EpiMass

A Clinical and Epidemiological Platform for Mass Gatherings

Miguel Felício Raposo Vieira

Thesis to obtain the Master of Science Degree in

Biomedical Engineering

Supervisors: Prof. Mário Jorge Costa Gaspar da Silva
Prof. Bruno Emanuel da Graça Martins

Examination Committee
Chairperson: Prof. João Miguel Raposo Sanches
Supervisor: Prof. Mário Jorge Costa Gaspar da Silva
Member of the Committee: Dr. Henrique Manuel Gil Martins

June 2018
Acknowledgements

I would like to thank all of those involved in this work, especially my supervisors, Prof. Mário Gaspar da Silva, Prof. Bruno Martins, and Dr. Ricardo Mexia. A special thanks to the Department of Epidemiology of Instituto Nacional de Saúde Doutor Ricardo Jorge, and to the teams present at the first aid posts of the 2017 Andanças Festival and the 2018 Torres Vedras Carnival.

This work was produced with the support of INCD funded by FCT and FEDER under the project 22153-01/SAICT/2016.
Abstract

This work presents a clinical and epidemiological web platform designed for supporting the basic needs of a first aid post during Mass Gatherings (MGs), called EpiMass. The World Health Organization (WHO) defines an MG as a concentration of people which has the potential to strain the local emergency response mechanisms. EpiMass was built over a 3-tier architecture, based on the MEAN stack (MongoDB, Express, Angular, and Node.js), i.e., a state-of-the-art open-source framework for developing web applications. It enables clinical and epidemiology professionals to perform actions and gather information relevant for their roles. The platform offers integrated data visualization tools and an API which gives the ability to extract the data for analysis with other platforms. This allows for near fast real-time reporting with a preliminary analysis giving some statistics and visual clues over the data, as well as the possibility to do a more thorough analysis with external specialized software. An early prototype of the platform was implemented and tested during the 2018 Torres Vedras Carnival, where it has shown to be viable to collect data during MG events.

Keywords: Mass Gatherings, Clinical Information Systems, Epidemiological Surveillance, Web Platforms, MEAN Stack
Resumo

Este trabalho apresenta uma plataforma web clínica e epidemiológica, denominada EpiMass, projetada para dar suporte às necessidades básicas de um posto de primeiros socorros durante eventos de massas. A Organização Mundial de Saúde (OMS) define um evento de massas como uma concentração de pessoas que tem o potencial de sobrecarregar os mecanismos locais de resposta a emergências. A EpiMass foi construída sobre uma arquitetura de três camadas, com base no stack MEAN (MongoDB, Express, Angular e Node.js), uma estrutura moderna de código aberto para desenvolver aplicações web. Permite aos profissionais de saúde e a epidemiologistas realizar determinadas ações e recolher informação relevante, de acordo com as suas funções. A plataforma disponibiliza ferramentas para visualização de dados, e uma API que permite a sua extração, de forma a poderem ser analisados por outras plataformas. Isto permite elaborar relatórios em tempo real e fazer uma análise preliminar sobre os dados, bem como possibilita fazer uma análise mais detalhada recorrendo a software especializado. Um protótipo inicial da plataforma foi implementado e testado durante o Carnaval de Torres Vedras, em 2018, onde a plataforma se mostrou viável para recolher dados durante eventos de massas.

Palavras-chave: Eventos de Massas, Sistemas de Informação Clínica, Vigilância Epidemiológica, Plataformas Web, Stack MEAN
# Contents

## List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>xi</td>
<td></td>
</tr>
</tbody>
</table>

## List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>xiii</td>
<td></td>
</tr>
</tbody>
</table>

### 1 Introduction

1.1 Objectives ........................................ 1
1.2 Contributions ..................................... 2
1.3 Methodology ........................................ 2
1.4 Outline of the Dissertation ......................... 3

### 2 Literature Review

2.1 Mass Gatherings ..................................... 5
2.2 Public Health Surveillance Systems .................... 7
2.3 Epidemiological Surveillance in Portugal ............... 9
2.4 Summary ............................................. 10

### 3 Technologies for Web Development

3.1 Web Platforms ....................................... 11
  3.1.1 3-tier Architecture ............................ 11
  3.1.2 REST APIs ....................................... 12
  3.1.3 MEAN Stack ..................................... 12
  3.1.4 Offline Operation ............................... 15
3.2 Security and Authentication of Communications .......... 16
  3.2.1 Server Authentication .......................... 16
  3.2.2 Client Authentication .......................... 17
3.3 Overview ............................................ 18

### 4 EpiMass

4.1 Requirements ........................................ 19
4.2 EpiMass Architecture ................................ 21
  4.2.1 User Interface ................................. 22
  4.2.2 Application Server ............................. 25
  4.2.3 Database ....................................... 26
4.3 Dashboards .......................................... 27
  4.3.1 Administration Dashboard ....................... 28
  4.3.2 Clinical Dashboard ............................. 29
4.3.3 Epidemiologist Dashboard ........................................ 34
4.3.4 Producer Dashboard ............................................. 34
4.4 Evaluation at the Torres Vedras Carnival .......................... 35
4.5 Overview .................................................................. 37

5 Conclusions ................................................................. 41
5.1 Further Developments .................................................. 41

References ................................................................. 43
List of Tables

2.1 Examples of MG events according to type. .................................................. 6
4.1 Functionalities according to each user role. ................................................. 22
## List of Figures

2.1 Location of music festivals in Portugal. Image credits belong to APORFEST. 6

3.1 3-tier architecture. 12

3.2 MEAN stack architecture. 13

3.3 Comparison between a table with two rows, on the left, and a collection of two JSON-like documents, on the right. 15

3.4 JWT diagram. Diagram available on https://jwt.io/introduction/. 17

4.1 Epidemiological surveillance team of the 2017 Andanças Festival. Source: INSA. 20

4.2 SVIG Andanças Festival. 20

4.3 Homepage, available at https://epimass.inesc-id.pt. 23

4.4 Login page. 24

4.5 Example of a success notification toast. 24

4.6 Personal profile page. 24

4.7 Administration dashboard. 28

4.8 Functionality of registering a new user. 29

4.9 Clinical dashboard. 30

4.10 Functionality of registering a new patient. 31

4.11 Functionality of searching for a patient. 32

4.12 New occurrence form. 33

4.13 Epidemiologist dashboard. 34

4.14 Producer dashboard. 35

4.15 Teams presented at the first aid posts of the 2018 Torres Vedras Carnival. Source: INSA. 36

4.16 Performance tests at the Torres Vedras Carnival. 37

4.17 SVIG Torres Vedras Carnival. 38

4.18 Distribution of the health occurrences by diagnostic group. 39

4.19 Distribution of the health occurrences by age group. 39

4.20 Distribution of the health occurrences by age group, filtering only men. 40

4.21 Distribution of the health occurrences by destiny. 40
Chapter 1: Introduction

The World Health Organization (2016) describes a Mass Gathering (MG) as "(...) a planned or spontaneous event where the number of people attending could strain the planning and response resources of the community or country hosting the event". Olympic Games, the FIFA World Cup, Hajj, and the Glastonbury Festival are all examples of MGs.

Portugal is a country particularly prone to hosting MGs, due to its climate, the importance of tourism and also its culture. The number of MG events in Portugal has been increasing over the years, mostly because of the increased number of cultural events, namely music festivals, where people from all over the world come together to attend performances by their favorite artists.

The challenges associated with this kind of events can place a strain on the local resources. The extent of their impact may vary according to the MG features, environmental factors, participant characteristics and venue characteristics. The main hazards include transmission of infectious diseases, injuries and trauma, illnesses due to the use of alcohol and drugs, environmental effects, natural disasters, and deliberate acts (World Health Organization, 2015). A sudden epidemic outbreak must also be taken into account at MGs. Therefore, public health surveillance systems have a crucial role in the early detection of a possible outbreak, in order to avoid its spread among the participants.

