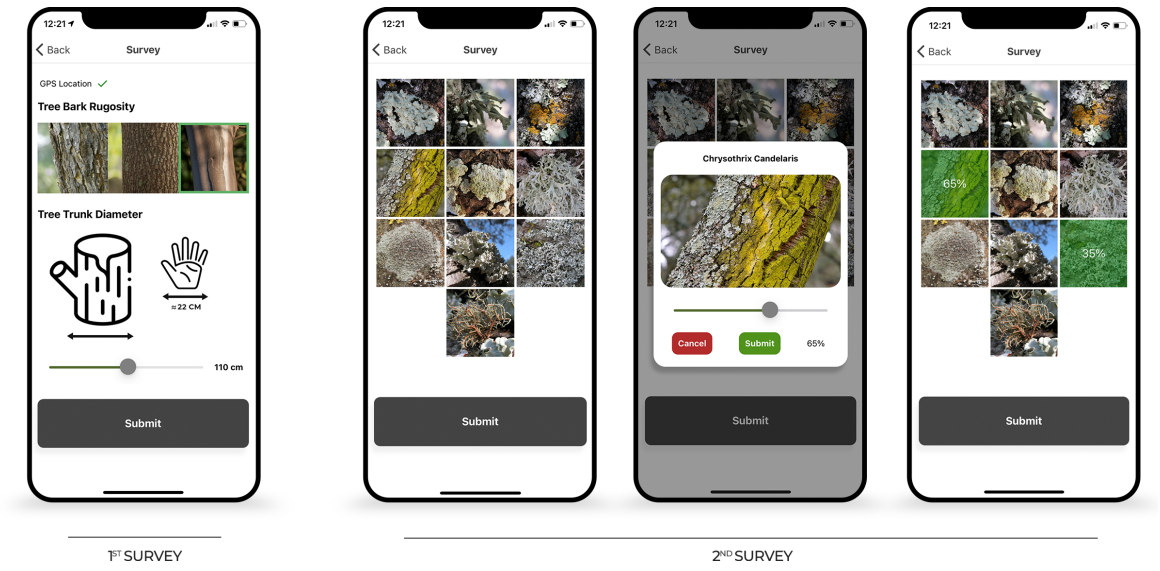


Figure 6.3: eFlechten Survey Screens.

In the Citizen Science scope, due to the lack of knowledge of lichen species identification and to avoid human error, all the survey components were considered. An Image Picker mechanism was considered simple enough to tackle this apprehensiveness, providing a simple UI that displays images that are easily recognizable, easing the process of lichen species identification in the field.

All the chosen components have required field which makes the 'react-native-json-forms' to wait until all fields are filled up. A submit action is required to go from the first survey to the second survey, displaying a final submit button that sends all data is aggregated in a JSON format to be stored in the Server in a 'answers' collection.

6.1.5 Feedback Extension

After data submission, the server feedback endpoint will be called, sending the recently added data. This endpoint will handle data and give feedback to the system. For this work, the Immediate Feedback is crucial to fulfil the requirements. Thus, this instantaneous feedback needs to activate two features of the Generic Crowdsourcing System: the Immediate and the Differentiated Feedback. This two features are programmed in the Server in order to obtain the desired system outputs. These two feedback components are completed someway asynchronously, which means that the Immediate Feedback don't needs to 'wait' until the Differentiated Feedback completes.

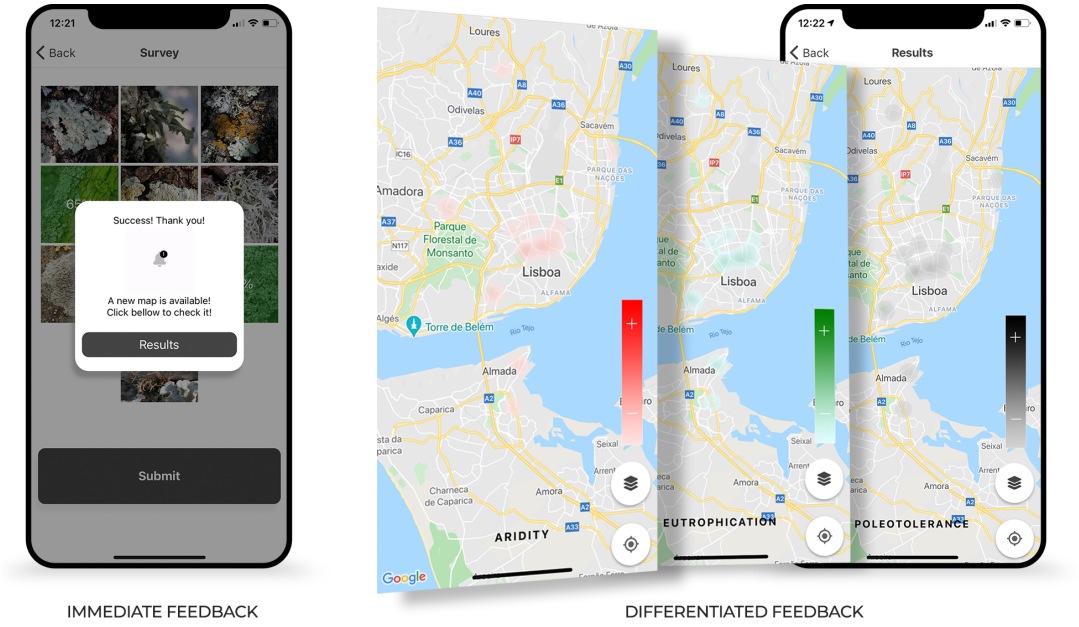


Figure 6.4: eFlechten Feedback Screens.

Immediate Feedback

The Immediate Feedback corresponds to the module that sends a simple response back to the mobile interface, requiring no data processing. In the case of this work, the response is a simple message to be displayed on the mobile application.

Differentiated Feedback

The Differentiated Feedback is more complex than the Immediate Feedback and does the data processing necessary for the metrics calculus needed for this system functionality. The feedbackArray file imports the diffFeedback files needed to fulfill the different stages of the data processing. In this work, it was declared 2 files to be loaded.

diffFeedback1, the first differentiated feedback module loads all the 'raw answers' from the database, even its own, and compares to the recently added one. If the distance between them is lesser than the one declared on the configuration file, in this case 30 meters, than the user inputs are stored in an array and declared as being in range. This processing step is a simplified approach to obtain an

average, having considered a 30m x 30m block of influence that will taken in consideration on the Lichen Sampling Method, described on Subsection 2.3.6. Then, each one of the user inputs that are considered in range have to updated as well so, the same logic have to be applied to them too. In the end of this diffFeedback1, a JSON file is saved to the 'DiffFeedback1' collection on the database. This file contains the latitude and longitude of each point to be inserted or updated. It also contains the survey id of each point and the compilation of all the lichen species and respective occupation percentages that belong to the block of influence.

diffFeedback2, the final differentiated feedback module loads the Metric Indexes based on Lichen Species, represented on Table 2.1. It also loads the DiffFeedback1 JSON file associated with the recently submitted survey, generated on the previous step. By mapping each point to be updated, this module generates the final indexes with the equations that are described on the Metric Index Model, detailed on Subsection 2.3.6. Thus, all the points in consideration are updated based on their id, the survey identifier. Since the mobile interface will display the indexes with maps, this differentiated module will generate files in a JSON format containing the latitude, longitude and the respective indexes of the three metrics associated with that location. This file is then stored in the database, on a 'DiffFeedback2' collection. This weighted geolocated point is ready to be used on the Results Section of the mobile interface, described on the Section 6.1.6.

6.1.6 Results Extension

The desired specific requirements for the context of this dissertation suggested the usage of maps for their simplicity and interactivity. Furthermore, since the metric scales are relative, the system output must be represented by maps and not absolute values. The ResultsScreenExtension file, present on the extension directory, is modified to import the MapExtension, a file that is stored in a newly created folder called map. The MapExtension imports the 'react-native-maps' library [84] that allows to use Google Maps API. A map can be then rendered on the mobile application on the Results Section, showing the user location. In order to display the geolocated metric indexes the most efficient and perceptible way possible, Heatmaps were used. Besides the map mobile render, the imported package provides some components API, including Heatmaps. This component API needs weighted geolocated points, loaded from the 'DiffFeedback2' collection from the database. This file is sent as a response to the POST request to the Server endpoint 'results/getData'. The results of the Differentiated Feedback is then passed as props to the HeatMap file, a component created in the map directory.

