University of Glasgow | School of Computing Science

Interactive Website For Rabies Researchers

Christos Kitsos

School of Computing Science
Sir Alwyn Williams Building
University of Glasgow
G12 8QQ

Level 4 Project — March 25, 2016

Abstract

The Interactive Website For Rabies Researchers aims to provide, an online data visualisation and information sharing platform, to people involved in the fight against rabies in Southern Tanzania. The website will provide an interface to an already existing mobile surveillance tool, for the utilisation of rabies related data gathered over the past years. The project was created by Katie Hampson and Rebecca Mancy who are directly involved in the campaign against rabies in Tanzania. Katie and Rebecca are the main clients of this project, and it is through constant interaction with them, that the requirements were identified and design decisions were made. The system implemented was tested throughout the development process, and the final product underwent an extensive user evaluation that was carried with potential users from the United Kingdom and Tanzania. The system was also evaluated through interviews with the main clients of the project in order to ensure that it met all their expectations. The final product fulfilled all the expectations and requirements that were set and the feedback that was received was predominantly positive.

Education Use Consent

I hereby give my permission for this project to be shown to other University of Glasgow students and to be distributed in an electronic format. **Please note that you are under no obligation to sign this declaration, but doing so would help future students.**

Name: _____ Signature: _____

Contents

Chapter 1

Introduction

1.1 Context

Rabies is an extremely serious viral disease that is almost always fatal following the onset of clinical signs. Dogs and other warm blooded animals infected with the disease are the main source of the vast majority of human rabies deaths. The disease is transmitted mainly through animal bites[1].

Over the past 100 years Rabies has been eradicated in most high income and developed nations[2]. The disease though still continues to cause the death of up to 55.000 humans every year mainly in underdeveloped and developing countries in Africa and Asia[3]. Treatment of rabies is possible only if the inflicted person seeks medical assistance immediately after the incidence and gets the treatment needed, due to the nature of the disease if any clinical symptoms appear it almost certainly leads to death[2]. Areas with high prevalence of rabies also get negatively affected socioeconomically, with the financial burden of rabies control and prevention in Africa and Asia, being estimated to exceed 500 million US dollars. Although these calculations might be underestimated as they do not take into account the negative impact of rabies in local communities and economies , considering more than 3 billion people reside in areas where rabies is a present issue[3].

One of the main obstacles that affected nations face against the prevention of rabies, is the difficulty of early detection of the disease outbreaks on human and animal populations, which is fundamental for the creation of effective strategies against the disease[4]. The detection of outbreaks is particularly weak in underdeveloped countries due to the poor and unorganised reporting of deaths and transmission cases inflicted by the viral disease. Thus the case has been made that in order to effectively fight such an under reported viral disease, surveillance systems and reporting infrastructure will have to be strengthened[5].

One of the nations that are directly affected by the viral disease is the United Republic of Tanzania which is located in east Africa within the African Great Lakes region. The southern part of the Nation has attracted international attention by the Bill & Melinda Gates Foundation. The foundation funds a WHO-coordinated project with the aims of controlling the viral disease and eventually eliminating rabies completely from the affected region. The WHO project aims to achieve this by promoting surveillance methods, mass dog vaccinations, improved education, making treatments more widely available, and encouraging an overall better sustainable strategy for the elimination of the disease[6].

1.2 Project Overview, Aims and Objectives

The aim of this software engineering project is to design a web based platform for rabies researchers to share and visualise data and progress, from the campaign against rabies in the afflicted areas of Southern Tanzania. More specifically it aims to serve researchers, government officials, veterinarians and health workers involved with the campaign against rabies in 28 districts in Southern Tanzania, that are funded and supported by the Bill & Melinda Gates Foundation[7].

By supporting these groups through a web site, the project aims to improve the strategies incorporated in the fight against rabies, and provide an effective tool to track the progress that has been made over the past years against the disease. The web platform will have to be easily accessible from areas with unstable Internet connections, and should be usable from a wide variety of devices such as smart phones and personal computers[7].

By making such a platform available to everyone with an Internet connection regardless of their location or the devices they use, the project's goal is to provide information and data to people that are directly involved with these afflicted areas too, such residents of the areas, health workers and veterinarians. The website more specifically aims to provide a way to: visualise rabies related data, share important information and download rabies related reports and files.

The overall project against rabies in Southern Tanzania which this current software project is part of, is a combined effort of various non-profit, government and academic organisations including : Ifakara Health Institute, University of Glasgow, World Health Organisation, Sokoine University of Agriculture, Government of Tanzania, Tanzania Ministry for Health and Social Welfare and the Tanzania Ministry for Livestock and Fisheries Development[8].

Furthermore the overall project receives funding by various international organisations that are invested in eliminating neglected diseases around the world such as rabies. Some of these organisations are : Wellcome Trust, Bill & Melinda Gates Foundation, UBS optimus foundation, Biotechnology and Biological Sciences Research Council and the Medical Research Council[8].

The project described in this dissertation was created by Katie Hampson and Rebecca Mancy. Katie leads field research in Tanzania and works with a network of collaborators on fundamental and operational research, Rebecca works together with Katie as a computational ecologist. Katie and Rebecca are the clients of this project and thus they are considered the two main stakeholders that will be mentioned throughout this dissertation. In addition they both play a major role in the campaign against rabies in Southern Tanzania and the already existing infrastructure there.

Through the support of the fore mentioned organisations and the already existing infrastructure the project's objective is to accelerate the speed with which progress is made against rabies in Southern Tanzania, and provide a skeleton system that can be expanded to cover more affected areas and different under reported diseases around the world in the future.

1.3 Report Outline

This dissertation outlines and describes all the steps that were taken over the software development process of the system, from the requirements gathering process to the evaluation and submission of the finished product. Bellow is a synopsis of the dissertation divided into the main parts of the development process :

Chapter 2 :**Background**
An overview of the existing artifacts that are involved with the project and the research that has been done prior

to its start.

Chapter 3 : **Requirements**
An overview of the different requirements identified, as well as the methods that were used to capture them.

Chapter 4 : **Design**
A description of the main modules identified in addition to design decisions and sketches.

Chapter 5 : **Architecture and Implementation**
A broad examination of the development and implementation of all the aspects of the system, including technical details and major decisions that were made.

Chapter 6 : **Evaluation**
An in-depth overview of the methods used for the evaluation of the whole system, and the major outcomes of it.

Chapter 7 : **Future**
An outline of what the future holds for the system from unfinished requirements and users feedback.

Chapter 8 : **Conclusion**
A retrospective of the product itself, the results of the evaluation and the impact of the system.

1.4 Terminology

- WHO - World health organisation
- Rabies - Rabies is a viral disease that causes acute inflammation of the brain in humans. It spreads mostly through warm blooded animal bites. It is almost always lethal if it is not treated soon after a bite.
- PEP - Post-exposure prophylaxis : It is the vaccination required for the treatment of rabies
- Gates District - Districts in Southern Tanzania that are part of the campaign organised by the Bill and Melinda Gates Foundation
- Esurveliance - An already developed mobile phone surveillance system which is being used in Southern Tanzania
- PDF - Portable Document Format is a file format used to present documents
- CSV - Comma-separated values file is a file that stores tabular data in plain text document
- API - Application programming interface
- JSON - JavaScript Object Notation. JSON is a lightweight data-interchange forma
- NPM - Node Package Manager. NPM is a manager for NodeJS based server that is responsible for installing and updating packages into the server.
- WYSIWYG - ”What you see is what you get text” editor

Chapter 2

Background

This chapter will provide an overview of the overall rabies campaign in Southern Tanzania, and the already existing infrastructure which this project is to be based on.

2.1 Rabies in Southern Tanzania

Since 2010, the government of Tanzania has implemented a large-scale rabies prevention and control programme across 28 districts, coordinated by WHO and funded by multiple non profit organisations including the Bill and Melinda Gates Foundation[8, 6]. This programme involves provision of lifesaving post-exposure prophylaxes to bite victims to prevent the onset of rabies, and mass vaccination of domestic dogs to interrupt transmission and ultimately eliminate disease. The overall project covers more than 150,000km2 catchment area and it affects up to 10 million inhabitants[8].

The biggest obstacle that these districts and the nation of Tanzania faces is the greatly unreported cases of rabies in humans. Information and data of animal bite injuries is an effective tool that can be used to estimate the negative impacts the disease causes a certain community, and can be also be used to monitor epidemiological trends across the affected districts[4].

The unreported incidence in Tanzania, has been slowing the progress made against the disease. Dog bite injuries are an accessible source of epidemiological data, that may be used to estimate the public health burden of rabies and to monitor epidemiological trends in developing countries[4]. Due the difficulties faced in surveillance, the best estimates of the burden that the disease causes on communities and areas, are based on modelling studies. Furthermore most low-income countries including Tanzania use paper based systems for reporting incidents with disappointing results, due to limitations such as, the high probability of falsely recorded data and high cost. Thus it has been identified that for a successful strategy against rabies, a well constructed and efficient paper-less system will have to be used for the effective reporting of disease cases[4, 7].

It has been demonstrated that mass dog vaccination campaigns, is one of the most effective strategies for the prevention and elimination of rabies. But large scale dog vaccination campaigns cannot be done too often ,due to their high cost and the high organisation they require. Thus coherent and well structured data is detrimental to these large scale campaigns so they can be as effective as possible[9, 8].

The use of mobile technologies as a tool for health surveillance has been demonstrated through small-scale examples, and has been proven that they can be highly effective[9]. A successfully large scale phone-based system has been in use since 2011 in Southern Tanzania, and it is directly involved with the software engineering project outlined in this dissertation[8].

2.2 Rabies Mobile Surveillance Tool in Southern Tanzania

Surveillance is critical for managing preventative health services and controlling infectious diseases. Surveillance involves the routine collection, analysis and dissemination of data to guide health policy and practice[7]. But paper-based surveillance is slow and often incomplete[4], therefore it does not allow effective monitoring or timely responses. Surveillance for zoonotic diseases (spread from animals to humans) requires inter sectoral collaboration between the health and veterinary sectors. For rabies, health workers need to report animal bites to veterinary officers to trigger outbreak investigations, and veterinarians need to alert medical authorities to exposure risks from animal cases. Then these outbreaks need to be reported to government officials, for the organisation of large scale preventive measures in the affected areas. This process is hard to be completed efficiently without a strong underlying infrastructure, for collecting the data and sharing the information gathered[8].

An inter sectoral mobile phone based tool has been developed for the surveillance of rabies incidents in southern Tanzania. The system as a whole enables then real-time reporting of rabies related animal bites and the human and animal vaccine use, administered through every region. The mobile phone-based surveillance system is used by over 300 health and veterinary officers across the involved areas, to record data and monitor information. The system has improved the quality and completeness of the data recorded, while at the same reducing the overall costs compared to a paper based system[8].

The system takes advantage of the high popularity of generic mobile phones in the United Republic of Tanzania. The low cost of mobile phones and the widespread network coverage of populated areas make them one of the most ideal methods for gathering data. The system uses GSM which enables regular phones to connect to the Internet through GPRS, even in areas with no mobile Internet coverage such as 2G. Health workers and veterinarians are given the ability to report various events, including bite incidents and PEP doses/shortages. Using a data-entry SMS text messaging based application, the data reported by users gets saved into a MySQL database located in Tanzania[8].

The system developed for this project takes advantage of the Mobile Surveillance Tool and the data collected by it, by using directly the data stored in the MySQL database in Tanzania for sharing information and producing all the data visualisations[8].

2.3 Conclusion

The use of mobile devices for the effective reporting of infectious diseases has been proven to be successful in various occasions. The introduction of such system in the area of South Tanzania has proven it's value by improving the quality of data gathered substantially[8]. The system outlined in this dissertation is aiming to take advantage of the data that has been gathered with this mobile surveillance tool, and compliment it by providing a simple way for researchers, veterinarians, health workers and government officials to share and visualise the data with ease.