Despite the importance of these issues, an integrated tool for managing epidemiological risks is not available in Portugal yet. The size of most national MGs is not comparable to the scale of the Olympic Games or the Hajj, for which some work has already been done in this field, due to the magnitude of these events and the number of countries involved. Therefore, it is of utmost importance to develop a system for improving public health surveillance of MGs in Portugal.

The Department of Epidemiology of Instituto Nacional de Saúde Doutor Ricardo Jorge (INSA) has been tracking multiple MG events in Portugal, including the BOOM Festival (Mexia, 2016b), the Andanças Festival (Mexia et al., 2017), and the Torres Vedras Carnival (Rodrigues and Mexia, 2015), for epidemiological surveillance. The teams in the field identified the need of a tool to collect and analyze data in real time, which would enable detection of any situation that could put public health at risk during an MG, such as a sudden epidemic outbreak. This demand led to a partnership between Instituto Superior Técnico (IST) and INSA, for the development of a lean clinical information system that could be rapidly deployed at MG events to support professional teams and collect data in real time for epidemiological surveillance.

1.1 Objectives

This dissertation introduces a new clinical information system, called EpiMass, which supports the main requirements initially identified by INSA, and further refined in this thesis:

- Offer an information system for supporting the basic needs of a first aid facility, runnable in tablets and
smartphones, and capable of operating offline.

- Perform real-time data collection for epidemiological surveillance.
- Support multiple events at the same time and store data over time in a common repository.
- Provide tools for epidemiological analysis of the collected data.
- Exchange data with other information sources, such as different health care providers.

1.2 Contributions

A first prototype of the EpiMass platform was built to collect and analyze data for epidemiological surveillance, providing basic information supporting healthcare teams at the MGs. EpiMass has two functional modules available, namely a clinical module for the registration of health occurrences, and an epidemiological module for real-time reporting and preliminary analysis of the data collected.

The clinical module was tested during an MG event and the feedback received by the healthcare teams in the field has contributed to the validation of the approach taken for coping with the need of the epidemiologists at MGs. The EpiMass prototype provides a solution for the demand for an epidemiological surveillance platform identified by health professionals of INSA for, creating a solid, however expandable, tool to this purpose. In the end, by offering a better information of events during MGs, this platform will hopefully help to improve the general public health of the Portuguese population.

1.3 Methodology

The main steps taken to develop EpiMass were the following:

1. Preliminary study of the problem and key features of the approach needed to tackle it, through fieldwork at the Andanças Festival between August 8th and 11th, 2017. I have joined the epidemiological surveillance team of the festival, to understand the real needs of the professional teams at an MG and derive the system requirements for the platform.
2. Literature revision regarding public health surveillance at MGs.
3. Identification of platform requirements by stakeholders through meetings and interviews, namely with INSA, Andanças Festival, Cruz Vermelha Portuguesa, BOOM Festival and Comissão para a Planificação da Resposta em Saúde no contexto de Situações Críticas e de Exceção no Algarve.
4. Definition of the software architecture of the EpiMass platform and selection of the base software to support EpiMass.
5. Development of the EpiMass clinical module taking into account the user requirements.
6. Validation of the robustness and functionality of the clinical module during an MG event, at Torres Vedras Carnival on February 2018.
7. Development and evaluation of the epidemiological analysis module, using historical data from past events.

1.4 Outline of the Dissertation

This dissertation is organized into six chapters. The next chapter provides an introduction to public health surveillance at MGs, and reviews some MG systems found in the literature.

In the third chapter, I describe the considered technologies for web development, along with the main procedures to make web applications more secure and reliable.

Chapter 4 details EpiMass, starting with the requirements, passing through its architecture, and then covering the main features of the clinical module. It also explores the developed data visualization module for epidemiological analysis.

Finally, in the last chapter, I summarize the main accomplishments and contributions of the work, along with ideas for future developments which would likely further improve it.
Chapter 2: Literature Review

This chapter reviews the literature focusing on public health surveillance at Mass Gatherings (MGs). Section 2.1 explains the main concepts underlying an MG event, providing a classification of MGs into several types and some examples of events for each of them. Section 2.2 reviews surveillance systems documented in the literature. Section 2.3 explores epidemiological surveillance at Portuguese MG events. Finally, the state-of-the-art is discussed in Section 2.4.

2.1 Mass Gatherings

The World Health Organization (2015) defines an MG as a "(...) concentration of people at a specific location for a specific purpose over a set period of time and which has the potential to strain the planning and response resources of the country or community”. MGs can be either spontaneous or planned. Spontaneous MGs may include events, such as funerals of celebrities, which are difficult to plan, while planned MGs can be prepared in due time by the health authorities. The WHO classifies planned MGs into four types: sporting events, cultural events, religious events, and political events. Table 2.1 presents some examples of MG events in each type.

The Eurovision Song Contest and the Web Summit are recent examples of large MG events held in Portugal, both taking place in Lisbon and attended by a large number of people. Moreover, in Portugal, frequent large gatherings of people are soccer matches and religious celebrations. While the former may take many people to stadiums\(^1\), the largest of the latter takes place around the Sanctuary of Fatima, as great masses of people get together for the celebrations of the apparitions of Fatima, a significant event of the Catholic Church. Fatima’s official website states that the Sanctuary of Fatima hosted 9.4 million pilgrims during 2017, due to the several Centenary commemorations.\(^2\)

Furthermore, the number of music festivals in Portugal registered a new record in 2016, with an increase of 18% when compared with the previous year. According to the Associação Portuguesa de Festivais de Música (APORFEST), there are 249 organized music festivals in Portugal and the total number of spectators went from 1.8 to 2.1 millions, between 2015 and 2016.\(^3\) Figure 2.1 displays the location of music festivals in Portugal in 2016, according to APORFEST.

The NOS Alive’17 edition sold out the tickets three months before the doors opened. During the three days of the festival 165,000 people attended. Among them 22,000 participants from 80 different countries, as reported by the organization at the official website of the festival.\(^4\)

---


\(^3\)Information available on December 14th, 2017, at http://www.aporfest.pt/single-post/2016/09/22/243-festivais-de-m%C3%BAsica-em-Portugal-no-ano-de-2016-estudo-Aporfest

\(^4\)Information available on December 14th, 2017, at http://www.everythingisnew.pt/noticias/nos-alive17-edicao-
Table 2.1: Examples of MG events according to type.

<table>
<thead>
<tr>
<th>Type</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sporting</td>
<td>Olympic Games, FIFA World Cup, Super Bowl</td>
</tr>
<tr>
<td>Cultural</td>
<td>Fairs and music festivals, such as Glastonbury Festival</td>
</tr>
<tr>
<td>Religious</td>
<td>Pilgrimages, for instance, the Hajj and Kumbh Mela</td>
</tr>
<tr>
<td>Political</td>
<td>Rallies and protests</td>
</tr>
</tbody>
</table>

Llorente et al. (2017) surveyed the literature on MGs published worldwide between 2000 and 2015. 96 articles with relevant information were reviewed, of which 46% were related to sports, 25% to music, and 23% to religious/social content. The Hajj, the Olympic Games, the World Youth Day, and the FIFA World Cup are the four most frequently studied MG events, due to their magnitude and the large number of participants from distinct countries, which translate into a large potential impact, both regionally and globally.

MG events can be characterized by multiple variables. The WHO provides characterizations of MG events, dividing them into MG features, environmental factors, participant characteristics, and venue characteristics. Brunette et al. (2017) state that MGs can be described in terms of their location, venue, purpose, size, participants, duration, timing, activities, and capacity. Milsten et al. (2002) highlight as important variables the weather, event type and duration, age of attendees, crowd mood and density, attendance, and

Figure 2.1: Location of music festivals in Portugal. Image credits belong to APORFEST.
alcohol and drug consumption. All these variables can interact in complex and dynamic ways representing specific challenges for public health officials.