Since Heatmaps need gradients to be rendered and each point has three metrics to be displayed there will be an overlap of gradients. So, in the name of perceptibility, the HeatMap file will render three different heatmaps to represent each metric. Each heatmap is associated with a specific gradient that is available on the metrics file on the same directory. The file passed as props containing the geolocated information with the three metric indexes will be assigned respectively. Lastly, a button was created to perform the toggling between heatmaps, giving to the user an immersive experience. On Figure 6.4 it is possible to observe the final feedback display on the mobile application.

6.1.7 GeoJSON Download

On the web application, Data Download feature, described on Subsection 5.2.6 is extended to this work, allowing to transfer surveys answers data. The format chosen is GeoJSON [85], a format for encoding a variety of geographic data structures that can be loaded for example to ArcGIS [86] or QGIS [87], geographical information systems that are used to collect, analyze, and represent spatial data in a way that can be understood by the end user.

The GeoJSON Download is activated with a button that is programmed in the ResearcherExtension file, on the extension directory. This GeoJSON modification was done to the same file, in the exportFile props of the Download component.

6.1.8 JSON Survey Submission

If the Researcher needs to change the Survey, no recompilation is needed. On the web application, JSON Survey Submission feature was programmed specifically for this work, providing a button that upon click, renders a text input box. This box receives a JSON format file that is uploaded to the Server by a POST request to the 'surveys/submit' endpoint. Then, on the back end side, this JSON file is stored in the database, on the 'surveys' collection. This is an important feature of the eFlechten, allowing to perform immediate modifications to the survey that will be displayed on the mobile interface. This allows the researcher team to have some freedom inside the application but, with the impact of this action this must be done with the programmer, ensuring the integrity of the system.

6.2 Urban Green Spaces

In the context of his MSc Thesis for Electrical and Computer Engineering, Miguel Leitão, driven by the lack of documented reliable information about Urban Green Spaces (UGS), extended and adapted the Generic Crowdsourcing System for this scientific context. This work proposes to extend the GCS tool in order to enable the development of an application able to extensively gather detailed information about Urban Green Spaces by dynamically surveying its visitors. The projected is hosted in a GitHub Page [88] and is available for usage.

In this scientific context, the activationJSON file was changed to contain all the areas of interest, in this case the green spaces of Lisbon, as a case study. This means that a user in Lisbon, entering in a green space, will be notified and asked to fill up a survey. Furthermore the Profile Extension Module was programmed, forming a new set of in scope questions to complement the profile of the user.

To fulfil the purpose of the surveying, this project extended the generic survey component to load JSON format questions that are stored on the server, being sent to the mobile interface. There are two survey sections presented: one is dedicated to describe the user experience while in the UGS and the other allows the user to mark points of interest in the maps. For each submit from the user, a new survey question is loaded from the server, using the HTTP Request that is previously set on the generic tool. This mechanism represents the dynamic surveying option that is up for usage on the GCS tool. Lastly, all the form questions, being progressively answered, are stored on the database in a JSON format that can be used for later usage.

On this project, no differentiated feedback was used, being only displayed a message originated by the immediate feedback mechanism. The Server Immediate Feedback Extension module is programmed to send a simple response message to the mobile request. The Differentiated module is then left blank, not being used but allowing the system flow to work normally. The Results Extension display the user inputs, loading the sensor information stored on the database and presenting them to the user. This data can be displayed in the form of markers or answers, providing in a simple UI all the information of interest of urban green spaces.

Chapter 7

Results

In the phase of this dissertation planning, it was developed a report in the scope of Introduction to the Research in Electrical and Computer Engineering. On this report was mentioned a Testing stage that would take place after the Extension of the Generic Crowdsourcing System, the eFlechten application. After the deployment of eFlechten, this stage would have allowed to test the developed system application, with the help of volunteered participants in a defined area, following the standard lichen sampling adapted to this Citizen Science Scope.

The participants, equipped with mobile devices with GPS location services and internet access were going to have the Expo application downloaded, and after the credentials insertion, the eFlechten would be available for usage. This phase would last for 2 to 3 weeks in a specific area. Since the problem was proposed by the Centre for Ecology, Evolution and Environmental Changes, based at the Science Faculty of Lisbon University, the surrounding areas like Campo Grande would be an adequate location for the model evaluation. The surveying phase will be followed by the responsible Researchers, reporting issues and errors that might come up. After this phase all the inserted data, along with the final processed outputs will be meticulously analyzed, comparing the results with some existing ones, elaborating a final scientific report about the developed tool.

7.1 Usability Evaluation of eFlechten

System Usability Scale (SUS) [89] is a simple, ten-item scale giving a global view of subjective assessments of usability. It covers a variety of aspects of system usability, such as the need for support, training, and complexity, and thus have a high level of face validity for measuring usability of a system.

SUS method consists on 10 items evaluated with a 5-point graphic scales, anchored at the end points with the terms ”Strongly agree” for 1 and ”Strongly disagree” for 5. Each item’s score contribution will range from 0 to 4. For items 1,3,5,7,and 9 the score contribution is the scale position minus 1. For items 2,4,6,8 and 10, the contribution is 5 minus the scale position. To obtain the overall value of SUS a multiplication to the sum of the scores by 2.5 is needed, getting a SUS score range of 0 to 100.

Due to external circumstances that conditioned the planned Testing phase, the methodology to be

used had to be restructured. So, in order to overcome this impracticability, an online survey to evaluate the developed system was considered. In name of simplicity, a Google Forms was developed with the intention of being sent and answered, having as targets researchers and non researchers.

This Google Forms would be divided into two sections: the User Identification and the System Usability Scale. SUS is narrowed to the eFlechten System, which means it is required the installation and usage of the mobile application. On Appendix A is the Installation Guide sent to the testing users. It covers Android and iOS users and presents a simple and perceptible step guide for the app installation.

To obtain a complete usage of the application, a set of steps after installation were required to follow. Due to the world's circumstances, a dummy survey submission was considered to be the most approachable way of evaluation. To provide the Google Forms for the final app evaluation, a test is made to the user, analysing if the usage scheme was correctly followed.

- Register, using the normal form or a third party app
- Login, using the normal form or a third party app
- Survey submission of a tree with smooth bark, with 100 cm of diameter.
- Survey submission of two random lichens with two random occupation percentages.
- Results Analysis Test: point out the metrics that are available on the eFlechten app.
- Explore the application and, if possible, submit a real survey in the field.

After the usage scheme validation based on the test answer, a Google Forms was sent to the testing user. On Appendix B is the Google Forms Results of the eFlechten System.

7.1.1 User Identification Results

This section pretends to point out some results of the Google Forms regarding the user profile. The user sampling has only 2,9% of Researchers, resulting in a majority of 97,1% of non Researchers. 97,1% was too the percentage of users with age comprehended between 18 and 25. The mobile operating system of the users was balanced, having 60% of Android users. Most of the users have already used a crowdsourcing application, with an advantage of 62,9%. On Appendix B on the Section B.1 is represented the User Identification graphics extracted from the Google Forms Analysis.

7.1.2 SUS Results

In order to make the System Usability Scale, an Excel table was generated based on the results from the SUS section in the Google Forms. Then the formula was applied to the available cells getting the SUS score for each result. The average of the SUS score was 82.4 in a 100 score range, having 9% of the results bellow 50, 89% of the results above 60 and 34% greater than 90. On Appendix B on the Section B.2 is represented the SUS Results graphics extracted from the Google Forms Analysis.