Chapter 3

Requirements

This chapter provides an insight on the requirement gathering process, and outlines the requirements that were identified along with their descriptions and priority ranking.

3.1 Requirements Gathering

The initial requirements of the project were mainly revolved around implementing a data visualisation website for rabies researchers. Due to the fact that the main stakeholders/clients Katie and Rebecca, were directly involved in the development of the project, an ”agile” like software development process was used, which resulted in weekly meetings with the main stakeholders where live demos and prototypes were presented to them. Through the discussions that occurred during these meetings, additional requirements regardless of priority kept getting identified until the late stages of the development process.

3.2 Functional Requirements

During the development process, requirements of different priorities and functionalities were identified. Functional requirements dictate the expected behaviour and features of the system as a whole, and of each individual module. These requirements were ranked in terms of their distinctive priority as Top priority, Medium priority and Low priority by considering their importance to the successful completion of the system.

The main functional requirements were identified by considering the expectations and needs that the different groups of users had. Thus the main groups of users directly involved with the functional requirements were identified to be following :

- Superuser
- Website Administrators
- Registered Users
  - Government Officials
  - Researchers / Scientists / Academics
  - Health Workers / Veterinarians

- Non Registered Users

The requirements were constructed with the fore mentioned stakeholder groups in mind, considering their expectations, knowledge, devices used, Internet access and objectives.

3.2.1 High Priority

High Priority requirements cover all the essential features and functionalities that are needed for the successful completion of the system. Table 3.1 which can be found bellow outlines all the high priority functional requirements that have been identified and provides a short description for each of them.

FR	Requirement	Description
1	Data Visualisation Page	There must be a page where the user can interact with the rabies related data gathered in Southern Tanzania
2	Map visualisation	Part of the data interaction must be through a dynamic map consisting of the affected districts
3	Graphs/Charts Visualisation	Part of the interaction must be through graphs/charts etc
4	Select District in Data Visualisation	The user must be able to select a specific district he wants to get visualisations for
5	Select Date in Data Visualisation	The user must be able to select the date range he wants the data to come from
6	Bite Incidents visualisation	The user must be able to get information about bite incidents
7	PEP vaccinations visualisation	The user must be able to get information about PEP vaccinations
8	Vaccination Campaign visualisation	The user must be able to get information about vaccination campaigns
9	Dogs Vaccination visualisation	The user must be able to get information about dog vaccinations
10	Animal Sample Results visualisation	The user must be able to get information about results from animals samples that were submitted to laboratories
11	Suspect Animals visualisation	The user must be able to get information about suspect animals that have been recorded
12	Use of already established database	The system must use the already developed MySQL database from the e-surveliance project
13	Static Content	The website must be able to serve various static content
14	Publications Page	The website must have a page with links to related publications/articles
15	General Information Pages	The website must provide general information about the project and its purpose

16	Mobile Website view	The interface of the website must respond appropriately if it gets accessed through a mobile device

Table 3.1: High Priority Requirements

3.2.2 Medium Priority

Medium Priority requirements are important features that extend the high priority functionalities and implement additional features, in order to cover the needs of all different groups stakeholders. These requirements are not mandatory but they should be implemented in order to provide a robust system. Table 3.2 which can be found bellow outlines all the medium priority functional requirements that have been identified and provides a short description for each of them.

FR	**Requirement**	**Description**
17	Download maps/graphs	The user should be able to download the maps/graphs off the data visualisation page in different formats to use in personal reports etc
18	Login Authentication	The system should provide a login authentication service in order to restrict non registered user access to specific areas of the website
19	User Registration	The website should provide a way to register new users
20	Ability to change password	The website should provide to a registered user a way to change password
21	Download Raw Data	A registered user should be able to download raw data(CSV) files off the website
22	Download PDF reports	A registered user should be able to download PDF reports files off the website
23	Select date range for downloadable content	A registered user should be able to select a date range he wants the CSV files/PDF reports to be about
24	Select District for downloadable content	A registered user should be able to select a district he wants the CSV files/PDF reports to be about
25	Select data/report type of downloadable content	A registered user should be able to select a a specific type of data/report he wants to download
26	News Page	The website should provide a news/articles page
27	Read News posts	The website should provide any user the ability to read a post in the news page
28	Add/Edit/Delete News Posts	The website should provide an administrator user the ability to add/edit/delete posts in the news page

Table 3.2: Medium Priority Requirements

3.2.3 Low Priority

Low Priority requirements represent features that do not create new main components, but instead compliment the already implemented functionalities and enhance the usability of the software. Table 3.3 which can be found bellow outlines all the low priority functional requirements that have been identified and provides a short description for each of them.

FR	Requirement	Description
29	Snapshot URL of visualisation page	The website should allow the user to generate a URL with the state, of the data visualisation page to share with others
30	User Deletion	The system should allow a super user to delete registered users and administrators
31	Reset password	The system should allow a registered user to reset their password if they forget it
32	Send raw data / PDF reports to email	The system should allow a registered user to send a raw data files(CSV) or a PDF reports to their email in case they want it for later use or they are not on their own device

Table 3.3: Low Priority Requirements

3.3 Non Functional Requirements

Non Functional Requirements do not affect directly any of the functionalities of the system, but specify the criteria on how the system should operate. To identify the non functional requirements, the behaviour of the system as a whole had to be considered and the ways the various stakeholders will interact with it. These requirements did not get identified according to the objectives of the stakeholders when using the web site, but according on their overall interaction with the system. Thus over the development process the following stakeholders and their expected non functional needs were identified :

- Website Administrators
  - Modifying Static Content
  - Administrative actions (i.e Database management, Log analysis)
- Future developers
  - Expanding the system
  - Maintaining the system

- Users
  - Diverse devices being used
  - Unreliable Internet Connections
  - Outdated browsers/operating systems

Table 3.4 which can be found bellow outlines all the non functional requirements that have been identified and provides a short description for each of them.

NFR	Requirement	Description
1	Easily Expandable System	The system must be build in a way that allows for easy future expansion and integration
2	Reponsive interface	The website must be responsive to different screen sizes
3	Mobile friendly System	The website must react accordingly when accessed over a mobile device
4	Cross-Browser System	The website must work on different and outdated browsers
5	Simple Implementation	The system must be simplistic so future developers can extend it without much experience
6	Documentation	The system and its functionalities must be clearly documented for further maintenance and development
7	Maintainability	The system must have thorough error checking and be easily maintained and debugged
8	Low Network Load	The system must have low network as a lot of users will have slow and unstable Internet connections
9	Secure	The system must follow strict security standards to protect the sensitive data it utilises
10	Intuitive Interface	The website interface must be clear, simple and easy to use
11	Logging	Important user actions should be logged for future analysis

Table 3.4: Non Functional Requirements

3.4 Conclusion

The requirement gathering process began at the start of the project and lasted until the late stages of the development. Over that time, many new requirements were identified that were not even considered at the early stages of the project. Further more the requirements were identified with regards to all possible stakeholders and the ways they are going to interact not only with the website but the system as a whole.

Chapter 4

Design

This chapter will outline the major design decisions, and how they were made over the entire development process of the project. Furthermore the major modules that were identified will be presented along with some rough sketches of the layout and structure of the system. Due to the agile-like development of the project, the front end of the system (i.e website) didn't have an official final design, but it evolved through the development process, and was guided by constant feedback from the main stakeholders.

4.1 Design Process

The design model that was followed over the development period of the system was an "agile" one. The design decisions were made through weekly discussions and constant interaction with the clients. The design decisions were changing dynamically in conjunction with the introduction of new requirements. Small and large changes were made over the whole development process and design decisions kept being made until the late stages of the project.

4.2 Use Cases

After the main requirements and various stakeholders were identified, a short list of use cases was created in order to help identify the required modules that would make the website. This was done not only by using the requirements identified, but also by getting constant feedback on the front end functionality from the main stakeholders. Bellow you will find a short outline of the use cases that were created :

- As a user I want to be able to and gain information through visualisations on the progress against rabies in southern Tanzania
- As a user I want to be able to download charts/graphs for personal use
- As a user I want to learn more about the project and the parties directly involved in it
- As a user I want to be able to access and read news and articles about the progress against rabies in southern Tanzania
- As a registered user (and above) I want to be able to Login and gain access to restricted parts of the website

- As a registered user (and above) I want to be able to download raw data (CSV) and PDF reports
- As a registered user (and above) I want to be able to email to myself raw data (CSV) and PDF reports
- As a registered user (and above) I want to be able to change my password
- As a registered user (and above) I want to be able to reset my password if I forget it
- As an administrator user I want to be able to register new users
- As an administrator user I want to be able to post articles and news on the website
- As superuser I want to be able to delete registered users

4.3 Identified Modules

By inspecting every individual use case and requirement, it was possible to identify the main front end modules that would be needed in order to cover the main functionalities of the website. The identified modules are outlined in Table 4.1 together with the requirements that every module covers, and a small overview of the design choices including high level technologies chosen for the completion of the specific module.

Module	**Requirements Covered**	**Overview of design**
Data Explorer	F1 F2 F3 F4 F5 F6 F7 F8 F9 F10 F11 F12 F17 F29	The data explorer page will use cutting edge libraries to generate highly dynamic and interactive charts, and will utilise the web server API to gather the data needed
Static content	F13 F14 F15	Static content due to its nature will be hard coded into the HTML part of the web application and it will be loaded automatically, without having to call any sort of requests
Login	F18	Login and athedication will be handled by the web server via the use of well established authentication libraries in order to provide a multi access and secure environment
CSV/PDF downloader and Mailer	F21 F22 F23 F24 F25 F32	The CSV and PDF files will be served through an API in the web server, where they will be generated dynamically according to the users selections and transferred(piped) back to the client or emailed to the users personal email address
Password Changer	F20	The password change will be handled directly by the server in order to do it in a secure manner through an API
Forgot Password	F31	The forgot password module will be served by the web server through a series of API request's and responses. It will utilise highly secure design patterns to protect user accounts from getting exposed
User Registration	F19	User registration will be a functionality provided only to the website administrators and all of it will be handled in the web server through an API

News	F26 F27 F28	The news module will serve as an ”interface” between the client and the news database where all the communication will be handled through the server, including (read/delete/add/edit) news posts
User Deletion	F30	The user deletion module will be available only to a single super user in order to minimise the vulnerability risks of the system. It will be completely managed by the web server, in order to make sure all the appropriate checks have been made before committing such an action

Table 4.1: Identified Modules

4.4 Front End Design

During the design process three modules were identified as having a major impact towards the overall completion of the website. The first one was the main layout and the ways the user would be able to move around the web site seamlessly, and how the flow between of pages would be structured. The second major design decision was the structure of the data explorer, where the design had to be intuitive, informative and at the same time simple to use too. The third module was the news page which had to be simple to use and highly informative.

4.4.1 Website Flow Layout

The design decision that was made for the flow of the website, was the incorporation of a global navigation bar. The navigation bar allows the user to move between all the main views of the website seamlessly, with no restrictions based on a users current location.

Navigation bar for Desktop Devices
The navigation bar bellow is a reconstruction of one of the original sketch that was created together with the stakeholders. The specific navigation bar is made by considering mainly non mobile device users. Through the navigation bar, complex movement flows are eliminated through the website, in favour of a simplistic approach that allows users to move freely between the web site pages with ease.

Figure 4.1: Reconstructed Navigation Bar Sketch for non mobile users

Navigation Bar for Mobile Devices
During the later stages of the development process and with the introduction of new requirements, an alternative

navigation bar had to be designed for mobile users. The mobile navigation bar had to be less cluttered and take advantage of the height compared to the width of mobile devices. By clicking the three bars button the navigation bar expands downwards, and presents vertically the same options as the desktop navigation bar.