These are many risks that can arise before, during, or after an MG event. Some of them are foreseeable and feasible to manage. However, the unforeseeable ones can place a strain on the local resources. The World Health Organization (2015) points out as main hazards:

- The transmission of infectious disease.
- Injuries and trauma.
- Illnesses due to the use of alcohol and drugs.
- Environmental effects, for instance, dehydration and hypothermia.
- Natural disasters.
- Deliberate acts, such as terrorist attacks.

Furthermore, Brunette et al. (2017) highlights the number of casualties that have occurred as result of poor crowd management, structural collapses, fires, and violence.

Outbreaks have been also reported at MGs. Gautret and Steffen (2016) observe that many of these outbreaks result in the international spread of communicable diseases and that the most common ones have been vaccine-preventable diseases, mainly measles and influenza. On the other hand, common communicable diseases include gastrointestinal infections, resulting from non-compliance with hygiene rules and inadequate sanitation. Botelho-Nevers and Gautret (2013) surveyed various outbreaks related to large scale open air festivals, between 1980 and 2012. Outbreaks of respiratory and gastrointestinal diseases were the most found in the literature. For instance, 3,175 women became ill with gastroenteritis caused by an uncooked tofu salad at an outdoor music festival in Michigan, in 1988 (Lee et al., 1991).

The Hajj, one of the largest MGs, takes approximately three million Muslims from around the world to the annual pilgrimage to Mecca. AlBarrak et al. (2018) performed a study to evaluate the proportion of community-acquired pneumonia (CAP) cases among Hajj pilgrims in 2016. It was concluded that CAP has an important clinical impact during Hajj and mentioned that a proportion of CAP cases were attributable to S. pneumoniae, a pathogen for which vaccines are available. The literature demonstrates that risks must be identified and evaluated, so public health officials can take control and mitigation measures in order to reduce the number of threats and vulnerabilities, as well as its potential impact on public health (World Health Organization, 2015).

2.2 Public Health Surveillance Systems

The Centers for Disease Control and Prevention (2006) state that public health surveillance should be implemented at MGs, since it allows the early detection of outbreaks and other health-related events, enabling public health officials to deploy control measures in due time. Furthermore, according to Nsoesie et al.
MG events present multiple opportunities to improve existing surveillance systems, as well as develop new digital technologies to help the detection and prediction of disease outbreaks. They have reviewed novel approaches to disease surveillance at MGs, including syndromic surveillance systems, mobile phone applications, and telemedicine. Some of these surveillance systems, as well as others found in the literature, are presented below.

During the Athens 2004 Olympic Games, a syndromic surveillance system was implemented to detect syndromes potentially related to an epidemic outbreak or the deliberate use of biological agents (Tsouros and Efstathiou, 2004). The data was manually collected through a specially designed syndromic surveillance form. Data sources included hospitals, primary health care centers, health care clinics at the Olympic venues, and other health care providers. The data was then coded to a standard Microsoft Excel and converted into a format that would allow a statistical analysis by the syndromic surveillance data management team. The system was fully implemented and operational and its effectiveness was evaluated inside and outside the Olympic venues.

Yang (2017) describes a new syndromic surveillance system introduced for the Beijing Olympic Games in 2008: the Beijing Olympic Games Infectious Disease Surveillance System (BOG-IDSS). The system was designed for disease surveillance targeted at notifiable diseases, and syndromic surveillance targeting health-related symptoms. Sporadic and newly imported infectious disease cases were timely controlled and no secondary cases occurred. BOG-IDSS facilitated the early detection of outbreaks or epidemics, contributing to the absence of major outbreaks, epidemics, or significant transmission of communicable diseases, and ensuring the safety of the public health during the Beijing Olympic Games.

Neto et al. (2017) conducted a study to evaluate the use of a participatory surveillance application, called Healthy Cup (“Saúde na Copa”). It was the world’s first application of participatory surveillance for an MG event. The pilot of Healthy Cup was tested during the FIFA World Cup 2014, in Brazil, for the early detection of acute disease outbreaks. After the download of the application, the users were able to record their health condition. The application was available for smartphones and as web app, meaning that it could be accessed through any web browser. It was an innovative way to perform epidemiological surveillance through community engagement, with several advantages, such as lower costs and faster data collection, compared to traditional epidemiological surveillance. The Healthy Cup application gave origin to the Guardians of Health, a participatory surveillance application used during the Olympic Games in Brazil, in 2016.

A mobile-based health information system was developed and deployed at Kumbh Mela, in India (Kazi et al., 2016). Kumbh Mela is the largest congregation of pilgrims on earth, gathering over 70 million people. The system was using a tablet-based data collection interface customized for disease surveillance, allowing real-time data analysis. It also had offline synchronization capabilities. A total of 49,131 episodes of illness were registered and the most common recorded complaints were musculoskeletal pain, fever, cough, coryza, and diarrhoea. This study provided important insights regarding the use of mobile technology for
public health surveillance at MGs.

2.3 Epidemiological Surveillance in Portugal

Portugal hosted the European football cup (EURO 2004), between June 12th and July 4th, 2004. There were a total of 2,003 patients during the 31 matches, from a total of 1,165,192 spectators (Soares-Oliveira, 2015). Portugal’s Northern Regional Health Authority hosted 12 matches during EURO 2004, in which ten foodborne outbreaks, seven cases of meningococcal disease and one case of legionnaires disease were detected, demonstrating the importance of surveillance activities during MG events (Gonçalves et al., 2005).

Cordeiro et al. (2013) implemented a syndromic surveillance system at the BOOM Festival in 2010, to provide early detection of disease, threats or syndromes which require immediate control measures by public health officials. A total of 2,287 episodes of illness were reported, mostly due to traumatic complaints. A total of 126 episodes of gastrointestinal symptoms and 111 respiratory symptoms were reported. The authors recommended to use the surveillance system in the further editions of the festival, and eventually adapt it to other MG events.

Rodrigues and Mexia (2015) reported on the epidemiological surveillance performed during Torres Vedras Carnival in 2015, an annual MG event with about 350,000 participants that takes place in an urban context. There were two identified cases with epidemiological potential during the event. The authors observed that control measures were taken to avoid its spread and emphasized the importance of surveillance systems at MGs. It was also mentioned that the system should be improved, since the data collection through paper forms, could not allow real-time surveillance.

A new approach for epidemiological surveillance was developed and later used at the 2016 Andanças Festival (Mexia, 2016a). The developed system provides an electronic record to collect all health episodes. Most reported episodes were small trauma and no major health problems concerning public health were detected. Mexia observed that epidemiological surveillance systems provide an adequate tool to monitor health status at MG events and enable early detection of outbreaks. It was also pointed that it would be very helpful to have a ready to deploy epidemiological surveillance system.

In sum, the Department of Epidemiology of INSA has been present at many different MG events in Portugal for epidemiological surveillance, such as the BOOM Festival (Mexia, 2016b), Andanças Festival (Mexia et al., 2017), and Torres Vedras Carnival (Rodrigues and Mexia, 2015). The experience accumulated has allowed INSA to develop, implement and, at the same time, improve their own solutions at MGs. Currently, INSA is using the approach developed by Mexia (2016a) for epidemiological surveillance. The electronic record provided by the system is based on REDCap\(^5\). REDCap is a web application to manage surveys online in a secure way. I used this system at the 2017 Andanças Festival and I realized that it is not possible to analyze data in real-time since the system only collects the data to a database, and it does not provide

\(^5\)https://www.project-redcap.org/
any interface or API for data analysis.