7.1.3 Analysis

Regarding the user profile there was a low percentage of researchers and elder users which is expectable due to the user sampling that have filled up the Surveys. Regardless of the somewhat low percentage of 37,1% of users who never used a crowdsourcing application, should come as no surprise if that value is lower in reality for the amount of nowadays applications that use some kind of crowdsourcing.

The SUS avarage score was beyond positive, proving that its UI is quite simple and perceptible, making eFlechten a powerful but easy tool of mapping lichens, in the scope of climate change and air pollution. It also contributes to the proof of concept of the extensibility of the Generic Crowdsourcing System.

Besides the surveying fields described from the Google Forms, there was a optional long answer suggestion box. There were some user inputs regarding bugs related to the third party login with Facebook. This happens because Facebook OAuth needs a Key Hash that was generated on Expo, validating the application with Facebook. But this generated Key sometimes changes from smartphone to smartphone since the application is hosted on Expo and is not natively installed. This was too a mentioned suggestion, since eFlechten is deployed on the Expo environment which is not ideal nor practical.

7.2 Requirements Achievement

On Subsection 3.1.2, the framework functionalities of the GCS were highlighted. The achievement of the raised requirements for each mentioned feature will prove the success of the framework development.

- **R1 - The framework should allow the registration of users.**
  - A registration mechanism was created, using the client interface to obtain credentials that are processed by the Server and saved on the database.
- **R2 - The framework should allow the login of users.**
  - A login mechanism was developed using a caching service, storing the user login session using its email.
- **R3 - The user authentication should be made natively or using third party applications.**
  - Allowed by the OAuth technology, the GCS can establish a authentication with a third party platform, allowing registration and login.
- **R4 - The framework should store a Profile of the user.**
  - One of the sections on the Menu Screen, displayed after login, is the Profile Section. This section displays the name and the photo of the user, if available.
- **R5 - The framework should allow the extensibility of the Profile.**

  - The Profile Section can be appended with an Extended Profile. This allows user extra information to be changed, stored or displayed.

- **R6 - The framework should allow the implementation of a Ranking system.**
  - The GCS allows the update of the user ranking points. The Profile Section displays them with the user ranking level.

- **R7 - The framework should allow a surveying mechanism.**
  - On Survey Section, forms can rendered based on an upcoming configuration file. This section can be acessed by the Menu Screen and is allocated to store the survey to display to the user in real time.

- **R8 - The framework should provide static or dynamic surveys.**
  - According from what is requested from the mobile application, the server endpoint will send a static or dynamic survey. JSON configuration files are sent according to what surveying method is defined.

- **R9 - The framework should use a external library that renders surveys and handles data, while providing extensibility.**
  - After developing the 'react-native-json-forms'[81], a library could be used on the Surveys Section. This library can render a UI with the survey based on a JSON configuration file. Data is handled according to what was programmed. This library also allows extensibility by accepting natively stored custom survey components.

- **R10 - The framework should allow data validation.**
  - On web application, Researcher can perform data validation, discarding dubious inputs.

- **R11 - The framework should allow data download.**
  - On web application, Researcher can download survey answers in the format that was programmed.

- **R12 - The framework should allow immediate and differentiated feedback.**
  - GCS provides a Server endpoint that, upon request from the mobile application, activates immediate and/or differentiated feedback. The first consists on a simple object that is sent as a response, with no data processing required. The second is composed by several sequential modules that process survey data asynchronously. The results are stored and are accessed by the mobile application with the Results Section.

- **R13 - The framework should allow display the processing results to the user.**
  - On Results Section the mobile application can display the most important findings related to the survey outcomes.

- **R14 - The framework should allow a Help Guide Presentation.**
  - On the Menu screen of the mobile application is possible to find a horizontal carousel component that presents a tutorial about the application usage and other relevant information.
- **R15 - The framework should allow database and caching service.**
  - For user login session state a caching service was used, based on Memcached framework. For user information, survey answers and processed data it was used MongoDB, a document based database.
- **R16 - The framework should be a singular source code.**
  - The Generic Crowdsourcing System, hosted in its whole on a GitHub Page [83]. This singular source code was replicated and extended into two different projects, the Urban Green Spaces developed by Miguel Leitão [88] and, on this thesis scope, eFlechten, present on a GitHub Page [90].
- **R17 - The framework should be tested and monitored.**
  - The GCS was tested and monitored in the Expo during development. Updates and dependencies should constantly tracked since deprecation are frequent.
- **R18 - The framework should be low weight, simple, fast and understandable for the programmer.**
  - The GCS used simple, popular and low weight development languages. The Expo provides a development environment that can be deployed in a wide range of devices and modifications can be observed in run time, not always requiring a full application render.
- **R19 - The framework should be user friendly.**
  - The GCS was focused on the simplicity required for a crowdsourcing project, balancing between the user experience and the functionality of the tool.
- **R20 - The framework should be compatible with Android and iOS.**
  - Using React Native, it was possible to develop for a wide range of devices since this language creates native apps for Android and iOS. Expo contributes to this React feature, providing to the programmer a run time environment that could be used on both operating systems.
- **R21 - The framework should be prepared for external attacks.**
  - Encryption of passwords, usage of HTTP requests using POST, OAuth and some built-in web security framework options was used on this system, contributing to a greater safety.
- **R22 - The framework should preserve the user privacy.**

– To the GCS, only the name, email and password is required, being stored on the database. Encryption of passwords is done on the server with the comparison of the encrypted objects, never performing a decryption of the user credentials. To access user smartphone sensors, permissions have to be granted to the application. If the permission access is denied by the user, no further survey action can be taken.

7.3 Code Reuse

One objective of the GCS was to provide a tool that could be used in multiple projects with minimized development effort. One measure of such effort reduction is the number of lines of code that a final application would reuse from GCS. Using 'VS Code Counter' [91], a powerful code analysing tool, it was possible to obtain some conclusions when comparing the Generic Crowdsourcing System with eFlechten, the extended framework on this thesis scope. Inside each system component, the external modules that the developed projects depends upon, inside the *node_modules* folder and 'package-lock.json', a large file that describes the exact dependency tree generated, were both ignored in this analysis. Thus, it was possible to obtain a 79.1% usage of the Generic Crowdsourcing System on the eFlechten framework, having this way 20.92% lines of code that belongs only to the extension fraction. JavaScript, the most dominant language used on eFlechten had a 97.4% of presence, while CSS had 1.7% and HTML 0.9%, both used on files of the web application.

7.4 Safety

In order to make a system trustable, Safety is an aspect that is important to take in consideration. Software Safety looks at how the software interacts and what levels of controls it has in the control of potentially unsafe hardware or systems. Using defensive coding techniques or making the correct technological choices can increase the level of safety of a software system.

On the Generic Crowdsourcing System and eFlechten, one defensive coding technique was the encryption of the passwords. It was used 'crypto', a module in Node.js which deals with an algorithm that performs data encryption and decryption. A method of this module, createCipheriv, was used on the login and on the register server endpoint, encrypting the user password to be stored on the database. This method uses an algorithm and an initialization vector (iv), a fixed-size input to a cryptographic primitive that is typically required to be random or pseudorandom.

Concerning the HTTP Requests, GET parameters are passed via URL. This means that parameters are stored in server logs, and browser history. When using GET, it makes it very easy to alter the data being submitted to the server as well. That is why confidential data is sent with a POST method, since the information is sent on the message body.

When registering the application using OAuth 2.0, it is generated the client ID, a public identifier for the app and optionally a secret, known only to the application and the authorization server. Like single-page apps, mobile apps also cannot maintain the confidentiality of a client secret which can compromise

the system security. That is why Generic Crowdsourcing System and eFlechten don't store the client secret and uses the Authorization Flow. With this an external browser is launched in order to ensure the native app cannot modify the browser window or inspect the contents.

Like single-page apps, mobile apps also cannot maintain the confidentiality of a client secret. Because of this, mobile apps must also use an OAuth flow that does not require a client secret. The current best practice is to use the Authorization Flow along with launching an external browser, in order to ensure the native app cannot modify the browser window or inspect the contents.