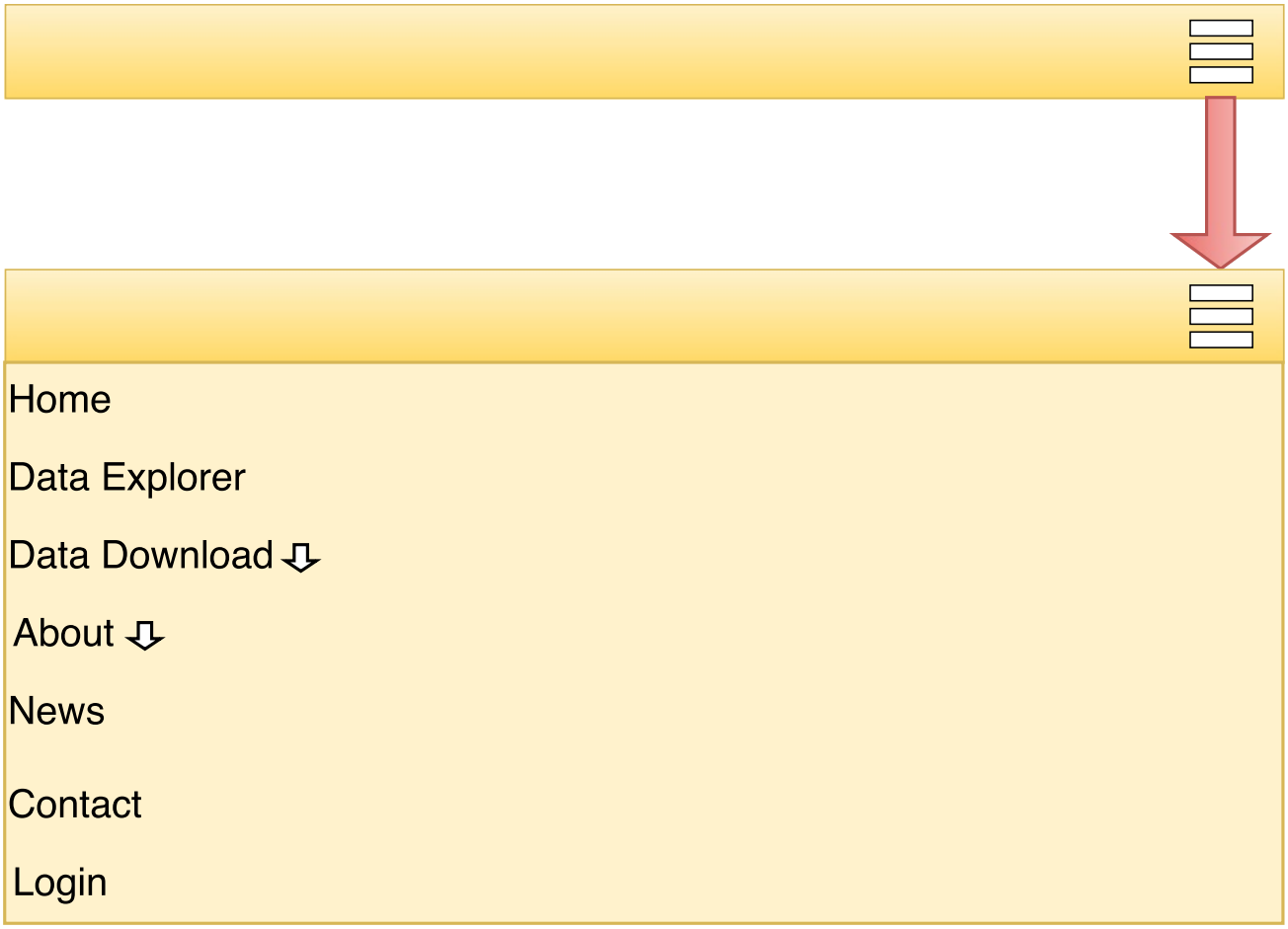


Figure 4.2: Reconstructed Navigation Bar Sketch for mobile users

4.4.2 Data explorer Layout

The most important module of the project and the original functional requirement is the data explorer view. The main idea behind it is to provide researchers with an interactive platform, in order to interact with the complex database that was made from the esurveliance system. Thus the data explorer has to be simple to use while at the same time promoting the clear identification of patterns in the data presented.

The final design utilised data exploration mainly through an interactive map. There, users have the ability to select districts they want to generate graphs and charts for. The interactive map as a selection tool was chosen over a long list of available districts due to the ability to ”jump” between districts with ease, as having to go through a long list of 28 districts would be time consuming. The map together with a data range selection tool allows the users to select the district and date range they are most interested in, while at the same time providing an overview of all the districts through the map visualisation.

Figure 3.4 which can be found bellow provides a reconstruction of an abstract design that was made together with the stakeholders. Keep in mind that the data shown in it is completely made up and was mostly used to showcase the layout of the page.

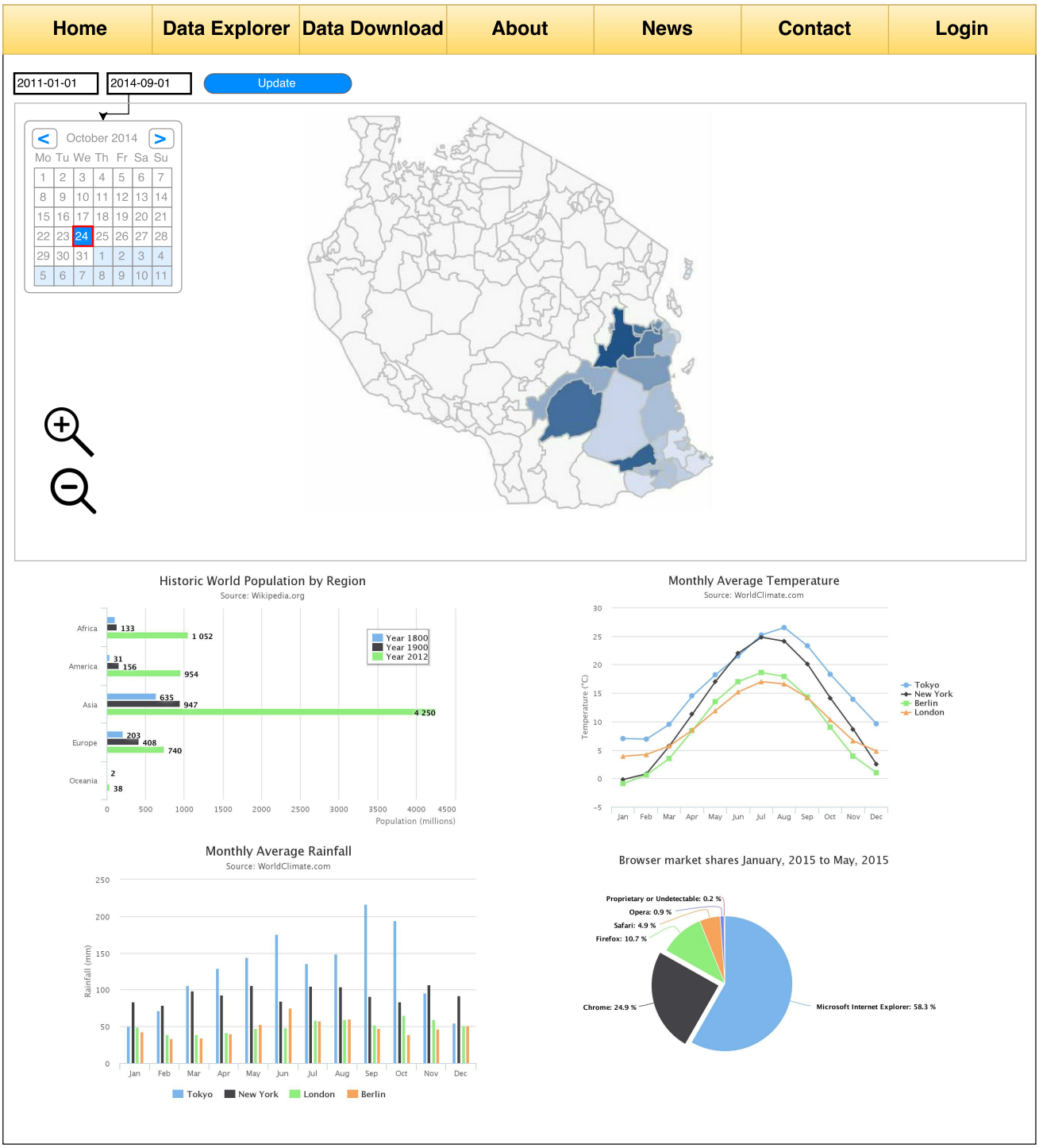


Figure 4.3: Reconstructed Data Explorer Sketch

4.4.3 News Page Layout

The news page had to present the latest news posts to every type of user. The design of this page has the following characteristics. Non administrator users are able to see a list of relevant posts and can select which one they want to read; this is done through the ”Read” button underneath every article. The latest post article is always fully visible, and there is no need to click the ”Read” button to view it. That way users can get the latest important news without the need of looking through the news Titles. An administrator user is given some additional options, the

”add new post” button, ”edit” button and ”delete” button. The ”add new post” and ”edit” buttons should take the administrator to the a view which consists of a WYSIWYG text editor; the only difference is that the ”edit” button should populate the editor with the post selected.

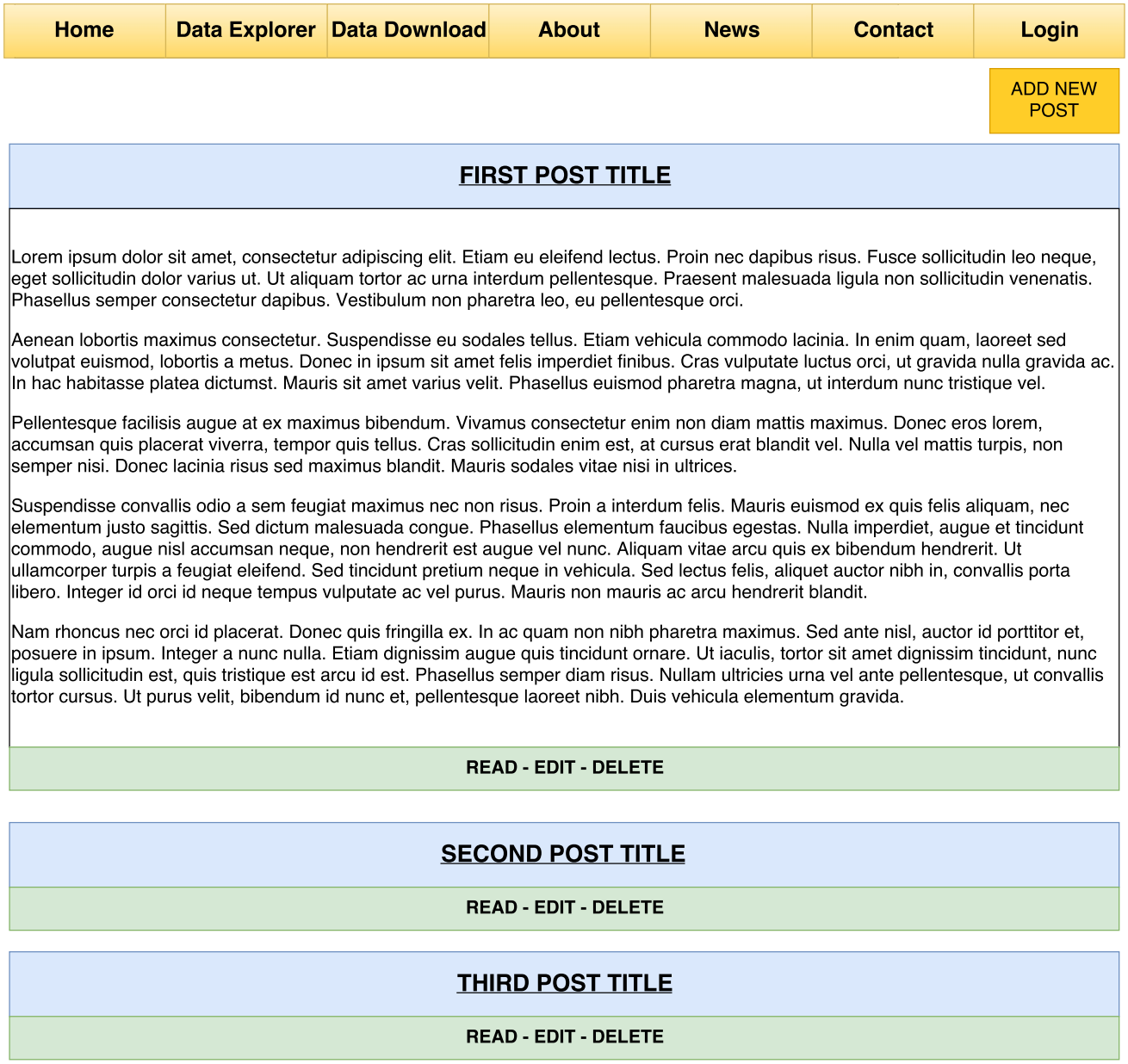


Figure 4.4: Reconstructed News Page Sketch

4.5 User Management and Access Rights

It was identified that the web site will serve four different types of potential users. Each user group will have different access rights and will be limited to certain features of the web site. The users identified are : Non registered users, Registered Users, Administrator User and a Superuser. Figure 4.5 showcases the different access rights each user group has over the main modules and functionalities.