The expertise on epidemiological surveillance at MGs in Portugal is limited, and there are no real-time surveillance systems available at most Portuguese MG events (Mexia, 2016a). I was present at a meeting about MGs at Instituto Português do Sangue e da Transfusão on February 21st. A new regulation related to MG events is being prepared, and epidemiological surveillance will be mandatory at all Portuguese MGs, which will increase the need for real-time surveillance systems at MG events.

2.4 Summary

MGs are characterized by a large number of people that come together in a particular location for a specific purpose. MG events have multiple variables with many inherent risks for public health. The health authorities of the host countries should identify the main risks and evaluate them, as well as potential impacts. A sudden epidemic outbreak is a threat always present at an MG event, as result of the increased infectious disease transmission due to the influx of attendees, and the poor hygiene from temporary food and sanitation facilities (Joseph S. Lombardo et al., 2008).

Public health surveillance systems are fundamental to identify an outbreak as soon as possible, so public health officials can act promptly. New digital technologies and approaches have been developed and implemented at large MGs and can be found in the literature. The BOOM Festival, the Andanças Festival, and the Torres Vedras Carnival are examples of Portuguese MG events where INSA has been doing epidemiological surveillance, ensuring the safety of the public health at these events.

However, the majority of the Portuguese MG events do not have a standardized way to collect data, and only a few actually perform epidemiological surveillance. For instance, at the Torres Vedras Carnival, the data is collected through paper forms, and only then duplicated to a digital format using Microsoft Office Excel. On the other hand, the teams at the Andanças Festival use the approach developed by Mexia (2016a). Still, none of these methodologies use real-time data analysis for epidemiological surveillance.

The new regulation will increase the need for real-time data collection, so epidemiological surveillance can be done in due time to identify and eliminate threats for public health. So, there is a need to develop a clinical information system that can be rapidly deployed in the field. This way, data can be easily collected during MG events, and then analyzed by epidemiologists. EpiMass was developed in this context, since it is a simple clinical and epidemiological platform, able to answer the basic needs of healthcare providers during MGs.
Chapter 3: Technologies for Web Development

This chapter covers the technologies used to develop EpiMass. Section 3.1 presents the state-of-the-art 3-tier architecture. It also explores the most recent frameworks and top programming languages used to develop web platforms. Section 3.2 addresses the concerns with security and authentication that must be taken into account when dealing with personal and clinical data. Finally, an overview of the chapter is available in Section 3.3.

3.1 Web Platforms

Web platforms are used worldwide everyday. A web platform is a standard way to build and deploy web applications. Facebook is a good example of an application implemented over a web platform. It is one of the biggest web platforms with 1.4 billion daily active users on average, according to data for December 2017, and 2.13 billion monthly active users as of December 31, 2017\(^1\). Web platforms can serve several purposes, including health. The number of web applications and web services related to health is increasing, at least in Portugal, according to what I could experience at Portugal eHealth Summit 2018. There were about 45 startups promoting their products\(^2\), which were mostly based on web platforms. The major advantage of a web platform is that it just needs a web browser to be accessed, whether you are on a computer, tablet or mobile phone, independently of the user’s operating system.

3.1.1 3-tier Architecture

Currently, many web platforms are based on a 3-tier architecture, which means that they are composed of three tiers, namely the presentation tier (client), the application tier (server), and the data tier (database), as illustrated in Figure 3.1. In a 3-tier architecture, the communication between tiers is made through API calls. An API (Application Programming Interface) is a set of clearly defined methods that allow the communication between the client, server, and database in a web application.

The presentation tier is the front-end and consists of the user interface, which is accessible through a web browser, or a specialized mobile application, and communicates with the application tier through the API calls defined by the web application. The application tier is responsible for the functional logic of the web application. The data tier contains the database management system, which is accessed by the application tier via API calls.

---

\(^1\)Information available on April 10th, 2018, at https://newsroom.fb.com/company-info/
\(^2\)Information available on April 10th, 2018, at http://ehealthsummit.pt/startups/
3.1.2 REST APIs

REST APIs were designed to take advantage of existing protocols, more precisely the HTTP protocol, for remote-procedure calls. Fielding and Taylor (2002) describe REST as an architectural style for distributed hypermedia systems. It is the most popular type of API.

REST APIs use HTTP methods to GET, PUT, POST, and DELETE contents. They use GET to retrieve an object, PUT to update an object, POST to create that object, and DELETE to remove it.

3.1.3 MEAN Stack

There are several sets of software components that can be combined (stacked) to implement a full web platform supporting a custom web application, such as the MEAN stack or the LAMP stack. Currently, the MEAN stack is replacing LAMP (Louridas, 2016) as the most popular choice. While the LAMP stack uses Linux as operating system, Apache as web server, MySQL as relational database, and PHP as scripting language (Lawton, 2005), the MEAN stack is a free and open-source full-stack JavaScript framework for building dynamic websites and web applications (Holmes, 2015). MEAN is an acronym for MongoDB, Express, Angular, and Node.js. Succinctly, MongoDB is a non-relational database management system, and Express is a web application framework for Node.js, designed for building web applications and APIs. Angular is a front-end (client-side programming) web application framework, and Node.js is a JavaScript runtime environment that executes JavaScript code on the server-side. MEAN is easier and simpler to use than LAMP, since it allows to develop a full-stack web application in just one language, JavaScript, instead of switching between languages on the client and server-sides.

Figure 3.2 illustrates the architecture of a application based on the MEAN stack, which is based on a 3-tier architecture. Shortly, when the client makes a request, it is firstly processed by Angular and only then parsed by Node.js. After that, Express makes the request to the database, MongoDB retrieves the data and returns the request back to Express. Express returns the response to Node.js and then it is returned to
Angular to display the result.

There is also .NET Core, a free and open-source web framework developed by Microsoft. However, a common example seen in .NET Core is to code the server in C#, and the client in JavaScript, using Angular, which implies the use of two different programming languages once more. Moreover, JavaScript is now the most popular programming language according to Stack Overflow’s Developer Survey Results of 2017, and its popularity has been increasing since 2013. Node.js and Angular are also the two most popular frameworks, according to the results, and MongoDB is the most popular non-relational database.

A more in-depth explanation of each component of the MEAN stack, namely Angular, Node.js, Express, and MongoDB, is given below.

**Angular**

Although the MEAN Stack is associated to AngularJS, I used the most recent version of a related software package, Angular, to develop EpiMass. Angular is an open-source TypeScript-based (a strict syntactical superset of JavaScript) framework created by Google from scratch for building web applications (Eschweiler, 2016). Angular has been totally rewritten and it is not compatible with AngularJS.

Angular provides a command-line interface (CLI) to generate new components, routes, services and pipes. Furthermore, it allows one to easily test the application locally while developing. Alternatively to Angular or AngularJS, the front-end could also be developed with React, a JavaScript library for building user interfaces, developed by Facebook and Instagram, or with Vue.js, an open-source JavaScript framework.

---

3Information available on April 10th, 2018, at https://insights.stackoverflow.com/survey/2017
4https://angularjs.org/
5https://www.typescriptlang.org/
6https://reactjs.org/
7https://vuejs.org/
for building user interfaces. Even so, I went for Angular since it is widely used in production and because it is the most popular front-end framework.

**Node.js**

Node.js is a JavaScript runtime environment for web servers built on Chrome's V8 engine (Wilson, 2018). It is an open-source JavaScript engine written in C++ that takes JavaScript code and compiles it rapidly to machine code. Node.js uses an event-driven, non-blocking I/O model that allows asynchronous execution of functions. This means that the server never waits for an API to return data. It moves to the next API after calling it and the Events module of Node.js helps the server to get a response from the previous API call. Lei et al. (2014) demonstrated that Node.js is lightweight and efficient comparatively to other programming languages that use blocking I/O.

The Node.js event-based engine is highly extensible through a package manager, named npm. It is the largest ecosystem of open source libraries in the world, with more than 650,000 total packages, 640,000,000 registry downloads per day, 4.5 billion registry downloads per week, and 17.7 billion registry downloads per month. Further, npm registry is a command-line client that allows developers to install, update and uninstall packages with extreme ease.