Cross-Site Request Forgery (CSRF) is when a attacker might attempt to inject a request to the redirect URI of the legitimate client on the victim's device, e.g., to cause the client to access resources under the attacker's control. Axios, used on the web application, comes built-in with some web security by protecting users against attacks such as CSRF, increasing the security of the HTTP requests and their responses done with this promised-based library.

The eFlechten Server is hosted on a INESC-ID Apache Server System, being one of the most popular web servers available for both Windows and Linux/UNIX. At the moment, it is used to host approximately 40% of websites and it is also often described as one of the most secure web servers.

Chapter 8

Conclusions

Human activity is affecting more and more the ecosystems that are crucial to the Earth dynamic fauna and flora. Climate change has large impacts on global scale: air pollutants are increasing day by day, contributing to Urban heat island effect, for example.

It is highly important that this humanity issue is tackled, otherwise the effects will be drastic and unprecedented in scale. For that, it is necessary that researchers from all over the world have the most efficient and accurate tools available, avoiding calculation errors and actions based on faulty values or conclusions. One way to overcome this unreliability is to have more and more information about a certain scientific field. But, on the other hand, the lack of time and resources aligned with the amount of variables and the Earth dimension contribute negatively to this ideal scenario.

Citizen Science, an emerging new form of interaction between scientists and citizens, allows social participation and involvement in scientific activities. Through surveying methods, even users with no scientific knowledge can contribute positively to the amount of information that exists in several scientific fields.

Lichens, being a powerful climate proxy, can provide diverse information about their surroundings. Modern lichen-based environmental analyses provide low cost, high-resolution spatial tools for modelling and mapping the effects of climate change that can complement the traditional networks of pollution or climate monitoring stations. Aligned with the principles of Citizen Science, lichen sampling methods based on their species richness, lichen cover, and lichen community composition, can contribute positively to the information scarceness.

8.1 Achievements

For this thesis, it was developed a Generic Crowdsourcing System, a powerful generic surveying tool composed with a Server and two client interfaces: a mobile and a web application, for complementary usage. The main objective of this generic system is to be possible to adapt to any scientific crowdsourcing need, regardless of the scope in question. The programmer, along with the team of the Researchers, will adapt this tool and extend it to fulfil the narrowed requirements raised by the scientific force. With

little programming effort, the GCS can be extended, making small changes to the existing extension files and modules, fulfilling the requirements any scientific scope.

As a proof of concept of the GCS adapatibility, eFlechten System was developed to fulfill the narrowed requirements upraised by the Centre for Ecology, Evolution and Environmental Changes (cE3c) - FCUL, supported by iLTER. By making a few changes to the 'skeleton' of the Generic Crowdsourcing System, it was possible to extend, adapting native features to the surveying needs of a given scientific framework.

eFlechten, capable of returning immediate feedback to the user, is a crowdsourcing application inherited from the Generic Crowdsourcing System that allows to map lichens in order to fight the scarceness of air pollution and climate change data. It provides a Register and a Login Section, with a normal and third party app authentication method. With an average SUS score of 82.4, eFlechten displays a menu screen that provides simple UI features to the user. It provides a Survey that lets users to insert tree trunk information and a lichens image picker that allows user to identify the species and the respective coverage. This sampling lichen method is explained in the Help Guide, present on the Menu Screen. There's also available the Results section, containing a map that presents the three metrics relevant for the climate change and air pollution scope. Aridity, Poleotolerance and Eutrophication indexes are represented in the form of heatmaps, representing a dynamic and interactive way of displaying the metric indexes to the user. For each survey submission, a Ranking component updates the user rank points, introducing a gamification level to this application.

8.2 Future Work

Regarding the Generic Crowdsourcing System, despite of the minimal programming effort required for extending it to fulfil narrowed scientific scopes, there is a lack of modular activated modes. This could be useful to customize the core application, modifying, adding or removing blocks automatically. For example, if a certain scientific research team wants to add a chatroom section to the Menu Screen it could be done by simply turning a boolean value from a configuration file to true. This would load a pre-defined component, adjusting the whole UI to fulfil the desired requirements. A dynamism force would be added to the GCS, aligning with to the existent surveying and feedback extension mechanism that is already efficiently implemented.

The GCS is also a on progress tool that should be constantly revised and updated, keeping dependencies deprecations or updates on track. Plus, there is always room for improvement, ensuring a better file organization, modularizing the existing code for better interpretation and possible modifications. Safer communication mechanisms should be studied, avoiding external attacks.

Due to the Citizen Science scope that eFlechten is in, survey data can contain errors or misunderstandings that could put at risk system data reliability. Consequently, another field that could be added to the eFlechten Survey along with the LichensImagePicker is a camera element, available on 'react-native-json-forms'. This would allow the user to take a picture to the surveying area, making visible the sampling lichens on the tree bark. Upon Survey submission, eFlechten Server could provide an AI Image Recognition System, powered by deep learning, specifically Convolutional Neural Networks

(CNN), a neural network architecture that is biological inspired on a connectivity pattern that resembles the organization of the visual cortex, being most commonly applied to analyzing visual imagery. This recognition system could predict the species from the image present in the user survey. With a given threshold, any incongruity detected between the user lichen species identification and the recognition system output from the survey data could be notified to the Researchers. This technique would require some eFlechten coding modifications but would provide a greater system data reliability.

eFlechten is published and hosted on Expo, a development environment tool that is available for iOS and Android. Despite its great features and easiness in terms of workflow, the Subsection 7.1.3 revealed that most users showed interest on having the eFlechten application on App Store or Google Play. This would ease the installation process and make it more appealing to the general public. The deployment on the digital distribution platforms would resolve the issues raised on this same Section regarding the third party login with Facebook, since a static Android Key Hash would be supplied by the Google Play upon deployment.

Since eFlecthen requires on field sampling, some areas could not have strong connection to the internet or none at all. Likewise, the GPS location can be inaccurate or unavailable. This represents a downside of this application since the survey submission must be done on the sampling location. So, a efficient way to tackle this problem is to offer an offline mode of eFlechten. This means that in case of network failure, is possible to store data in a mobile caching system, for example using 'react-native-cache' [92], and submit it to the server when network is available. Regarding the GPS location, if unavailable, the user could have a manual mode, inserting the coordinates of the sampling location or to pin a marker on a displayed map.

Bibliography

[1] L. See, P. Mooney, G. Foody, L. Bastin, A. Comber, J. Estima, S. Fritz, N. Kerle, B. Jiang, M. Laakso, et al. Crowdsourcing, citizen science or volunteered geographic information? the current state of crowdsourced geographic information. *ISPRS International Journal of Geo-Information*, 5(5):55, 2016.

[2] Air quality and climate change research. `https://www.epa.gov/air-research/air-quality-and-climate-change-research`.

[3] M. C. Ribeiro, P. Pinho, C. Branquinho, E. Llop, and M. J. Pereira. Geostatistical uncertainty of assessing air quality using high-spatial-resolution lichen data: A health study in the urban area of sines, portugal. *Science of the Total Environment*, 562:740–750, 2016.

[4] W. Li, Y. Sun, D. Meng, and X. Li. Analysis of beijing's urban heat-island under the influence of extrme heat based on hj-1b data. In *2013 IEEE International Geoscience and Remote Sensing Symposium-IGARSS*, pages 3845–3848. IEEE, 2013.

[5] Climate change. `https://www.un.org/en/sections/issues-depth/climate-change/`.

[6] M. E. Hale. How to know the lichens. dubuque: William c, 1979.

[7] E. Llop, P. Pinho, P. Matos, M. J. Pereira, and C. Branquinho. The use of lichen functional groups as indicators of air quality in a mediterranean urban environment. *Ecological indicators*, 13(1): 215–221, 2012.

[8] J. Corburn. Cities, climate change and urban heat island mitigation: localising global environmental science. *Urban studies*, 46(2):413–427, 2009.

[9] G. D'Amato, L. Cecchi, M. D'Amato, and I. Annesi-Maesano. Climate change and respiratory diseases, 2014.

[10] M. Baaghideh and F. Mayvaneh. Climate change and simulation of cardiovascular disease mortality: A case study of mashhad, iran. *Iranian journal of public health*, 46(3):396, 2017.