User Groups	Non Registered User	Registered User	Administrator User	Superuser
Data Explorer	✓	✓	✓	✓
Change Password	✗	✓	✓	✓
Reset Password	✗	✓	✓	✓
File Downloader	✗	✓	✓	✓
Read News	✓	✓	✓	✓
Add/Edit/Delete News	✗	✗	✓	✗
Register New User	✗	✗	✓	✓
Delete User	✗	✗	✗	✓

Figure 4.5: User groups access rights by module functionality

4.6 Overall System Design

Due to the dynamically changing requirements an in-depth design of the overall system, was something that could not be captured until the later stages of the development. Nevertheless an abstract design was able to be visualised through the modules and requirements that were identified.

The most important points that were used for the overall abstract design of the system were the following :

- The data has to be queried from an already established database located in Tanzania
- The web server is responsible for the indirect communication between the database in Tanzania and the website, for the generation of graphs/maps and for providing the ability to download raw data files and PDF reports
- All the non rabies related data is to be stored in a separate database packaged together with the web server
- Different devices will access the web site through an Internet browser (No mobile application will be developed)

The above points can be used to identify the main system modules of the system. The Esurveliance database in Tanzania, which is to be used for the the data explorer and the generation of CSV/PDF files for the users to download. The website which is the platform that allows the users to interact with the database, and provide static content as well as news articles. Lastly the web server which is responsible for coordinating the web application and the two databases (1: the database in tanzania , 2: the webserver integrated database) together in order to produce a homogeneous system. Figure 4.6 showcases an abstract idea of how the overall system was expected to be structured.

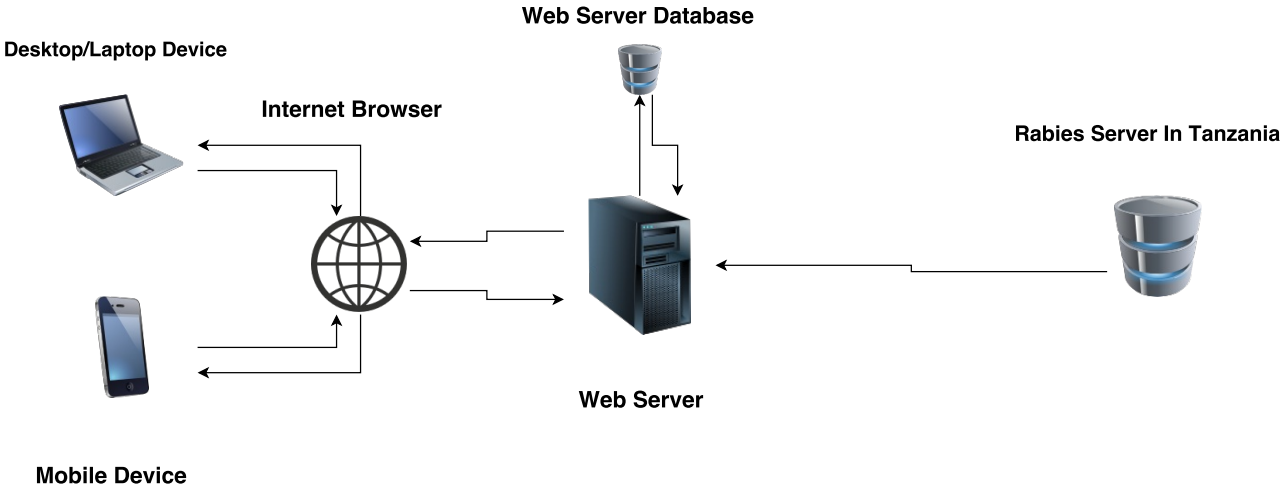


Figure 4.6: Abstract System Design

4.7 Conclusion

In this chapter a high level overview of the design decisions, that were made over the development process of the project were outlined. Due the abstraction of these design choices, and the introduction of new requirements these design choices were not absolute and by the end of the development process they were modified considerably.

Chapter 5

Architecture and Implementation

This chapter will provide an in-depth description of all the technologies that were used for the implementation of the system, along with their alternatives. In addition an overview of the system will be given, along with the logic behind all the main functionalities.

5.1 High Level Architecture

The system developed consists of three different levels of ”layers”. One is the client(or front end) which consists of everything that runs on a users device when he uses the website, this level includes the web browser that is being used and all the files supplied from the server such as the javascript files and CSS files. Another level is the web server(or middle ware). The web server is the base of the system, it is the entity that is responsible for serving important information and data to the client. Then there is the back end(or database) which consists of a single or multiple entities whose purpose is to store data that the web server and the client need in order to function properly.

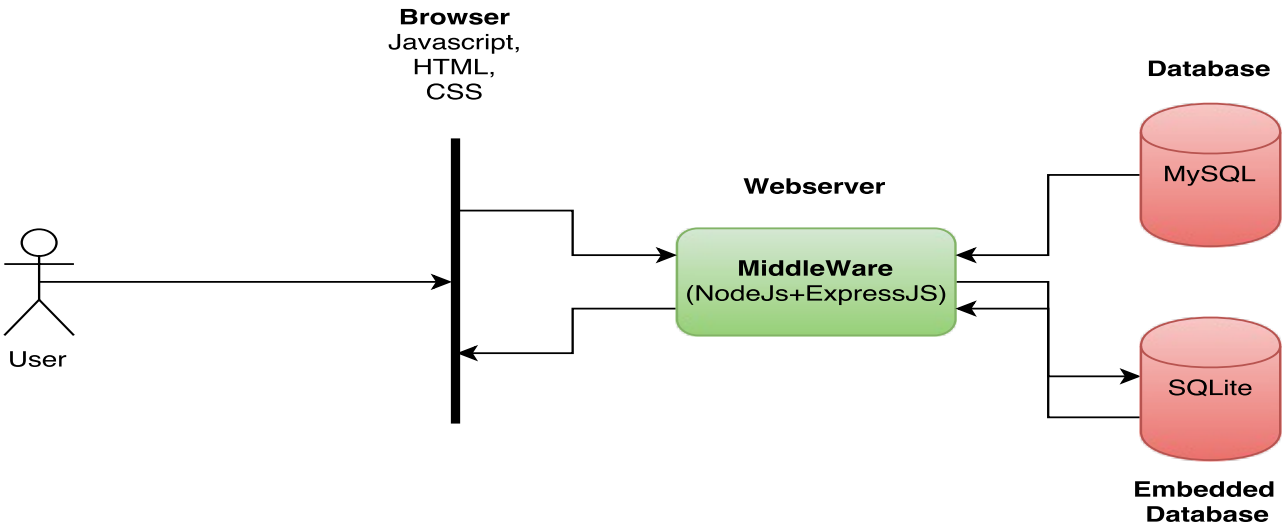


Figure 5.1: High Level Architecture of the System

The current system has a few unique characteristics due to the technologies that are being used and the already existing infrastructure. In this case the middle ware and the back end are interconnected in some cases.

As mentioned before the system uses an already developed MySQL database in Tanzania[8], the system has read-only access on the database which means that the database is not really part of the system or the back end in the orthodox way. On the contrary the system uses an additional database which it has full control off and it is embedded on the web server. That means that that database is practically part of the web server.

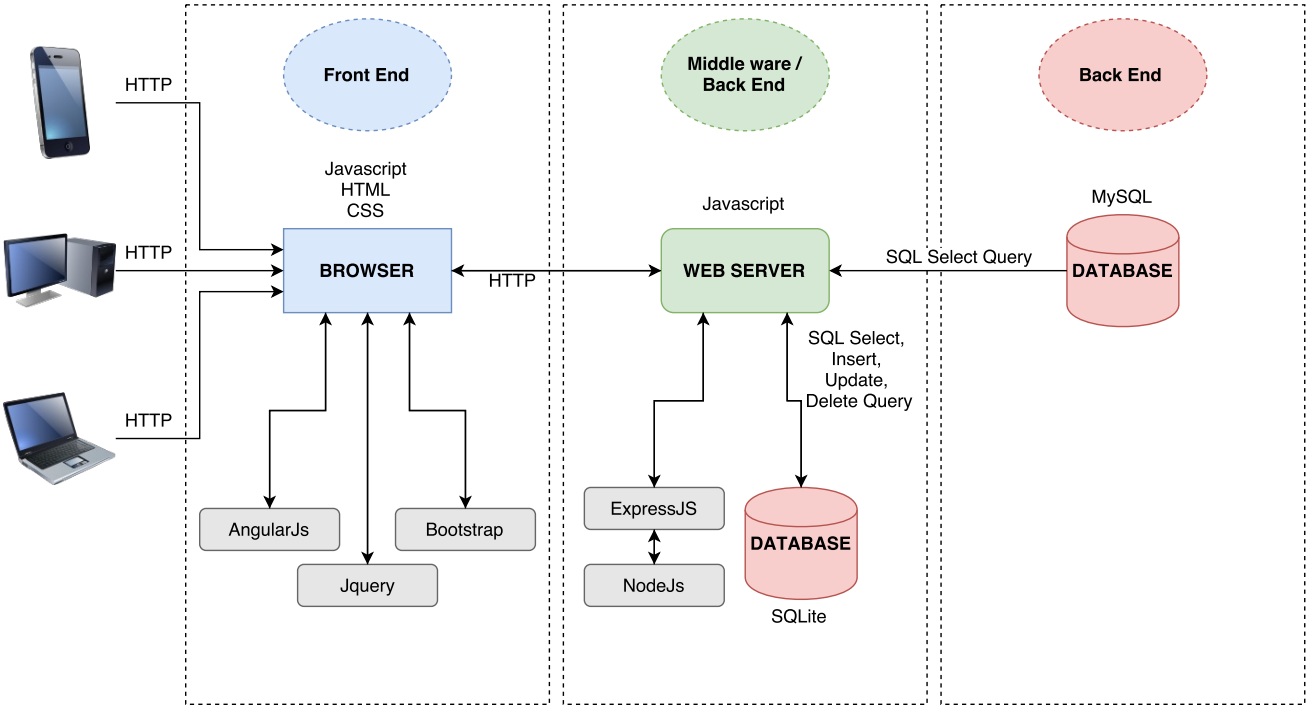


Figure 5.2: Detailed Architecture Design

Each layer of the system uses different and diverse technologies due to the fact that each layer has different purposes and different responsibilities. While all levels are separate from each other it is their combined interaction that forms the system that has been developed, in order to produce an Interactive Website for Rabies researchers.