Node.js is widely used in production. Many high-profile companies, such as Netflix, Linkedin, Uber, and PayPal are using Node.js for building their primary application.

**Express**

Express is a minimalist and well documented Node.js framework designed for building web applications and APIs (Hahn, 2016). Express is also a routing and middleware web framework that allows defining how the server handles requests. Furthermore, it has several methods to design REST APIs. Together with Node.js, they make it possible to build an entire website in JavaScript. It is one of the most packages installed via npm registry.

**MongoDB**

MongoDB is a free and open-source cross-platform document-oriented database, classified as a NoSQL database program (Banker et al., 2016). Unlike SQL databases, where data is stored in structured tables, in MongoDB, data is organized as collections of documents represented as JavaScript data structures (see Figure 3.3). These data structures are seen as documents in a lightweight data-interchange format, usually structured as an object written in key/value pairs, called JSON (JavaScript Object Notation).

---

8Information available on April 11th, 2018, at https://www.npmjs.com/enterprise
9Information available on April 11th, 2018, at http://www.tothenew.com/blog/how-are-10-global-companies-using-node-js-in-production/
10https://www.json.org/
Figure 3.3: Comparison between a table with two rows, on the left, and a collection of two JSON-like documents, on the right.

MongoDB’s collections do not enforce document structure, giving MongoDB a flexible schema. This flexibility is an advantage over relational databases since it allows to easily scale a database without redoing its schema. Further, this makes data integration faster and easier due to the flexibility to integrate with API methods.

Every time a document is created, if there is not a unique ID specified, MongoDB automatically generates a unique ObjectID for each document, making it unequivocally identifiable. All these advantages make MongoDB the first choice for many big companies, like Facebook, Cisco, eBay, and Medtronic, a medical device company.11

3.1.4 Offline Operation

Web applications may become useless when facing low network connectivity issues. Also, when offline, most web applications lose their main functionalities. There are strategies to circumvent this issues, implementing methods that allow web applications to still have some functionalities in these situations. Usually, these strategies consist in transferring the computation from the online cloud server to the local machine. This way, the application can still be partially running even in environments of offline mode or low network signal. Then, when back online, the data would be synchronized with the central server, and so, integrated and distributed by all the other parties using it (Ozzie et al., 2015).

PouchDB12 is an open-source JavaScript library to build web applications that allows the applications to store data locally while offline. Then, when the device is back online, the data is synchronized with CouchDB13, a database software. However, this is still very recent technology with many incompatibilities. In fact, it is not possible to use PouchDB with MongoDB, since CouchDB was designed to be used for synchronization, while MongoDB was not.

Alternatively, service workers can be used. Introduced by Google, a service worker is a script that runs

11Information available on April 11th, 2018, at https://www.mongodb.com/who-uses-mongodb
12Information available on May 4th, 2018, at https://pouchdb.com/
13https://couchdb.apache.org/
in the browser in the background\textsuperscript{14}. Briefly, when the user accesses the application for the first time, while being online, the service worker is registered and some methods and parts of the application are cached. Then, when the user faces low network connectivity issues or becomes offline, the application is partially loaded from the browser cache. Currently, there are already features like push notifications and background synchronization available. However, to deploy a service worker it is mandatory to have HTTPS on the server.

3.2 Security and Authentication of Communications

Security and authentication are fundamental to assure that a web application complies with data privacy requirements. It is mandatory to authenticate both the client and the server when transmitting data, to avoid any kind of security breaches from third parties. The ‘Basic’ HTTP Authentication Scheme\textsuperscript{15} is not secure, since the user ID and password are passed over the network, encoded using Base64. However, Base64 is a reversible encoding.

SSL stands for Secure Sockets Layer and, in short, it is a security protocol widely used to provide secure communication on the Internet. It uses encryption algorithms to establish a secure connection between two systems, a client and a server, preventing anyone from reading or modifying any information when transmitting personal or sensitive information, such as personal details, credit card numbers, clinical information, and login credentials. HTTP, the standard client-server protocol of the web, has a version that runs over SSL called HTTPS.

3.2.1 Server Authentication

In the case of a web platform, a certificate is installed on the server-side in order to authenticate the server. The browser connects to the server and verifies the certificate. If it trusts the certificate, the browser and the server use the key from the certificate to start an encrypted session. All transmitted data is encrypted with a session key via HTTPS. Every SSL certificate is digitally signed by a trusted Certified Authority (CA) and a browser only trust certificates that come from an organization on their list of trusted CAs, like DigiCert\textsuperscript{16}. EpiMass has an SSL certificate signed by DigiCert.

There are visual cues on the browser that can tell users that an SSL certificate is installed on the server, namely a green padlock icon at the address bar. Notice also that the URL for secure web applications begins with https:// rather than http://.

\textsuperscript{14}Information available on May 4th, 2018, at https://developers.google.com/web/fundamentals/primers/service-workers/
\textsuperscript{15}Information available on April 17th, 2018, at https://tools.ietf.org/html/rfc7617
\textsuperscript{16}Information available on April 17th, 2018, at https://www.digicert.com/ssl/
3.2.2 Client Authentication

To make sure that a user is, in fact, who he claims to be, authentication can be provided through a username and password, which is simpler than requiring a digital certificate.

A JSON Web Token (JWT) is an open standard (RFC 7519) that defines a compact and safe way to represent a set of information between two parties as a JSON object. A JWT consists of a string composed by a header, a payload, and a signature, separated by dots (header.payload.signature). It is commonly used for authentication and information exchange, as illustrated in Figure 3.4.

Figure 3.4 illustrates how user authentication and information exchange work with a JWT. First, the user logs in through the login system using his credentials. When the user successfully logs in, the authentication server creates a JWT with a secret and sends it back to the user. The JWT must be saved locally, typically in local storage.

After the login has been validated, every time the user makes an API call, the JWT is passed through the authorization header along with the API call. The server verifies the JWT signature, processes the API call, and sends the response back to the user. This way, it is not necessary to exchange the password for every request made to validate its authenticity.

---

18 Information available on April 17th, 2018, at https://jwt.io/introduction/
19 Information available on April 18th, 2018, at https://auth0.com/docs/security/store-tokens
3.3 Overview

EpiMass is based on the MEAN stack, one of the most recent technologies for building web applications based on web platforms. The MEAN stack is a free and open-source JavaScript software stack that allows developing an entire web application in a simpler and easier way when compared with other alternatives. Angular is used on the front-end of a web application, while Node.js and Express are used to setup the server. Furthermore, REST allows to use JSON to represent an object, making it easier to integrate MongoDB with Node.js. These components are being used by many high-profile companies, since they are open-source and do not require purchasing license to be used.

HTTPS, or HTTP over SSL, is the standard protocol for encrypted web communication and is supported by all browsers and web servers. It uses encryption algorithms for preventing anyone from reading or modifying any information when transmitting confidential data. To support authentication, a JWT is used to exchange session information for user and API requests authentication. Both SSL security protocol and JWTs ensure trust and security to a web application, including EpiMass.

Regarding data security, the database of EpiMass is hosted at the Infraestrutura Nacional de Computação Distribuída (INCD)

20. Currently, the data is not encrypted. However, the MongoDB Enterprise, which is not free, includes advanced security features that allows MongoDB to encrypt data files, so only parties with the decryption key can decode and read the data. Finally, EpiMass was developed before the General Data Protection Regulation (GDPR) entered into application. Therefore, the new regulation must be taken into account in future developments, in order to confirm if the platform complies with all the guidelines.

20http://www.incd.pt/
Chapter 4: EpiMass

EpiMass is a clinical and epidemiological web platform designed for supporting the basic needs of a first aid post at Mass Gatherings (MGs). This chapter describes the steps involved in developing EpiMass, and the main functionalities available in the platform.