[11] J. Vieira, P. Matos, T. Mexia, P. Silva, N. Lopes, C. Freitas, O. Correia, M. Santos-Reis, C. Branquinho, and P. Pinho. Green spaces are not all the same for the provision of air purification and climate regulation services: The case of urban parks. *Environmental research*, 160:306–313, 2018.

[12] Onu: Sustainable development goals. https://www.un.org/sustainabledevelopment/sustainable-development-goals/.

[13] S. Rahmstorf, A. Cazenave, J. A. Church, J. E. Hansen, R. F. Keeling, D. E. Parker, and R. C. Somerville. Recent climate observations compared to projections. *Science*, 316(5825):709–709, 2007.

[14] I. P. O. C. Change. Climate change 2007: The physical science basis. *Agenda*, 6(07):333, 2007.

[15] J. A. Patz, D. Campbell-Lendrum, T. Holloway, and J. A. Foley. Impact of regional climate change on human health. *Nature*, 438(7066):310–317, 2005.

[16] C. Guerreiro. Air quality in europe: 2013 report. 2013.

[17] V. H. Dale. The relationship between land-use change and climate change. *Ecological applications*, 7(3):753–769, 1997.

[18] K. C. Seto, R. Sánchez-Rodríguez, and M. Fragkias. The new geography of contemporary urbanization and the environment. *Annual review of environment and resources*, 35:167–194, 2010.

[19] H.-M. Füssel, A. Jol, et al. Climate change, impacts and vulnerability in europe 2012 an indicator-based report. 2012.

[20] F. OECD. Fdi in figures, 2016.

[21] J. Lelieveld, J. S. Evans, M. Fnais, D. Giannadaki, and A. Pozzer. The contribution of outdoor air pollution sources to premature mortality on a global scale. *Nature*, 525(7569):367, 2015.

[22] S. Hales, S. Kovats, S. Lloyd, and D. Campbell-Lendrum. *Quantitative risk assessment of the effects of climate change on selected causes of death, 2030s and 2050s*. World Health Organization, 2014.

[23] K. Tzoulas, K. Korpela, S. Venn, V. Yli-Pelkonen, A. Kaźmierczak, J. Niemela, and P. James. Promoting ecosystem and human health in urban areas using green infrastructure: A literature review. *Landscape and urban planning*, 81(3):167–178, 2007.

[24] M. Jansson. Green space in compact cities: the benefits and values of urban ecosystem services in planning. *NA*, 26(2), 2014.

[25] Monitoring climate change in “data-scarce regions”. https://www.rural21.com/english/news/detail/article/monitoring-climate-change-in-data-scarce-regions.html.

[26] A. P. Møller. Environmental indicators of climate change: phenological aspects. In *Environmental Indicators*, pages 39–49. Springer, 2015.

[27] M. Conti and G. Cecchetti. Biological monitoring: lichens as bioindicators of air pollution assessment—a review. *Environmental pollution*, 114(3):471–492, 2001.

[28] M. I. Käffer, A. T. Lemos, M. A. Apel, J. V. Rocha, S. M. de Azevedo Martins, and V. M. F. Vargas. Use of bioindicators to evaluate air quality and genotoxic compounds in an urban environment in southern brazil. *Environmental pollution*, 163:24–31, 2012.

[29] P. L. Nimis, C. Scheidegger, and P. A. Wolseley. Monitoring with lichens—monitoring lichens. In *Monitoring with Lichens—Monitoring Lichens*, pages 1–4. Springer, 2002.

[30] C. Cislaghi and P. L. Nimis. Lichens, air pollution and lung cancer. *Nature*, 387(6632):463, 1997.

[31] A. Tonneijck and A. Posthumus. Use of indicator plants for biological monitoring of effects of air pollution: the dutch approach. Technical report, 1987.

[32] P. Pinho, S. Augusto, C. Maguas, M. Pereira, A. Soares, and C. Branquinho. Impact of neighbourhood land-cover in epiphytic lichen diversity: analysis of multiple factors working at different spatial scales. *Environmental Pollution*, 151(2):414–422, 2008.

[33] P. Pinho, S. Augusto, M. Martins-Loução, M. Pereira, A. Soares, C. Máguas, and C. Branquinho. Causes of change in nitrophytic and oligotrophic lichen species in a mediterranean climate: impact of land cover and atmospheric pollutants. *Environmental Pollution*, 154(3):380–389, 2008.

[34] C. Branquinho, G. Gaio-Oliveira, S. Augusto, P. Pinho, C. Máguas, and O. Correia. Biomonitoring spatial and temporal impact of atmospheric dust from a cement industry. *Environmental Pollution*, 151(2):292–299, 2008.

[35] N. M. Koch, C. Branquinho, P. Matos, P. Pinho, F. Lucheta, S. M. Martins, and V. M. Vargas. The application of lichens as ecological surrogates of air pollution in the subtropics: a case study in south brazil. *Environmental Science and Pollution Research*, 23(20):20819–20834, 2016.

[36] S. Munzi, S. Ravera, and G. Caneva. Epiphytic lichens as indicators of environmental quality in rome. *Environmental Pollution*, 146(2):350–358, 2007.

[37] H. A. Carreras and M. L. Pignata. Biomonitoring of heavy metals and air quality in cordoba city, argentina, using transplanted lichens. *Environmental Pollution*, 117(1):77–87, 2002.

[38] H. Van Dobben and A. De Bakker. Re-mapping epiphytic lichen biodiversity in the netherlands: effects of decreasing so2 and increasing nh3. *Acta Botanica Neerlandica*, 45(1):55–71, 1996.

[39] C. Branquinho, P. Matos, A. R. Vieira, and M. M. P. Ramos. The relative impact of lichen symbiotic partners to repeated copper uptake. *Environmental and experimental botany*, 72(1):84–92, 2011.

[40] A. J. Carr. Why do we all need community science? *Society & Natural Resources*, 2004.

[41] Z. Chen, R. Fu, Z. Zhao, Z. Liu, L. Xia, L. Chen, P. Cheng, C. C. Cao, Y. Tong, and C. J. Zhang. gmission: A general spatial crowdsourcing platform. *Proceedings of the VLDB Endowment*, 7(13): 1629–1632, 2014.

[42] Waze. `https://www.waze.com/`.

[43] Chinacrowds. http://www.chinacrowds.com/.

[44] G. Li, Y. Zheng, J. Fan, J. Wang, and R. Cheng. Crowdsourced data management: Overview and challenges. In *Proceedings of the 2017 ACM International Conference on Management of Data*, pages 1711–1716, 2017.

[45] I. Blohm, S. Zogaj, U. Bretschneider, and J. M. Leimeister. How to manage crowdsourcing platforms effectively? *California Management Review*, 60(2):122–149, 2018.

[46] A. D. Shaw, J. J. Horton, and D. L. Chen. Designing incentives for inexpert human raters. In *Proceedings of the ACM 2011 conference on Computer supported cooperative work*, pages 275–284, 2011.

[47] T. Hoßfeld and C. Keimel. Crowdsourcing in qoe evaluation. In *Quality of experience*, pages 315–327. Springer, 2014.

[48] C. Eickhoff, C. G. Harris, A. P. de Vries, and P. Srinivasan. Quality through flow and immersion: gamifying crowdsourced relevance assessments. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, pages 871–880, 2012.

[49] G. Graham, J. Cox, B. Simmons, C. Lintott, K. Masters, A. Greenhill, and K. Holmes. How is success defined and measured in online citizen science: a case study of zooniverse projects. *Computing in science and engineering, PP (99)(22). ISSN*, pages 1521–9615, 2015.

[50] K. Crowston and N. R. Prestopnik. Motivation and data quality in a citizen science game: A design science evaluation. In *2013 46th Hawaii International Conference on System Sciences*, pages 450–459. IEEE, 2013.

[51] A. Baruch, A. May, and D. Yu. The motivations, enablers and barriers for voluntary participation in an online crowdsourcing platform. *Computers in Human Behavior*, 64:923–931, 2016.