5.2 Technologies

One of the most important aspects of the implementation during the development of the system, was the correct and well justified selection of technologies. In this subsection all the technologies chosen and their alternatives will be thoroughly described and compared. The reason this report will discuss the technologies chosen so much is due to the fact that this system will continue to be developed and maintained after the end of the project by different developers. Thus the well justified choice of technologies will not only affect the functionalities of the web site, but also the people that will work on it in the future.

5.2.1 Development Tools

Technology	Purpose	Alternative Options
Git	Provide source code managment	SVN
Openshift	Free web application hosting environment	Heroku
Amazon AWS	Highly expandable and reliable web services hosting environment	Openshift/Hekoru

Table 5.1: Development Tools

Git

A revision control tool is almost essential for any kind of software development project. These tools allow a developer to manage and keep track of the source code, they are especially important for development teams consisting of multiple people. Revision control allows multiple developers to work on the same project at the same time.

The two most popular revision control systems are Git and SVN. Git was the tool that was chosen for this project. The tools have multiple differences but Git is currently more popular and supports more functionalities when compared to SVN[10]. In addition Git is decentralised which means that development cant continue without having an direct access to the code repository[11].

Openshift

During the development of the system a hosting environment had to be set up so the website could be accessible. As the website was still in development a free environment was preferable due to the amount of testing that had to be done. Further more the hosting environment needed to support the technology that the system was mainly implemented in (ie NodeJs).

The main identified options were Openshift and Heroku that provided nearly the same functionality. The biggest difference between them was that the free version of Heroku allowed the system to be operational only for 30 minutes if it remained idle[12]. Openshift's free version on the other hand didn't limit the availability of the system and thus was labelled to be the better option[13].

Amazon AWS

After its completion, the product will need to be hosted on a highly secure, reliable and expandable server hosting environment. Amazon AWS provides a huge variety of industry standard services that can be used by the system, and guarantees robustness of both server hardware and Internet connectivity, as well as scalability in case the number of users were to grow rapidly. Amazon AWS provides the opportunity for a seamless transition to more powerful machines, and also to distributed load-balancing server infrastructure. Amazon AWS was chosen as the future hosting environment of the system due to the cutting edge services it provides, the industry standard security and the ability to provide a robust and highly scalable environment[14].

5.2.2 Main Back End Technologies

Technology	Purpose	Alternative Options
NodeJs + ExpressJs	Provide a back-end web server framework for serving the website	Python + Django
MySQL	Handles all the data collected over the past years through the Esurveliance project	None
SQLite	Handle storage of user information, posts/articles from the news view and logs	MongoDB

Table 5.2: Main Web Server Technologies

NodeJs + ExpressJs

The technology used for the implementation of the web server is an extremely important one, it affects the structure of the whole system and restricts any future development to that specific technology. The web server framework was chosen with regards to how it fits the functional and non functional requirements of the project. The system had to be built with regards to the functional requirements, simplicity, expandability and easy maintainability by future developers from various backgrounds.

The main options for the main back end technology were NodeJs+ExpressJs and Python+DJango. They were mainly chosen due to their relatively simple implementation and the large amount of functionalities they provide. Both technologies have the ability to complete all the functional requirements of the system, though their ways of doing that are different. Due to the importance of this decision a set of important aspects were identified for both technologies respectively[15, 16, 17].

- NodeJs + ExpressJs
  - It is written in Javascript which makes switching between front end and back end development easy
  - ExpressJS provides an extremely easy and simplistic environment to develop
  - It is ideal for creating API based back ends that the front end application requires
  - Node Package Manager makes the installing and updating, of libraries and modules extremely easy
  - NodeJS can support fully asynchronous events which makes response times faster
  - A number of modules and libraries can be used in the client as well as in web server
- Python + Django
  - Some of the functional requirements are already built in Django such are authentication and user administration
  - Django is a mature framework and thus most common problems encountered in web development have been already solved
  - Python is a clean and easy language to write code in
  - Django has a larger community

Both technologies provide features that can benefit the implementation of the system in different ways. At the end NodeJs and ExpressJs was chosen, because it covered more non functional requirements when compared to Python and Django. Namely the fact that NodeJs is written in Javascript meant that any future developer would not be required to know any additional languages, as we can assume that most web developers have some basic knowledge of Javascript programming. Further more ExpressJs makes the writing of the web server and the API extremely simplistic and easy, as the implementation doesn't require more than two separate files, one being the server configuration and the other one being the API[16]. In addition NPM provides an easy way to incorporate new functionalities into the web server, and some of its modules can be used in the front end of the system as well. This provides any future developer with more freedom to develop new functionalities in different layers of the system[18, 15].

Overall both technologies can be used to provide a simple and successfull implementation, but NodeJs + ExpressJs was chosen mainly due to the better coverage of non functional requirements.

MySQL

Due to the specification of the project an already developed database in Tanzania that the web server has no control of had to be used, in order to query the data needed for the major functionalities of the system. The specific database was made in MySQL and thus there were no other alternative technologies that could be used.

SQLite

In addition to the mobile surveillance tool database located in Tanzania, the system required an additional database that had to be fully controlled by the web server. This database would be responsible for storage of users, new articles and logs. Due to the limited data that would be stored, the database had to be simple and light.

The main considerations were SQLite and MongoDB. SQLite is similar to MySQL with the difference that it doesn't need a hosting environment and it can be simply embedded onto the web server[19]. MongoDB on the other hand is a non-relational database that works extremely well with nodeJS due to the ability to store JSON objects in it. MongoDB provides more functionalities when compared to SQLite, but these functionalities were unneeded for the limited use of the web server database[19, 20]. SQLite was chosen as the web server database due to its lightweight structure, the familiarity of the stakeholders with SQL databases, and the potential ability to integrate the Tanzanian database with the websites database in the future.

5.2.3 External Back End Technologies

Technology	Purpose	Alternative Options
Passport	Provides user authentication and various security measures	Stormpath
NodeMailer	Provides a fully customisable email service	Express-Mailer
PDFMake	Dynamic PDF client and server side generation	PDFKit
JsonToCsv	Customisable JSON to CSV file conversion	NodeCsv
node-SQLite3	Provides the ability to create and query an embedded SQLite database	None

node-MySQL	Allows the web server to query and establish connection with MySQL databases	None

Table 5.3: External Web Server Technologies

Passport

Due to the sensitive data that the system utilises, user authentication was one of the main requirements. NodeJs + ExpressJs provides many different ways of dealing with user authentication and authorisation; but it came down to two of the most popular technologies. Passport and Stormpath are modules that can implement user authentication on a NodeJs based web server. Passport is known for its simplicity, variety of options and customisation while Stormpath is a fully fledged enterprise oriented security technology[21, 22].

Passport was chosen for the system due to its simplicity and because it covers all the needed requirements while still providing the ability for further expansion (such as facebook/Google account authentication). In addition Passport is entirely customisable and there are more than 300 diverse custom authentication strategies available on its website[21]. Furthermore passport is completely free, while Stormpath requires a monthly fee for the utilisation of all its functionalities[22].

NodeMailer

One of the requirements of the system was the ability to send emails to registered users. There are many free and non-free email technologies for NodeJs based servers available. But there were two specific email modules that were identified to be simple enough to use, provided enough customisation and fitted all the requirements, NodeMailer and Express-Mailer.

Both of the technologies are extremely lightweight and simple to use, but NodeMailer was chosen for multiple reasons. NodeMailer provides more functionalities overall and is more well supported by the community. Furthermore Express-Mailer has a substantial restriction of forcing you to construct an email via the use of HTML, which with regards to this project was found to make things too complicated[23]. In addition NodeMailer is the de facto email service technology for NodeJS based servers[24].

PDFMake

A technology had to be found for the generation of dynamic PDF reports. The two most popular technologies available to NodeJS based web servers were PDFMake and PDFKit. PDFKit is the most customisable and extensive technology of the two but lacks in two important aspects. Firstly PDFKit is a lot more difficult to learn how to use successfully, and does not have built in support for table generation which is an important part of the reports required[25]. Both of the technologies would be viable but PDFMake was chosen due to it's simplicity and the built in table generation that it provides[26]. The built in table generation functionality is especially important as they are used in every type of PDF report, thus future developers would face a large obstacle on expanding the PDF generation functionality; if PDFKit was chosen.

JsonToCsv

Another requirement was to allow a user to download automatically generated CSV files. These CSV files consist of raw data taken from the MySQL database in Tanzania. For this functionality there were two available, technologies available JsonToCsv and NodeCsv. JsonToCsv was chosen over NodeCsv due to the easy configuration, but in return some additional functionalities had to be sacrificed. Due to the basic need of CSV generation that was needed by the system, the more simplistic JsonToCsv was chosen[27, 28]. In addition JsonToCsv has a large community which can ensure that it will keep getting updated in the near future.

SQLite3 and MySQL

Another set of technologies that had to be chosen were the ones that would enable the web server to interact with the two databases of the system. Due to the direct involvement that these technologies would have with sensitive data, it was decided to go with node-MySQL and node-SQLite3 because they are directly supported by NodeJS. In addition they were chosen due to their large community and high standards of security in order to protect the databases from harmful attacks, such as SQL injections[29, 30].

5.2.4 Main Front End Technologies

Technology	Purpose	Alternative Options
Javascript	Client-side processing	None
HTML+JADE	Displays information and structures the ”view” of a web page	Plain HTML
AngularJS	Provides the main functionality and structure of the front end application	ReactJs, Backbonejs, Plain Jquery
Jquery	Provides the ability to interact with HTML and is a main prerequisite of most external libraries.	Plain Javascript
CSS+Bootstrap	Provides rich, responsive and interactive user interface	Angular Material, Plain CSS

Table 5.4: Main Front End Technologies

Javascript

Javascript is essential for any non plain HTML website. Nearly every website on the Internet uses it, thus it was an obvious choice.

HTML+JADE

HTML is essential for all web applications, as it is responsible for the presentation of information. The front end of this project uses the JADE template on top of HTML. JADE is the default HTML template that comes with ExpressJS, it works exactly the same way HTML does but provides a few additional functionalities, such as the ability to render an HTML page dynamically through ExpressJs without the use of Javascript[16, 31].

Further more JADE makes writing HTML a lot faster and simpler, as it dumbs down the syntax to a more readable python-like format. Overall JADE+HTML can do everything plain HTML can do, with the addition of some extra functionalities when used together with ExpressJS[16, 31]. In addition as it can be seen bellow the JADE template avoids using closing brackets, which leads to more error safe HTML code. Furthermore the majority of the ExpressJS tutorials and examples on the Internet use JADE, which is an important consideration for the future developer of this project.