Section 4.1 presents the requirements gathered for conceiving EpiMass, including my experience at the Andanças Festival. Section 4.2 details the platform architecture, as well as the database structure. Section 4.4 covers the different user dashboards available, along with screenshots illustrating the functionalities of the platform. Section 4.5 presents the results of the assessment tests that were performed to the clinical module during the Torres Vedras Carnival, as well as the features of the epidemiological analysis module. Finally, Section 4.6 contains an overview of the chapter.

4.1 Requirements

Dr. Ricardo Mexia, an expert on epidemiological surveillance from Instituto Nacional de Saúde Doutor Ricardo Jorge (INSA), provided an initial list of requirements identified during past MG events in Portugal. I have also joined the epidemiological surveillance team of the 2017 Andanças Festival, a dance and music festival held in Castelo de Vide, Portugal. It is an urban MG of small dimension, when compared to other music festivals (for instance, the NOS Alive). Thus, it was a great opportunity to perceive the real needs of the professional teams present at the first aid post of the event, and derive in first hand the system requirements for the platform.

Figure 4.1 pictures the epidemiological surveillance team of the 2017 Andanças Festival. During the festival, I could interact with the approach/system developed by Mexia (2016a). This system consists on a web form to collect data at an MG, supported by REDCap1, as shown in Figure 4.2. I tested this solution and its behavior is much similar to a Google Form. Therefore, if a patient returns to the first aid post, all the fields regarding its personal data must be filled in again, in order to register the health occurrence. Also, this approach does not allow directly support epidemiological surveillance in real-time. It is necessary to interact with the database to get an insight of the status of the health occurrences.

1https://www.project-redcap.org/
The requirements for EpiMass were collected and discussed in further meetings and interviews with stakeholders, including INSA, Andanças Festival, Cruz Vermelha Portuguesa, BOOM Festival, and the Comissão para a Planificação da Resposta em Saúde no contexto de Situações Críticas e de Exceção no Algarve².

The main requirements for EpiMass are:

- Offer an information system for supporting the basic needs of a first aid post, including registration of all health occurrences. Many first aid posts are kept in temporary infrastructures with low resources,

and without a clinical information system to support the data collection. An information system to easily collect data during MGs would contribute to an improvement of public health in general.

- Present a lightweight web application runnable in tablets and smartphones. The professional teams do not always have the resources, like laptops and custom software at the first aid post to manage incidents and collect data. The platform must be capable to run in any device, so tablets and smartphones can be used.
- Collect data in real-time for epidemiological surveillance, allowing the early detection of outbreaks and avoiding spreading and contamination. Otherwise, if an outbreak is not detected in time, it can place a strain on the local emergency response mechanisms.
- Support multiple events at the same time and store data over time in a common repository. This way, a posteriori analysis can be done to compare data from different editions of the same event, or even for the evaluation of the impact of public health programs.
- Establish different user categories for different roles and accesses to the platform. Each user role has a set of functionalities available, according to its permissions. Furthermore, each user should only have access to one event at each time, and the administrators should not have access to clinical data. This increases data safety and allows a better control over secure information.
- Provide statistical information about each event, such as the number of patients treated and the patients which are on the waiting list, providing an overview of the health occurrence status to the event producers.
- Enable tools for epidemiological analysis of the collected data, with bar charts facilitating the creation of reports as well as the gathering of overviews about the data.
- Ability to collect data while operating offline (disconnected from the network), but keeping the application synchronizable with the central server for posterior analysis. The network connectivity may face problems during MG events, so it is important to maintain some functionalities available, such as the collection of data.
- Exchange data with other information sources, such as health care providers. For example, an option for clinicians to get access to patients allergies may contribute to a better diagnosis and treatment. Furthermore, if a patient needs to be transferred to a hospital, the collected data could be later integrated into the patient’s EHR.

### 4.2 EpiMass Architecture

EpiMass follows a 3-tier architecture, namely by separating the user interface, the web server, and the database, as already explored in Chapter 3.

The EpiMass platform offers several functionalities which are available according to each of the four supported user roles:
Table 4.1: Functionalities according to each user role.

<table>
<thead>
<tr>
<th>Functionality</th>
<th>Administrator</th>
<th>Clinical</th>
<th>Epidemiologist</th>
<th>Producer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create New Events</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Register New Users</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Register Patients</td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Search for Patients</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Manage the Waiting List</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Register Health Occurrences</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Perform Epidemiological Analysis</td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>View Health Occurrences</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
</tr>
</tbody>
</table>

1. **Administrator**, which is only for administrative purposes. The administrator is the user role responsible for managing users and events. This user does not have access to clinical data.

2. **Clinical**, which is the role assigned to the professionals at the first aid posts. Its main functionality is to register health occurrences during an MG event. Only this user role has access to patient data.

3. **Epidemiologist**, which is the only user with privileges to access the tools available for epidemiological analysis of the collected data.

4. **Producer**, which is the user role designated to the producer of the event, so this user can have an insight of the health occurrences status.

Table 4.1 presents a summary of the functionalities available for each user in each available role.

EpiMass is based on the MEAN stack (Holmes, 2015), an open-source JavaScript framework for developing web applications.

The user interface, application server, and database of EpiMass are presented below.

### 4.2.1 User Interface

I developed several components to build the user interface. These components use the Angular Material Design\(^3\) developed by Google, which is common in Google products, conveying a familiar experience to the user. Figure 4.3 presents the EpiMass homepage.

\(^3\)https://material.angular.io/
I also used the font-awesome package, i.e., a suite of pictographic icons, to provide a user-friendly look on the platform. Furthermore, I implemented toast notifications to notify the user that a given action was or not completed with success. The notification toasts appear on the bottom right corner of the browser.

In order to take advantage of the functionalities of EpiMass, the users must login first. The user login functionality is available to all users on the top right corner of the homepage. After clicking on the Login button, the login form is presented to the user.

Figure 4.4 illustrates this login page. There is no option to register an account since only administrators are able to do that. It was defined when the user roles were established.

To login, the user must provide his email and password. In fact, these credentials should have been sent to the user previously by an administrator of the platform. Then, the user gets a notification toast after submitting his credentials. An example of a success notification toast is represented in Figure 4.5.

After the login, the users can view their personal profile, independently of their role, on the top right corner of the navigation bar. This profile can be accessed by clicking on the Profile button that is displayed. Then, the user profile is presented, for now only including the user name and user role, but that can be extended in the future (e.g., with statistical information on user activities).

Figure 4.6 illustrates the profile page of an user of the platform.
Figure 4.4: Login page.

Figure 4.5: Example of a success notification toast.

Figure 4.6: Personal profile page.
4.2.2 Application Server

I developed a set of differentiated API methods for both administrative and clinical purposes, including functionalities to register new users, register patients, get events, get data from the registered health occurrences, and update these occurrences. There are more methods available for retrieving data and also for user authentication. Thus, an authorization method was implemented to handle the API requests on the server-side.

Moreover, the different APIs make data integration faster and easier, as they provide a flexible way to extract and manipulate data, and allowing the user to do so without having to understand the implementation behind the APIs, thus being able to use it in a blackbox manner. These methods are called through services every time a user makes a request through the platform. Then, the web server processes the requests and displays a response to the user on his browser.

EpiMass has two main modules available, namely the clinical module, and the epidemiological analysis module. The functionalities presented in these modules are controlled by the application server. The clinical module is available only for users in the clinical role. This module allows users to register health occurrences on the platform and is composed by a set of functionalities, namely:

1. Register a new patient.
2. Search for a patient.
3. Add a patient to the waiting list.
4. Register a new occurrence.

The epidemiological analysis module is only available for the epidemiologist role. This module has three bar charts available. The first one consists in the distribution of the health occurrences registered by diagnostic groups, allowing a preliminary epidemiological surveillance in real-time. The second bar chart shows the occurrences distribution by age group, while the last one presents the number of occurrences according to the destiny of the patient after going to the first aid post.