[52] R. M. Borromeo, T. Laurent, and M. Toyama. The influence of crowd type and task complexity on crowdsourced work quality. In *Proceedings of the 20th International Database Engineering & Applications Symposium*, pages 70–76, 2016.

[53] A. Finnerty, P. Kucherbaev, S. Tranquillini, and G. Convertino. Keep it simple: Reward and task design in crowdsourcing. In *Proceedings of the Biannual Conference of the Italian Chapter of SIGCHI*, pages 1–4, 2013.

[54] Proposta de plano de ação local para a biodiversidade em lisboa. https://www.am-lisboa.pt/documentos/1455213072E9qKI7fe1Fh81KO6.pdf.

[55] Lisbon fauna monitorization survey. https://app.maptionnaire.com/pt/7036/, .

[56] C. A. Lepczyk. Integrating published data and citizen science to describe bird diversity across a landscape. *Journal of Applied Ecology*, 42(4):672–677, 2005.

[57] R. E. McCaffrey. Using citizen science in urban bird studies. *Urban habitats*, 3(1):70–86, 2005.

[58] Fulcrum. https://www.fulcrumapp.com/.

[59] Device magic. https://www.devicemagic.com/, .

[60] Magpi. https://home.magpi.com/, .

[61] Fastfield. https://www.fastfieldforms.com/.

[62] B. Haltofová. Implementation of geo-crowdsourcing mobile applications in e-government of v4 countries: A state-of-the-art survey. *World Academy of Science, Engineering and Technology, International Journal of Computer, Electrical, Automation, Control and Information Engineering*, 11 (5):568–572, 2017.

[63] Zmapujto. https://www.zmapujto.cz/.

[64] Dejtip. http://dejtip.eu/.

[65] T-mapy. https://www.tmapy.cz/.

[66] H. M. Pereira and H. D. Cooper. Towards the global monitoring of biodiversity change. *Trends in Ecology & Evolution*, 21(3):123–129, 2006.

[67] P. Matos, L. Geiser, A. Hardman, D. Glavich, P. Pinho, A. Nunes, A. M. Soares, and C. Branquinho. Tracking global change using lichen diversity: towards a global-scale ecological indicator. *Methods in Ecology and Evolution*, 8(7):788–798, 2017.

[68] C4 model. https://c4model.com/.

[69] B. Singh and S. Kannojia. Languages and their importance in quality software. In *2012 International Conference on Communication Systems and Network Technologies*, pages 956–959. IEEE, 2012.

[70] What javascript framework to choose in 2020: A comparison. https://huspi.com/blog-open/what-javascript-framework-to-choose-in-2020-a-comparison.

[71] S. H. Jensen, A. Møller, and P. Thiemann. Type analysis for javascript. In *International Static Analysis Symposium*, pages 238–255. Springer, 2009.

[72] React native. https://reactnative.dev/.

[73] Node.js. https://nodejs.org/en/.

[74] Mongodb. https://www.mongodb.com/, .

[75] Memcached. https://memcached.org/.

[76] Expo. https://expo.io/.

[77] multer. https://www.npmjs.com/package/multer.

[78] Surveyjs. https://surveyjs.io/, .

[79] Surveymonkey. https://www.surveymonkey.com/, .

[80] Typeform. https://www.typeform.com/.

[81] Github: React native json forms. https://github.com/mleitao27/react-native-json-forms, .

[82] Npm: React native json forms. https://www.npmjs.com/package/react-native-json-forms.

[83] Generic crowdsourcing system. https://github.com/mleitao27/generic-crowdsourcing-sys.

[84] react-native-maps. https://github.com/react-native-maps/react-native-maps.

[85] Geojson. https://geojson.org/.

[86] Arcgis. https://www.arcgis.com/.

[87] Qgis. https://qgis.org/.

[88] Urban green spaces. https://github.com/mleitao27/gcs-urban-green-spaces.

[89] J. Brooke et al. Sus-a quick and dirty usability scale. *Usability evaluation in industry*, 189(194):4–7, 1996.

[90] eflechten. https://github.com/guilhermeneugenio/gcs-lichens-immediate-fb-sys, .

[91] Vscode counter. https://github.com/uctakeoff/vscode-counter.

[92] react-native-cache. https://www.npmjs.com/package/react-native-cache.

Appendix A

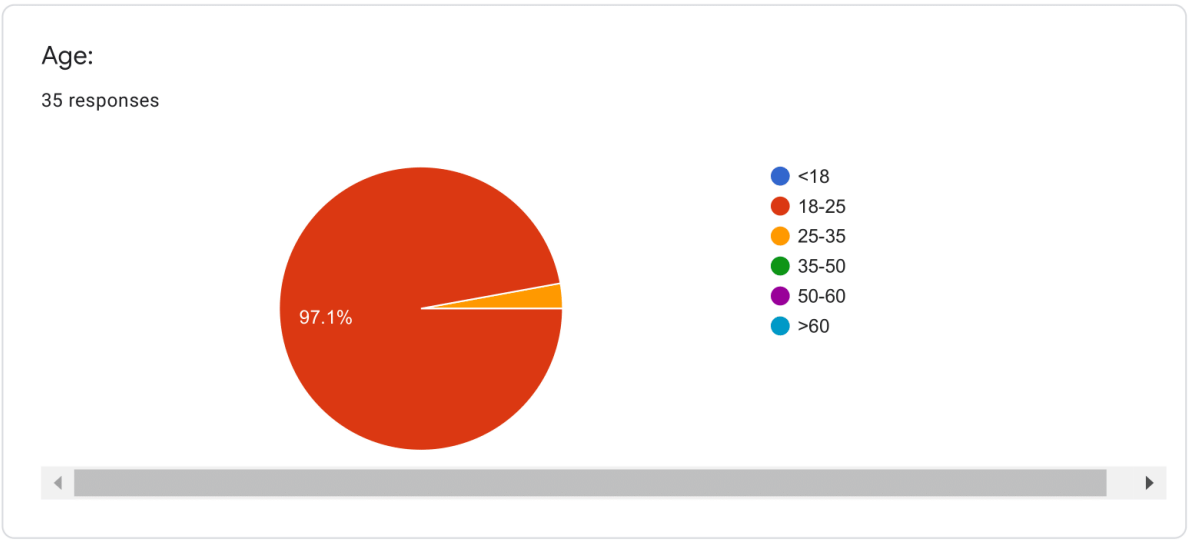
Installation Guide

Figure A.1: eFletchen Installation Guide.

Appendix B

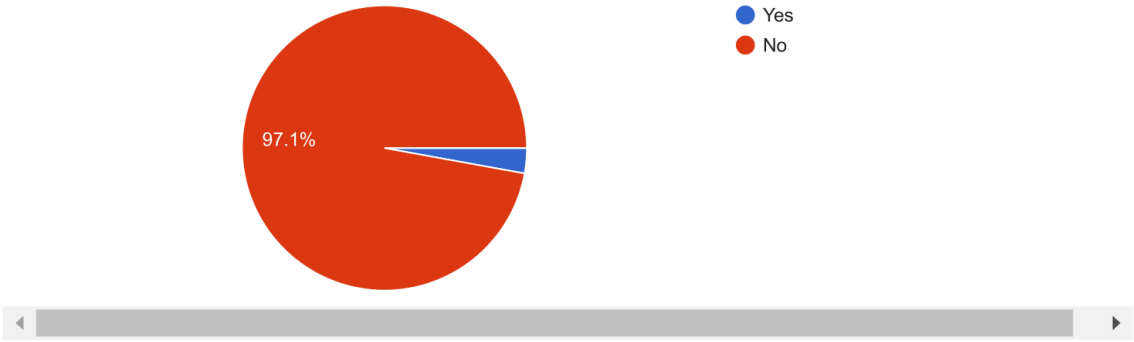
SUS Questionary Responses

B.1 User Identification



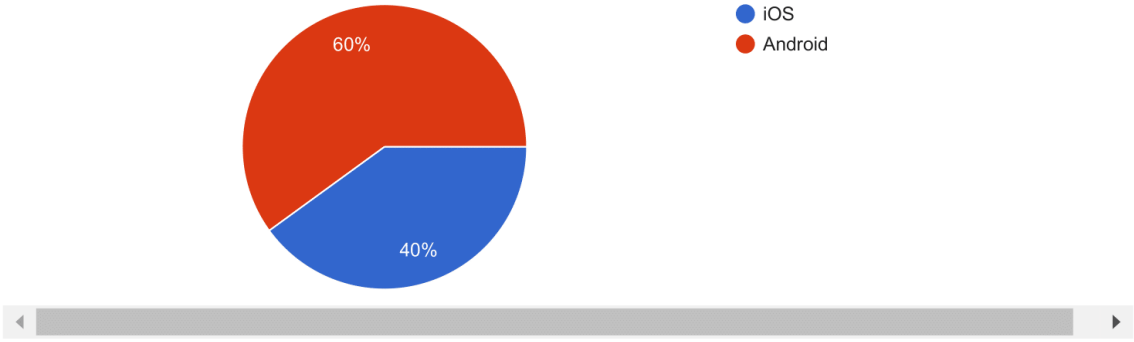
Are you a researcher?