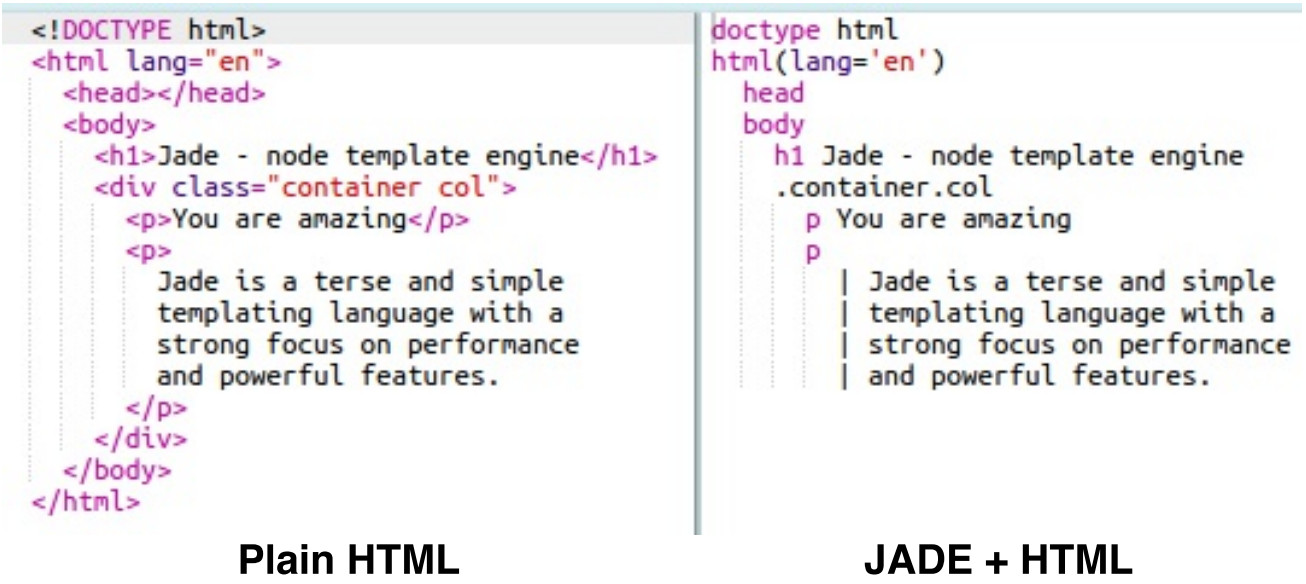


Figure 5.3: Comparison between HTML and JADE

AngularJS

The second most important technology after the web server framework is the front end framework that will shape the structure and the behaviour of the website. While plain Javascript can provide simple functionalities for a website, it can easily become very complex in large front end applications due to its lack of structure. Therefore it is extremely important to use a strong Javascript framework for the development of a website, as it will reinforce a clear structure and good programming practises.

Some of the most popular Javascript frameworks currently are the following : Angular Js, Backbone JS, Ember Js and React JS. Angular Js was chosen not only for its extremely popularity and continuous support from Google, but because it provides a huge array of functionalities and a great way to structure the front end of the application[32].

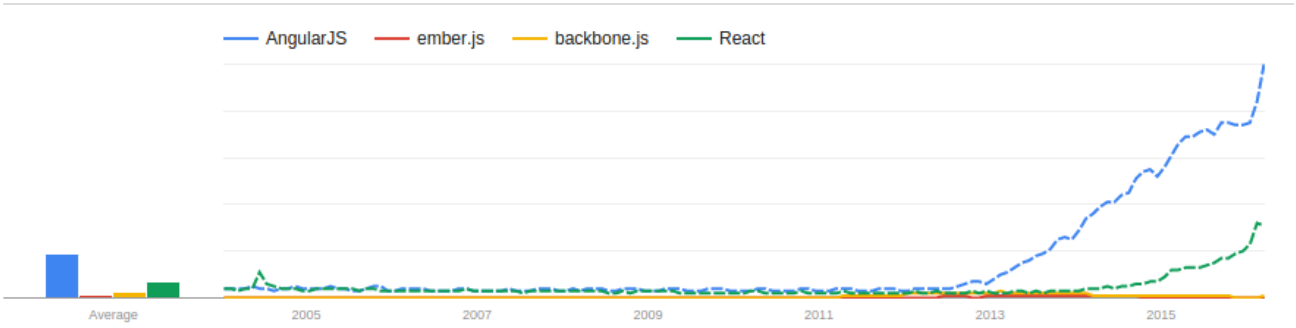


Figure 5.4: Popularity of main Javascript Frameworks

Angular's primary goal is to simplify the development of a dynamic web application through innovative concepts such as directives and two-way data binding. The combination of directives and two way binding is one of the most useful tools the framework provides a developer with[32]. These technologies allow the seamlessly dynamic change of HTML elements without the need of refreshing a web page. This not only simplifies the implementation of the client but also benefits the user interaction with the interface.

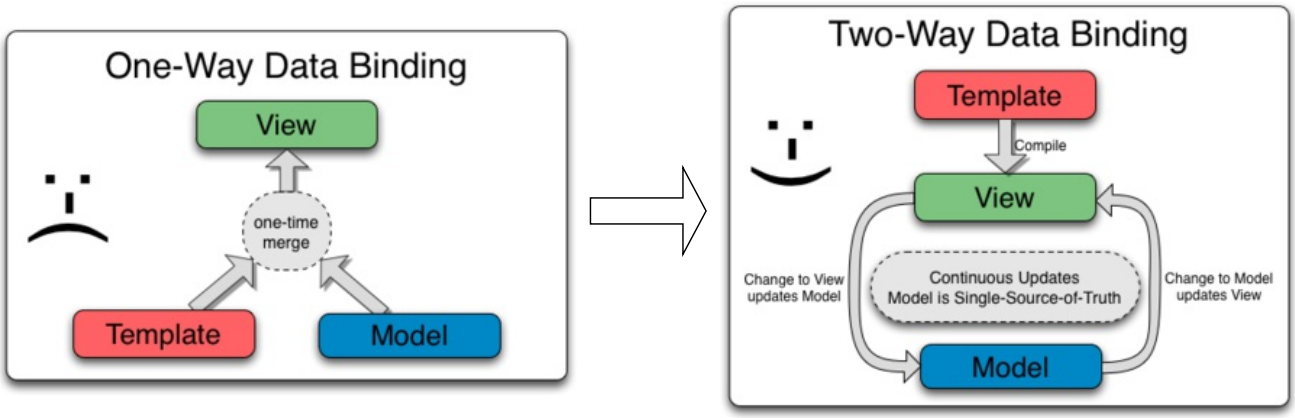


Figure 5.5: Difference bettwen traditional one way binding and Angular's two way binding

Furthermore AngularJs has the largest amount of contributors out of any other Javascript framework, which can assure that it will remain relevant for the foreseeable future[33]. Angular also supports a huge array of external modules and libraries, that can be incorporated into the front end to extend its functionality[34]. It is for these reasons that the AngularJs framework was chosen to structure and shape the web application.

Jquery

Jquery is by far one of the most used Javascript technologies, as it gives Javascript the ability to directly interact with HTML. For this reason the majority of Javascript-HTML-CSS interactions are done by jQuery[35]. For the specific project there was no real alternative to jQuery because all the CSS and data visualisation technologies depended on it[36, 37]. For this reason jQuery was not included into the client as a main technology but as a complementary one in order for other technologies to work properly.

CSS and Bootstrap

CSS is the main technology that gives a web page its "appearance". It is the look and the feel of a website as it is responsible for rendering any graphics, placing objects onto the correct position and sizing elements. Bootstrap is a CSS framework developed by Twitter and it is the most famous CSS technology at the moment. While there are a lot of CSS frameworks Bootstrap was chosen because it provides a specific functionality that is invaluable for this specific project, responsiveness. Bootstrap resizes and relocates objects dynamically and automatically according to the device used when accessing the website. This allows a web page to be rendered successfully on desktops and mobile devices without the need of implementing two separate views. On the downside bootstrap is not the most lightweight CSS framework but it was still identified as the best choice[36].

5.2.5 External Front End Technologies

Technology	Purpose	Alternative Options
HighCharts	Provides the ability to generate interactive graphs and charts	D3
HighMaps	Provides the ability to generate interactive map visualizations	D3
TextAngular	WYSIWYG text editor with rich HTML content	Custom Text Editor

Bootstrap Dialog	Provides a customisable and simple way to produce modals	Custom created Modals
Bootstrap Datepicker	Dynamic date selection module	Angular Date picker

Table 5.5: External Front End Technologies

HighCharts and HighMaps

The main requirement of this project is to provide researchers, with a way to visualise and interact with data. For this a highly customisable and well made front end visualisation technology had to be chosen.

The most popular data visualisation technology currently is D3. D3 is extremely customisable and allows a developer to construct virtually anything. It would be the perfect choice for this project if it wasn't for 2 main drawbacks. Firstly D3 has an extremely steep learning curve, which makes it very difficult to use without any prior experience, and the technology itself does not support some of the older Internet browsers that are still common in Tanzania[38]. Due to this a more cross browser friendly technology that would cover all the functional requirements, while still being easy to use; had to be found.

HighCharts and HighMaps which are both developed by Highsoft AS were chosen due to covering all the visualisation requirements. They respectively allow the easy generation of many different chart and map data visualisations. Both technologies are extremely simple to use as all it takes is modifying their configuration and passing the correct data needed to generate the visualisations. In addition they come with a wide range of plugins such as the ability to download a visualisation in the preferable file type. Furthermore the Highcharts website provides a large amount of examples that can be easily used and customised by developers with no prior knowledge[37].

TextAngular

Text angular is a technology that provides an interactive WYSIWYG text editor, that can be used on the client side of the system. It allows the user to format rich text and add graphics/images into a text box. The technology was chosen in order to provide administrators with a rich tool to write posts and articles on the news page. One of the important benefits it provides is that the text written inside the editor can transformed into plain HTML code which can then be saved in the embedded SQLite database[39]. This allows the query of rich articles and posts straight out of the SQLite database.

Bootstrap Dialogue and Datepicker

Bootstrap dialog and Bootstrap Datepicker are community made complementary modules for Bootstrap. They provide an easy and customisable way of creating pop-up modals and date ranges selectors. Bootstrap datapicker is an important module of the front end application as it covers an important requirement for the download PDF/CSV pages and the data explorer. Bootstrap dialog allows the developer to generate pop up modals without the need of implementing a lengthy Javascript logic for them to function properly[40, 41]. They were both chosen in order ot make the implementation of these functionalities much simpler.

5.3 Web Server

The web server was built in NodeJs + ExpressJs in order to keep the implementation as simple as possible. The server component of the system consists of only 3 files. In addition the web server and web application are

bundled in the same package as a result of using ExpressJS.

File Name	Purpose
Server.js	This file contains all the configuration of the NodeJS + ExpressJS web server and the configuration of important libraries such as Passport and NodeMailer
index.js	This file contains the routing functionality of the system and is responsible for rendering the correct HTML+JADE file when requested by the web application
api.js	This file contains the implementation of the API . Most of interaction of the web server with the client application is implemented in this file

Table 5.6: Web Server Files

NodeJS + ExpressJS offers high throughput via non-blocking asynchronous I/O which is ideal for web applications that are mostly based around making requests to an API. NodeJS comes with its own module packager called NPM. NPM allows a developer to install a module or a library onto the server automatically by just entering the following command ”npm install -S ’library name’ ”.

The API endpoints can be found in the Appendix of this dissertation.

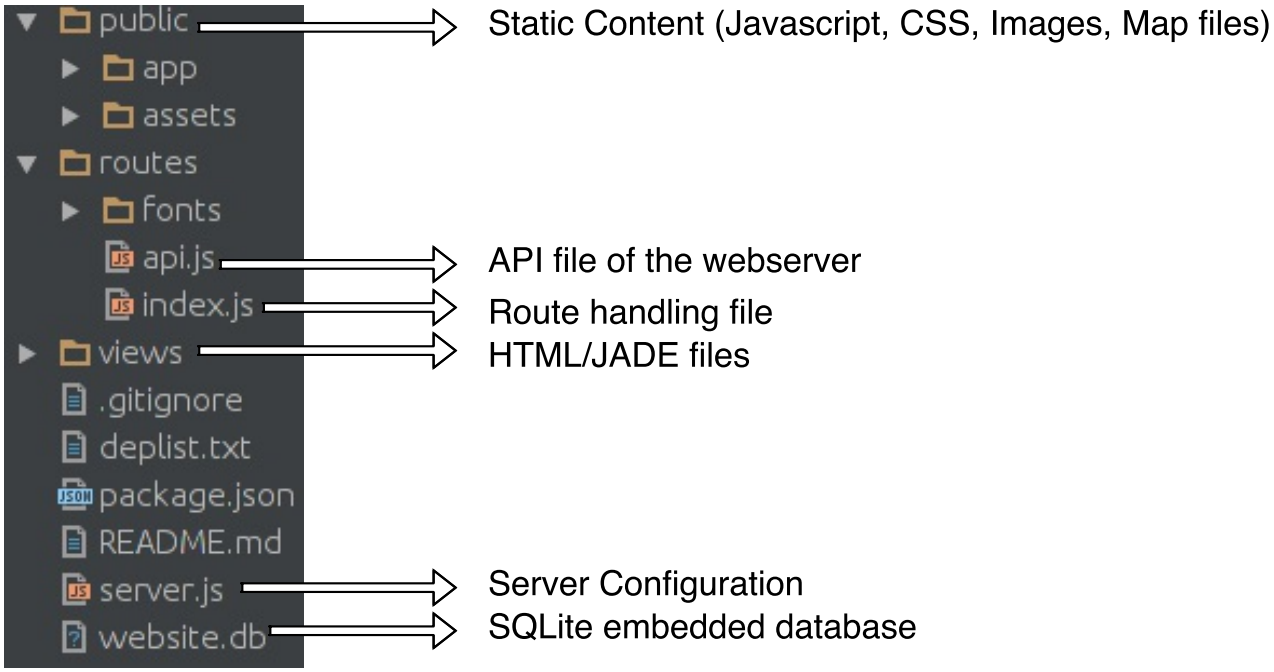


Figure 5.6: Web Server Structure

5.4 Web Application

The front end, client, web application or simply website is the platform that lets the user interact with the overall system. The client in this system is an entity of it self, it uses technologies and provides functionalities that are

different from the ones the web server does. The client is written in Javascript, JADE and CSS. In addition to these basic languages it utilises many technologies that improve the look, the structure and the interactivity that the front end website showcases.