I used Node.js for building the application server. I also used some packages publicly available at the npm repository to implement the desired features. The most important packages include:

- **express**, a web application framework designed for building web applications and APIs.
- **body-parser**, a body parsing middleware.
- **http** and **https** to handle HTTP/HTTPS requests.
- **file-system** to read the files from the SSL certificate.
- **passport** to authenticate requests.
- **passport-jwt**, a passport strategy for authenticating with a JSON Web Token.
- **jsonwebtoken**, an implementation of JSON Web Tokens.
- **bcryptjs** to hash the user passwords, with salt.
- **mongoose**, a MongoDB object modeling tool.
4.2.3 Database

The database is composed of JSON-like documents organized into three collections, namely:

1. **events**, which includes the created events. Each event has its own characteristics, namely the name, location, beginning date, and ending date. A unique identifier, called slug, is automatically generated, as shown below:

   ```json
   {
   "_id" : ObjectId("5b1eea57e4b680310b9cbabe"),
   "name" : "Santos Populares",
   "location" : "Lisboa",
   "beginDate" : "01/06/2018",
   "endDate" : "30/06/2018",
   "slug" : "santospopulares01/06/2018lisboa"
   }
   ``

2. **users**, which contains the registered users. This way, the user details are kept separated, being only accessible for administrative purposes. Each user has a name, email, password, role, and event. The field event is not present on users with the role admin since they do not have access to clinical data. An example of a clinical user is presented below:

   ```json
   {
   "_id" : ObjectId("5b1eeb05e4b680310b9cbabf"),
   "name" : "John Smith",
   "email" : "johnsmith@example.com",
   "password" : "2jEXLtR0uNYOV37na8HSDoTRMGQySKEhcxvfl4mf9h6",
   "role" : "clinical",
   "event" : "santospopulares01/06/2018lisboa"
   }
   ``

3. **patients**, which consists of the collected health occurrences. Each patient has an event associated, the arrival moment to the first aid post, its personal data and the occurrence data. Moreover, each document has the patient destiny after going to the first aid post, and the exit moment. This collection was structured in order to easily allow API calls. A document from the patients collection is presented below, with its several fields:

   ```json
   {
   "_id" : ObjectId("5b1fcf6a0f2b204e2c1b7861"),
   "arrivalMoment" : {
   "date" : {
   "day" : 13,
   "month" : 6,
   "year" : 2018
   },
   "time" : {
   "hours" : 2,
   "minutes" : 49,
   "seconds" : 29
   }
   }
   }
The database was built using the well known MongoDB, a NoSQL document-oriented database, due to its flexibility to integrate with API methods.

4.3 Dashboards

Dashboards were developed for each user role, providing different functionalities to the users. The access control is made through Router Guards on the client-side. The four available dashboards are presented below.
4.3.1 Administration Dashboard

The user interface of the dashboard changes dynamically according to the user role. The administrator dashboard, as the name suggests, is only available for the users with the role of administrator of the platform. Thus, after the login, the dashboard is presented to this user.

Figure 4.7 illustrates the administrator dashboard. The dashboard has two main functionalities, namely to create a new event, on the left, and to register a new user, on the right.

Create a New Event

To create a new event, the administrator fills the event name, location, beginning and ending date. A unique identifier for the event will be generated after the user clicks on the Create button. This identifier is the slug field in the events collection (see Section 4.2.3).

Register a New User

To register a new user, the administrator must fill the name, email, password, user type, and the event that this new user will be associated to. The event option will not be available if the user type administrator is selected, since the administrators only have permissions to do user management.

Figure 4.8 pictures the registry of a new user. A toast notification will also show up if the user does not fill in all the fields. Once again, this functionality is only available for administrators.
4.3.2 Clinical Dashboard

The clinical dashboard was designed according to the above mentioned list of requirements list. Firstly, the name of the user is presented along with the event where he is participating. Then, a panel with the number of patients already treated is highlighted, in order to keep track of the state of health occurrences at the event. This panel allows one to get an insight, in real-time, of how many patients were discharged and how many were transferred to another health care provider, such as to a hospital or to a primary care center. There is also a panel with the patients which are on the waiting list. It is presented on the right side of the dashboard allowing easy access to this important information.

Figure 4.9 illustrates the clinical dashboard with all the features described above.
Register a New Patient

Before submitting a new health occurrence in the platform, it is necessary to fill in the personal data of this patient. Thus, after filling the fields, the user may move on to the registration of the occurrence data, or put the patient in the waiting list.

Figure 4.10 shows the registry of a new patient, which consists on inputting the personal data of the patient. Here, the name and gender are the only mandatory fields to proceed, as these are key identifying parameters of each individual entry, required for searching the list.
Figure 4.10: Functionality of registering a new patient.

Search for a Patient

In case a patient has already been registered on the platform during the event, there is the option to search for him. This allows proceeding to the registration of the health occurrence directly, skipping the re-insertion of his personal details.

Figure 4.11 illustrates this functionality. The search is made through the name of the patient. The age and birth date are also presented, to identify and distinguish patients with the same name.
Add a Patient to the Waiting List

This functionality is useful when there are many patients attending the first aid post at the same time. After filling in the personal details, the user can add the patient to the waiting list, while he waits for assistance. The waiting list is present in the clinical dashboard, as shown in Figure 4.9.

Register a New Occurrence

After the registration of the personal data of the patient, it is time to proceed to the registration of the occurrence data. Each occurrence is a new document in the patients collection.

Figure 4.12 shows the form presented on the platform. The form was adapted from the one that has been previously used by INSA at MG events. It has been reviewed and improved over the years with the experience gathered at many MGs. Furthermore, I also included some fields of the form used by Cruz Vermelha Portuguesa at the Torres Vedras Carnival.
Figure 4.12: New occurrence form.
4.3.3 Epidemiologist Dashboard

The epidemiologist dashboard is only available for the users with the epidemiologist role. It is similar to the clinical dashboard, but instead of registering health occurrences, the user can visualize the data registered through bar charts developed for epidemiological analysis. Figure 4.13 illustrates the epidemiologist dashboard.

4.3.4 Producer Dashboard

Finally, the producer dashboard is composed only for the first panel of the clinical and epidemiologist dashboards. This panel allows the event producer to get an insight, in real-time, of the status of health occurrences in the first aid posts of the event. Figure 4.14 illustrates the epidemiologist dashboard.
4.4 Evaluation at the Torres Vedras Carnival

I integrated the epidemiological surveillance team of the 2018 Torres Vedras Carnival. At the event I had the opportunity to perform assessment tests of the EpiMass clinical module, as well as to get information on the field from the actual potential users of the platform. Figure 4.15 pictures the Cruz Vermelha Portuguesa team, and the epidemiological surveillance team of the 2018 Torres Vedras Carnival.

In this event, the platform was employed at two first aid posts at the same time, which shared information on the platform. All the health occurrences were registered with success. The registration of the occurrences was kept in duplicated, since the platform was only used for testing purposes. The platform was successfully tested online and locally (i.e., without an internet connection). Figure 4.16 illustrates me performing tests to the EpiMass platform at one of the first aid posts during the event. The teams at the field were satisfied with the result, particularly given that the form to collect data is very similar to the one they were using (see Figure 4.17).

The Cruz Vermelha suggested to add a panel to the dashboard with the number of health occurrences registered, how many patients were released, and how many had to be transferred. This functionality was contemplated in the requirements list and it is currently present at the EpiMass platform (see Figure 4.9).

After the fieldwork at the Torres Vedras Carnival, I started to develop the epidemiological analysis module of the platform. This module is only available for users with the epidemiologist role on EpiMass. It contains bar charts of the data collected, according to the diagnostic group, age group, and destiny of the patients after going to the first aid post. The charts were developed using Chart.js, an open-source library based on...
Figure 4.15: Teams presented at the first aid posts of the 2018 Torres Vedras Carnival. Source: INSA.

HTML5 and JavaScript for producing charts⁴. These charts are based on the data presented in the daily report of epidemiological surveillance of the Torres Vedras Carnival, and on the reports available on the INSA’s repository from other events.

Figure 4.18 illustrates the distribution of the health occurrences by diagnostic group, during the 2018 Torres Vedras Carnival. In a total of 249 health occurrences, the two most reported cases were due to acute alcohol intoxication, and traumatic injuries, with 119 and 72 cases, respectively. There were no outbreaks associated to this event.

Figure 4.19 illustrates the distribution of the health occurrences by age group of the participants, during the event. I also implemented the possibility to filter the health occurrences by gender. Figure 4.20 presents the same data, however a filter was applied to show only for men cases.

The age groups that were associated to more occurrences correspond to patients between 15 and 29 years of age, representing 80% of the cases. Regarding the gender, the incidence was about the same, namely 50% for both men and women.

Figure 4.21 shows the cases by the destiny of the patients after going to the first aid post. This chart shows that the patients had medical release in 109 occurrences, while in the other 140 occurrences registered, the patients were transferred to the Torres Vedras Hospital to receive special medical care.

---

⁴https://www.chartjs.org/
4.5 Overview

Software implementing a first prototype of the EpiMass platform has been developed in the context of my MSc thesis. Both the clinical and epidemiological modules demonstrated to be functional, and the feedback obtained was very positive. The platform is able to collect data in real-time, and provides tools for a preliminary epidemiological analysis during MGs. Furthermore, the performance of the clinical module was tested at the 2018 Tores Vedras Carnival, where there were reported about 250 health occurrences. There were no outbreaks during this specific MG event.

EpiMass is available at https://epimass.inesc-id.pt/ and can be accessed through any device with a web browser and network connectivity. The domain was provided by the Instituto de Engenharia de Sistemas e Computadores, Investigação e Desenvolvimento em Lisboa (INESC-ID)\(^5\). The central server and the database are hosted at the Infraestrutura Nacional de Computação Distribuída (INCD)\(^6\).

\(^5\)http://www.inesc-id.pt/
\(^6\)http://www.incd.pt/
**SISTEMA DE VIGILÂNCIA EPIDEMIOLÓGICO CARNAVAL TORRES VEDRAS 2018**

**FICHA DE REGISTO DE OBSERVAÇÃO E ENCAMINHAMENTO**

<table>
<thead>
<tr>
<th>Colar Vinheta Aqui</th>
</tr>
</thead>
</table>

- Está em Torres Vedras por causa do Carnaval? □ Não □ Sim □ Residente Concelho
- Os sintomas ocorreram no contexto de festejos do Carnaval? □ Não □ Sim
- Se Sim, onde? □ Recinto □ Domicílio □ Restaurante □ Bar □ Discoteca □ Rua (mas fora do recinto)

(a preencher pela Equipa de Intermagem na Triagem das Urgências)

<table>
<thead>
<tr>
<th>Registo de Ocorrência Nº</th>
</tr>
</thead>
</table>

- Data de Nascimento ___/___/____ ou idade em anos ____ ou idade em meses (se < 1 ano) ___
- Sexo M □ F □ Nacionalidade ____________________________ Telemóvel: ____________________________

1 - Data de início dos sintomas ___/___/2018

2 - Motivo do recurso aos cuidados de saúde (células em branco são resposta negativas)

- □ Febre
- □ Sinais ou sintomas respiratórios (tosse, pierna, dispneia)
- □ Vômitos
- □ Diarreia
- □ Tonturas
- □ Dor, local ____________________________
- □ Traumatismo por: ____________________________
- □ Sinais ou sintomas genito-urinários, quais ____________________________
- □ Outros, quais: ____________________________

3 - História Actual:

___________________________________________________________________________________

4 - Sinais vitais (se relevante): FC ___ TA ___/___ mm Hg T ___° SpO₂ ___ FR ___

5 - Hipótese de diagnóstico (a preencher por médicos)

6 - Casos relacionados? □ Não □ Sim Se sim, quantos? __ □ (listar nomes e contacto telefónico no verso)

- Saída: □ Alta □ Alta contra parecer médico □ Internado □ Transferido □ Óbito
- Data __/02/2018 Hora ___:___:___ Profissional Saúde (Assinatura): ____________________________

Ficha de registo CHO - SVIGCARNIVAL 2018

---

**Figure 4.17:** SVIG Torres Vedras Carnival.
Figure 4.18: Distribution of the health occurrences by diagnostic group.

Figure 4.19: Distribution of the health occurrences by age group.
Figure 4.20: Distribution of the health occurrences by age group, filtering only men.

Figure 4.21: Distribution of the health occurrences by destiny.
Chapter 5: Conclusions

This master thesis focuses on the requirements analysis and implementation of an early prototype of EpiMass. The EpiMass platform was developed according to the list of requirements gathered and presented to me by Dr. Ricardo Mexia, an epidemiologist from Instituto Nacional de Saúde Doutor Ricardo Jorge (INSA). These requirements had been refined during past MG events by the healthcare professional teams in the field. Moreover, to have a better understanding of the challenges involved, I was also present at an MG to understand and experience, in first hand, the system requirements for the platform.

State-of-the-art technologies for web development were used to develop EpiMass. The platform is based on the MEAN stack (MongoDB, Express, Angular, and Node.js), a full JavaScript framework for developing web applications. The best practices for security and authentication of communications were also taken into account, since the EpiMass handles private and sensitive data. Furthermore, all the technologies used are open-source, and they do not require additional costs such as licensing for using the software that was implemented.

The platform has user access control and there are dashboards for each user role, with different functionalities according to their permissions. The user interface was developed to be intuitive, familiar and user-friendly. It is fully responsive and is accessible from any device, since it is available online through a web browser. Currently, EpiMass has two fully functional modules, namely for data collection and for a preliminary real-time epidemiological analysis. In addition, the ability to easily produce reports was taken into consideration. Modularity was also a priority, and more modules can be implemented and added to the application server, increasing the number of available functionalities in the platform. The database was designed to be flexible, presenting many features for extending the API methods, for instance, to extend the interoperability with other web services.

EpiMass was tested during an MG. The platform was shown to be robust enough to handle this flow of data and seemed to be very well accepted by the professionals that participated in the study that were on the field. In fact, most of these professionals provided good feedback as well suggestions for improvement and, moreover, some professionals expressed their wish to use this platform in the near future in similar events.

5.1 Further Developments

Not all requirements were accomplished due to time constraints and the necessity to implement other essential functionalities. Currently, the clinical module still does not allow editing or deleting health occurrences already registered (i.e., the API methods were developed, but the functionalities are still not available on the client-side). Also, the epidemiological analysis module has limited features implemented, such as filtering the occurrences by day.
The ability to sustain losses of connection would translate in an increased robustness of the application, and therefore would be a massive gain for a platform such as EpiMass, working in often complex situations. Therefore, in this thesis, I deeply studied the subject and the technologies, but again, this is a very recent and permanently evolving technology, and in some cases lacking good documentation. In fact, a prototype of the offline module was started, but again, due to time constraints, unable to be completed. Nonetheless, these developments can still be picked up in future work.

Furthermore, the possibility to exchange data with other information sources was not developed. However, there are a set of API methods that can be called through a middle application to access data from health care providers, such as a patient’s history. These specific functionality depends on further interaction with service providers, namely the ones responsible for healthcare providers, both in the National Health System and the private sector. Finally, since EpiMass is modular, an additional module for participatory surveillance could be developed and added to the platform.
References