35 responses



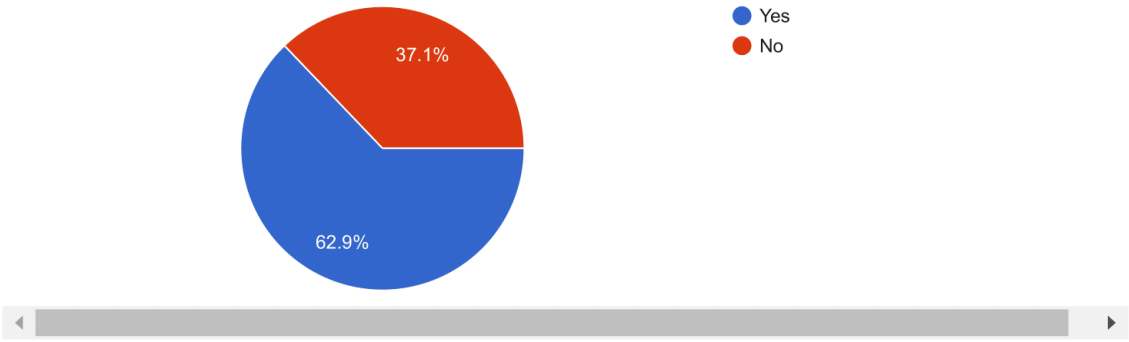
Mobile Operating System:

35 responses

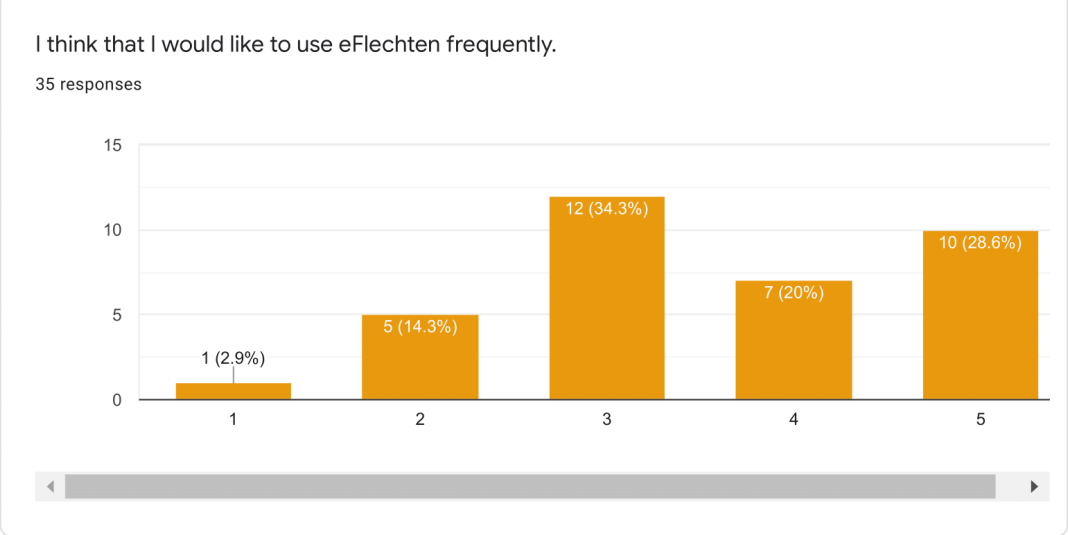


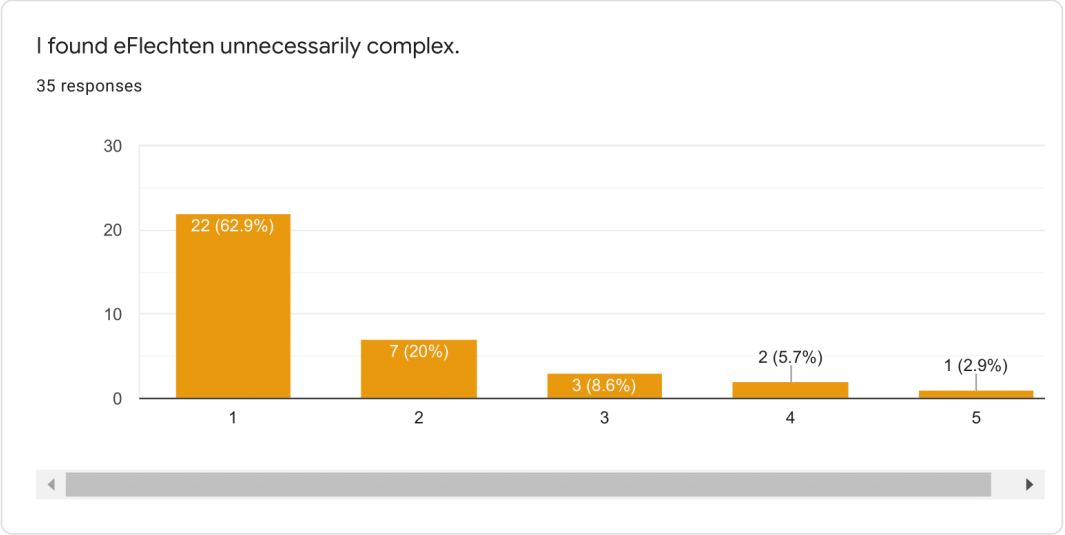
Have you ever used a crowdsourcing app?

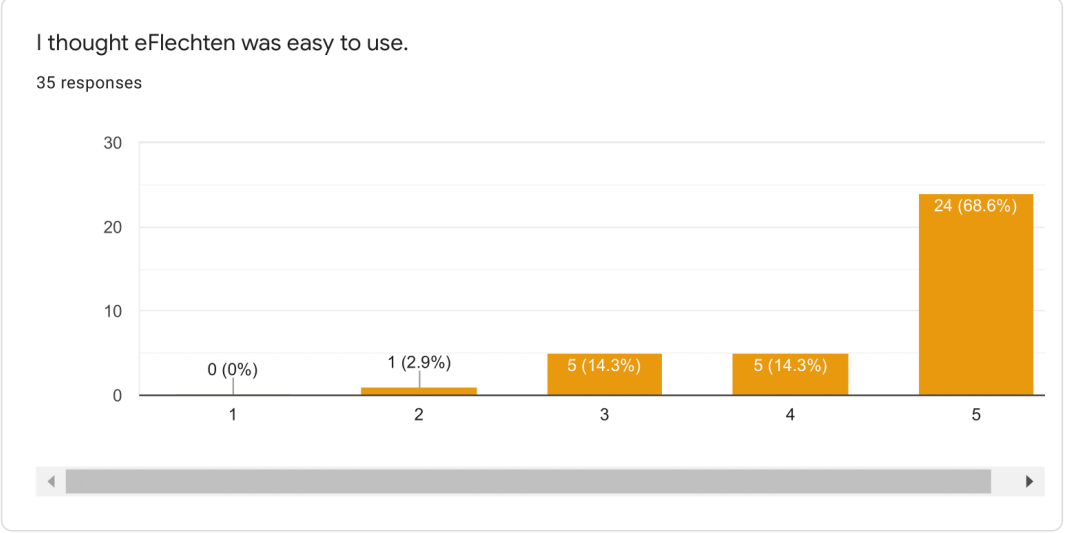
35 responses



B.2 System Usability Scale (SUS)