The technologies that are used the most in the front end are; AngularJS, Bootstrap and HighCharts/HighMaps. AngularJS is responsible for the structure and the logic that acts behind the visual representation of the web page. Bootstrap is responsible for the interface of the website while at the same time providing the logic that makes the website highly responsive on a wide array of devices. HighCharts and HighMaps in combination with AngularJS and JQuery provide the generation of interactive data visualisations in the Data Explorer Page.

The front end implements an MVC(Model-View-Controller) architectural pattern with the help of AngularJS. The MVC pattern divides the front end into three separate but interconnected parts[32].

- The **Model** , shared by both the view and the controller. The model represents the data that is shown to the user in the view and is editable by the controller.
- The **View** is essentially a HTML template that defines the structure of the page; static components like forms, buttons and text are defined here. It also contains variables which represent the objects held in the model part of the application and are updated dynamically. The style of the page is modified using CSS.
- The **Controller** which defines the the behaviour of the page and implements most of the functionalities of the client.

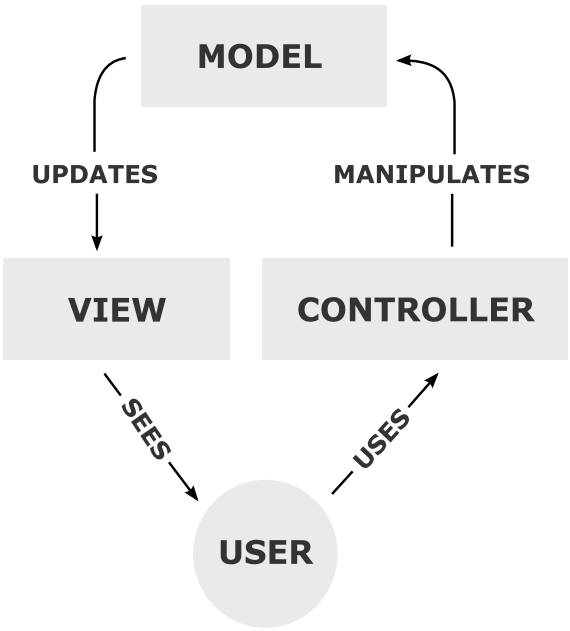


Figure 5.7: MVC Architectural Pattern

5.5 Database

The back end of the system is connected with two separate databases, the one used by the mobile surveillance tool located in Tanzania and another one that is embedded on the NodeJS based web server. Due to the fact that

the database in Tanzania contains sensitive data and the web server has read-only control this dissertation will only mention it from a high level standpoint.

SQLite is a relational database similar to MySQL, with their main difference being that SQLite can be embedded on a web server, while MySQL needs to be hosted separately. Due to the low requirements of the website in terms of database storage, the use of a lightweight database such as SQLite makes the overall system simpler. In terms of query functionality they are nearly the same and an SQLite database can be migrated into a MySQL with ease in the future, if the demands of the database increase[19, 29].

The web server uses an SQLite database for everything that has to do with the functionality of the web site as an entity, thus it deals with only non rabies related data; that involves the following high level items :

- User Authentication
- News/Posts Storage
- Logging of User actions

Bellow you will find figures and tables that give a general overview of the tables in the database, and provide a brief description of what every column gets used for.

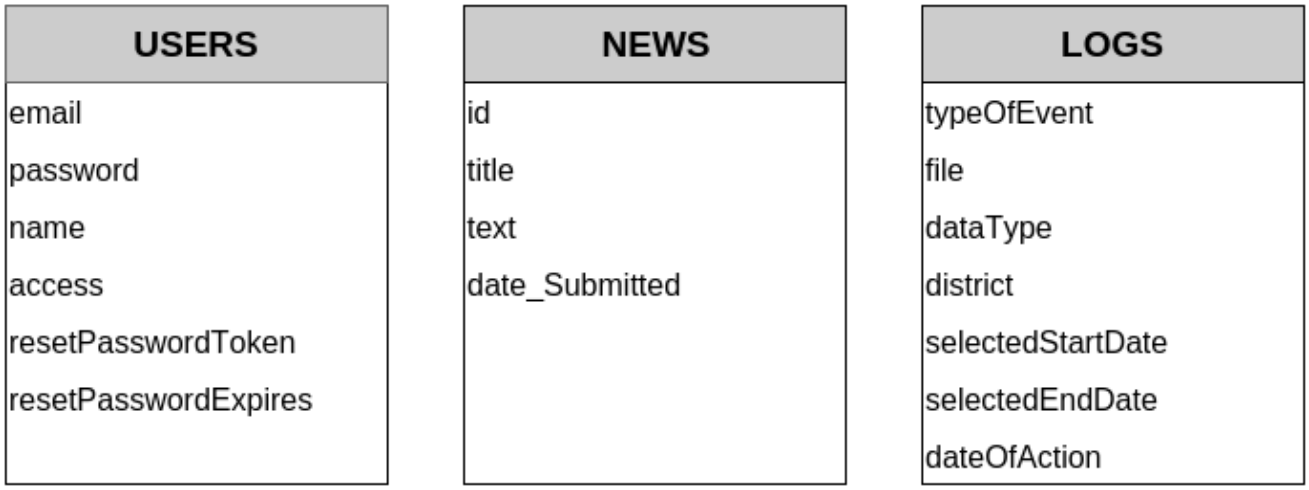


Figure 5.8: Database Table Overview

Column	Purpose
email	Stores users personal email. Also used as a unique user name for login
password	Hashed password of user acount
name	Name of user
access	Authorisation level of user
resetPasswordToken	Randomly generated token for the password reset functionality
resetPasswordExpires	Date and time until password token expires

Table 5.7: Users Table

Column	Purpose
id	Unique identifier for news post
title	Title of news post
text	Content or text body of news post
date_Submitted	Date of the specific date a news post was created

Table 5.8: News Table

Column	Purpose
typeOfEvents	Type of event logged, email/download
file	Type of file requested PDF/CSV
dataType	The type of data requested
district	District requested by the event
selectedStartDate	Start of date range requested by the event
selectedEndDate	End of date range requested by the event
dateOfActions	Date of when the event got requested

Table 5.9: Logs Table

5.6 Implementation

Identifying the right technologies for the development of the system is only half the job, the other half is implementing all the functionalities required. In this section an in depth overview of the implementation of every module will be thoroughly explained.

5.6.1 Data Explorer

The data explorer module serves one main purpose, to give the ability to the user to visualise and interact with the data gathered from the Esurveliance project in Tanzania, in a simple and informative way. Further more another functionality it serves is the ability to save specific ”states” of the page via an URL generation function, that can be shared with other people.

The ”state” URL can be constructed in the data explorer page by clicking a button. The URL is structured by appending parameters that affect the visualisations at the end of the default URL.

Figure 5.9: Default and State URL Structure

When the module is loaded it first checks if the URL it is accessed by is the default one or if it is populated with specific data. When the data explorer module gets the default or ”state” URL values, it uses the web server API to request the data needed; which is returned in a JSON format from the server in Tanzania. As the majority of the data formatting gets done by the web server the client only needs to make a few modifications, so each data set can work correctly for the different visualisations. The data explorer module passes the appropriate data into predefined HighChart and HighMaps configurations, which in turn generate all the visualisations.

The visualisations created by HighCharts and HighMaps are interactive, the user is given the ability to select specific dates to be shown in each chart visualisation, without the need to call the web server API. The data explorer needs to use the API for the generation of visualisations for different districts though, which is done dynamically without the need to refresh the web page due to the use of Jquery and AngularJS.

In order to construct a map visualisation HighMaps required a map of Tanzania in a GeoJSON format, which is a JSON like format for map files. A map of Tanzania was provided by the main stakeholders of the project but it was in shp (shapefile) format. In order to solve the problem QGIS which is a map editor tool was used in order to transform the map file into the required format. Another problem that was encountered was that the file that was generated was larger than 1 megabyte which was problematic, considering one of the main requirements of the project was to provide a lightweight platform. In order to solve this the Douglas-Peucker and Visvalingam algorithms were applied to the file[42, 43]. These algorithms simplified the small details of the map file to make it more uniform and reduce its size. The file ended up being 195kb in size which made a substantial performance difference.

Another issue that was encountered with the map visualisation was how the data from the database could be mapped to values in the map file, in order to generate the correct visualisations. The value that was chosen was the postcode of every district. That way it is possible to map the correct data in the map visualisation, as the database includes the full postcodes for any data that is submitted in it. So in the GeoJSON map file a postcode value was added into each district that the map consisted of. Using postcodes will allow future developers to

expand the number of districts visualised, and it also ensures the long term functionality of the data explorer due to the fact that postcodes are static and unique for each district.