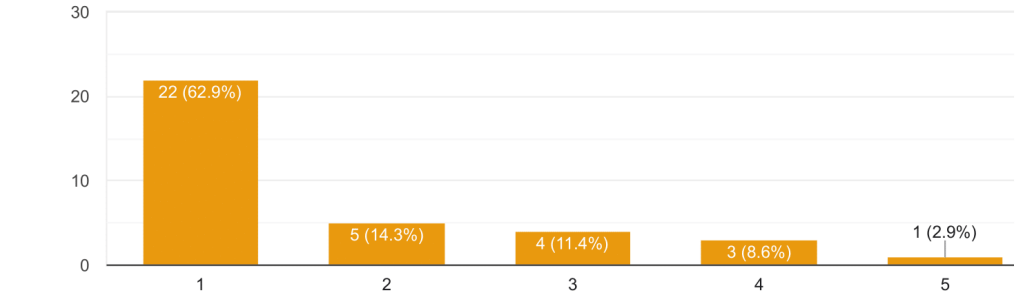






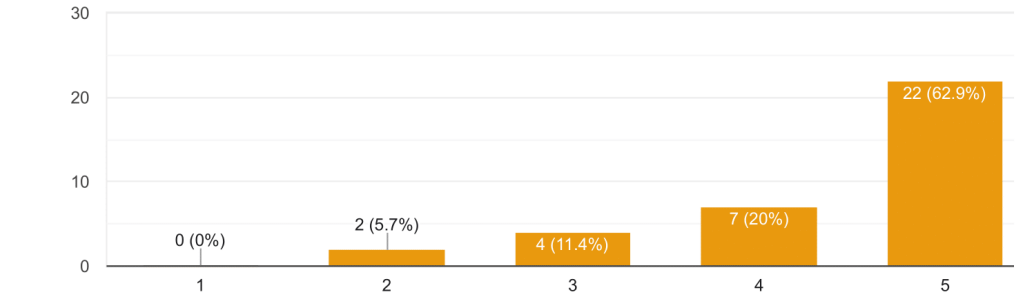
I think that I would need the support of a technical person to be able to use eFlechten.

35 responses



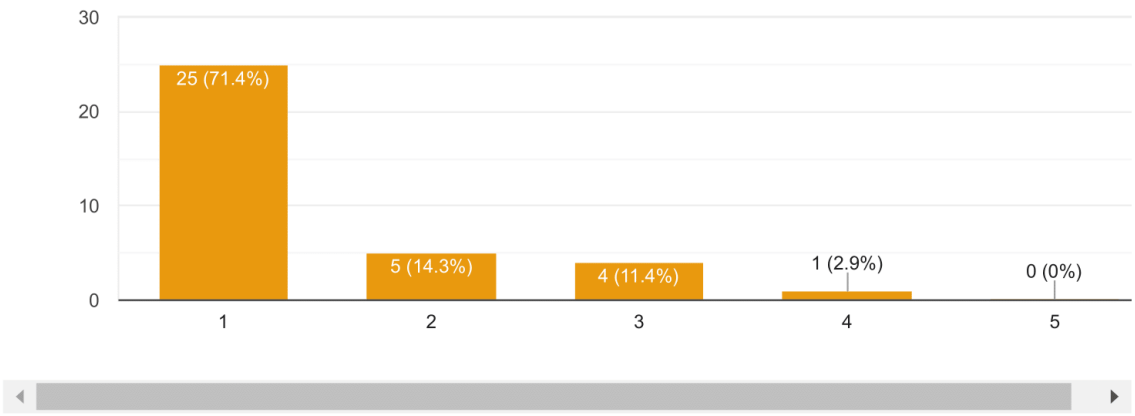
I found the various functions in eFlechten were well integrated.

35 responses



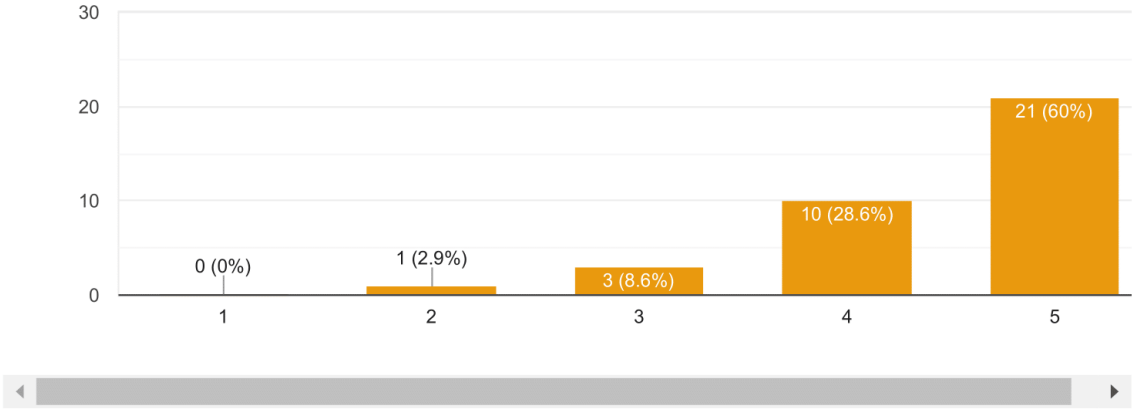
I thought there was too much inconsistency in eFlechten.

35 responses



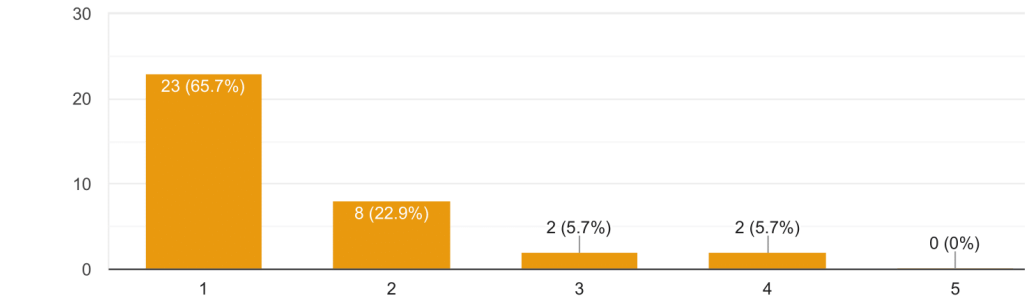
I would imagine that most people would learn to use eFlechten very quickly.

35 responses



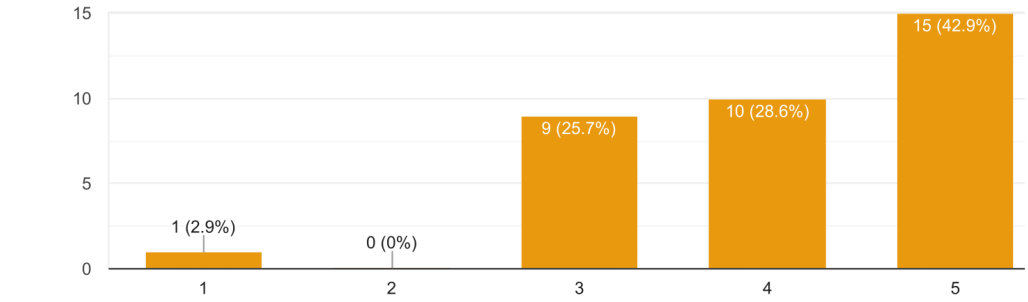
I found eFlechten very cumbersome (awkward) to use.

35 responses



I felt very confident using eFlechten.

35 responses



I needed to learn a lot of things before I could get going with eFlechten.

35 responses