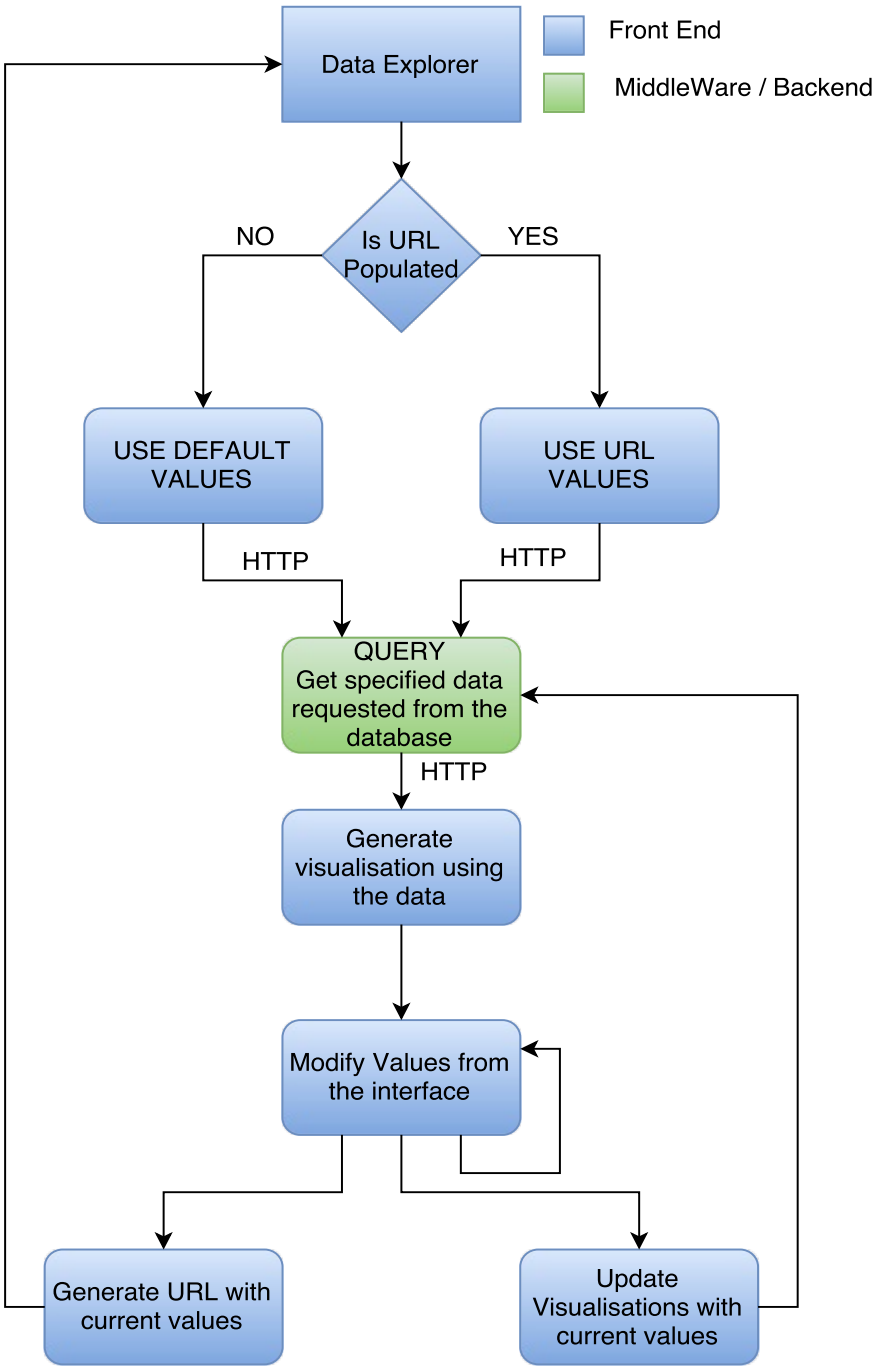


Figure 5.10: Data Explorer Functionality Design

5.6.2 Login and Forgot Password

The user authentication feature provides three distinct functionalities. One is the ability to log in as a user in order to gain access to protected pages and API's of the system, the other one is the initiation of the "forgot password" functionality which is used when a user can not remember his password. The last one is the ability of a user to change their password.

Login

When a user tries to log in into the system a HTTP request gets made to the server which includes the specified user name and password. The web server uses the user name to query the SQLite's users table and checks if the user exists. If the user exists the web server checks if the password supplied is the same as the hashed password stored in the database by using the NodeJs Crypto library. If the user supplied the correct credentials the web server calls the Passport local authentication strategy which serialises the user into the middle ware and ensures, that with each request the serialised user object gets passed as a response parameter. Whenever a user tries to access a protected web page Passport verifies if the user has been serialised before redirecting the request to the protected page[21].

Change Password

The change password functionality allows a user to change his password while he is logged in. When a user visits the change password page he has to provide his personal email address, his current password and the new password he requests. When this is done an HTTP request gets made with these parameters, passport then verifies if the user is serialised(logged in). The web server then checks if a user with that email exists, and then checks if the hashed password in the database matches with the password provided in the request. Subsequently the server replaces the password in the database with the new password after hashing it; only if the user provided the correct credentials and was logged in.

Forgot Password

If a user clicks the forgot password button in the Login view, the web server redirects him to a page that asks him to supply his personal email address/user name that he has been registered with. The page creates an HTTP request to the server including the email/username specified by the user, and then goes on to check if the user exists by querying the email in the SQLite database. If the user exists the web server generates a randomised token that it inserts into the resetPasswordToken column of the user table, and it generates a time stamp of exactly one hour later from the time of the event. The time stamp then gets inserted in the resetPasswordExpires column of the user table. After this is done the web server sends an email through NodeMailer to the the users personal email address with a URL link, that consists of the reset password page URL and the generated resetPasswordToken appended as a parameter[24].

When the user tries to access the emailed URL, an HTTP request gets called automatically. The web server then queries the SQLite database looking for a user, that has the appended token URL parameter in the reset-PasswordToken column of the users table. If a user with the token is found, the server checks if the current time is later than the one entered into the resetPasswordExpires column of the user table. If the resetPasswordExpires is expired, the web server redirects the user to the home page and removes the values from the resetPassword columns in the database as the token is no longer valid.

If the token is valid the user gets redirected to the reset web page, with a URL that includes the generated token and the expiry time stamp as parameters. At the reset page the user is asked to enter a new password desired twice, and after it is confirmed another HTTP request is made to the web-server. The server then checks again the SQLite database for the user with the specified token and if he exists, the old password gets updated to the new hashed password specified by the user. Afterwards the resetPassword columns get emptied and the user gets redirected back to the home page. In addition the web server sends a confirmation email every time a password is reset.

5.6.3 CSV/PDF Downloader

The CSV and PDF download pages are separate and cover different requirements, but they function in similar ways. They both revolve around the user requesting different PDF/CSV files for different districts and different time ranges. The user is also given the option to download the requested files or send them to his personal emails. The main difference between is the way the web server generates the requested file.

When the user visits either of these pages he gets requested to specify the date range, district and type of CSV/PDF file he is interested in, if he doesn't there are some preselected default values in place. By clicking the Download or Send to Email button an HTTP request is made to the web server with all the values selected as parameters.

According to the type of file requested, the web server queries the MySQL database in Tanzania to get the data that is required for the generation of the file requested. After the data has been queried the web server needs to create the file object dynamically without saving it before sending it to the user.

During the development of the system it was identified that the files could be generated either in the web server or in the front end. The decision to generate them in the back end was made in order to keep the client as light as possible, as potential users will be requesting these files from outdated devices.

CSV Generation

When the web server queries the database the object that is returned is in a JSON format. Converting a JSON object to a CSV format can be done in many ways. The web server uses the JsonToCsv library that is made for NodeJs servers, JsonToCsv generates a CSV formatted string when it is given a JSON object. The server needs to pre specify the headers of the CSV file and then pass the data into the JsonToCsv for the string to be created[28].

PDF Report Generation

When a PDF report file is requested the web server uses the PDFMake library to generate the report. The web server contains predefined templates for each type of PDF reports that can be requested, they consist of static content and configuration options such as fonts, font size and margins. In the predefined templates there is a table initialisation where the web server inserts the data that was retrieved from the MySQL database in Tanzania for the generation of the tables. The PDF file is created as an object on the web server and it is not saved anywhere[26].

File Download

If the user chooses to download the file the server reacts in the following ways. For the CSV file the server sends an HTTP response with the CSV formatted string as the content and sets the headers appropriately to let the browser know, that a file in CSV format is expected. For PDF reports the web server pipes the PDF object back to the client by simply using the pipe() method[16].

File Email

If the user requests to receive a personal email with the file requested, the web server sends an email to the user by using the NodeMailer library with the file generated as an attachment[24]. The email address is received from the request, as Passport makes sure that every request contains the user object in it, if the user is logged in.

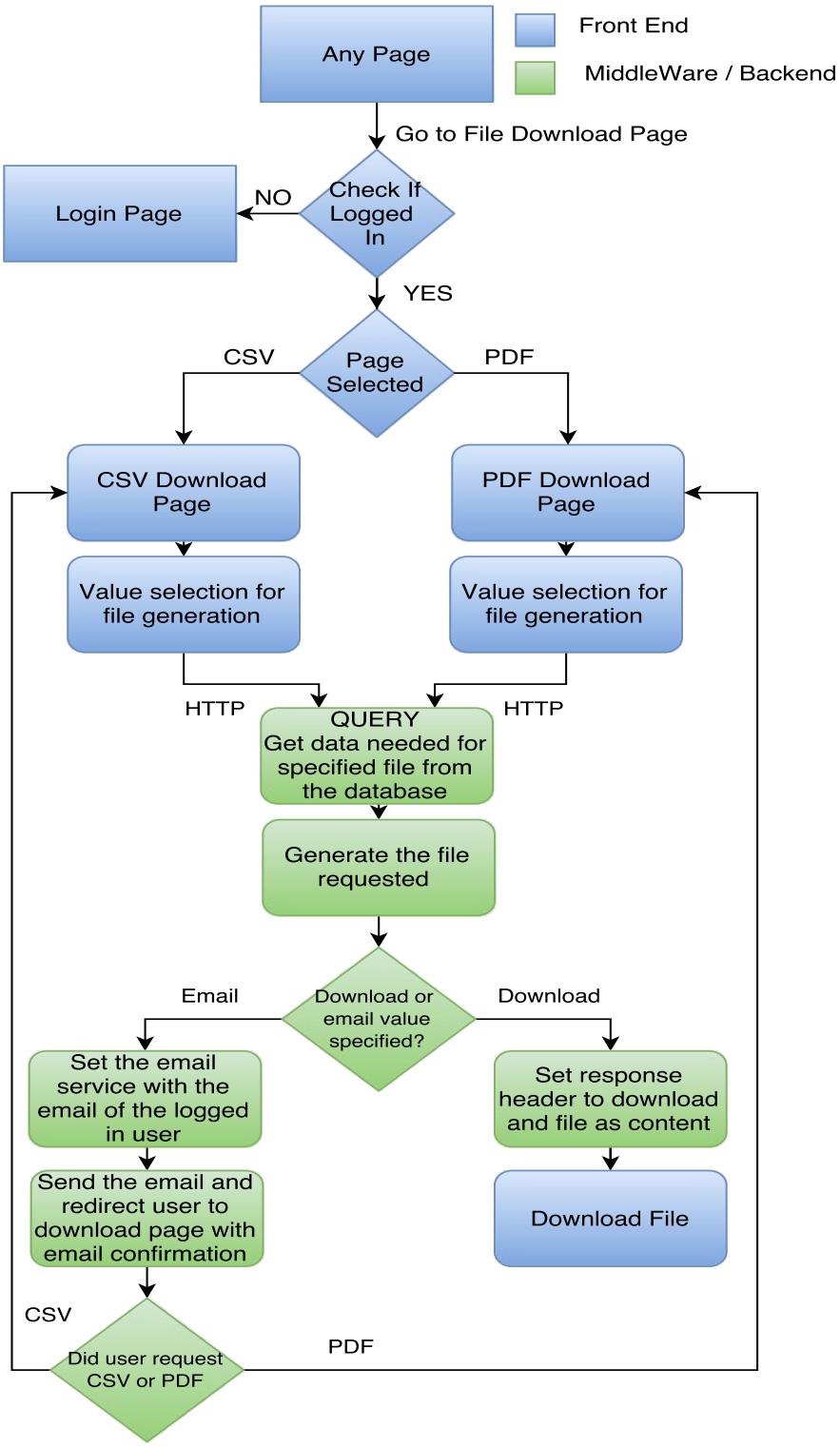


Figure 5.11: File Download Functionality Design

5.6.4 News

The news module of the system is responsible for providing users with the ability to create/read/edit/delete news posts according to each users access rights. The news posts are stored in the SQLite embedded database in the News table which consists of the post id, title, text(content) and date_Submitted columns.

Every time a user visits the new page an HTTP request gets made automatically to the web server requesting all the posts available, the web server then queries the database for all the available posts. The server returns a JSON object that contains all the news posts that were found in the database and the client populates the News view accordingly via the use of two way binding that AngularJS provides.

Non registered users and normal users can only read the posts on the new page. If a user is logged in as an administrator they are given the additional options to create a new post, edit a post and delete a post. All these actions are completed via the use of the web server API through HTTP requests. When an HTTP request is made for any of these actions the web server first makes sure that the request comes from an administrator user and then inserts, updates or deletes an entry from the SQLite database according to the request and the id of a new post. When adding or editing a post, the content of the request gets taken from the TextAngular editor through its two way binding feature, which provides the content of the editor in an HTML format[39]. By saving the content in the database in an HTML format, the posts are always rendered correctly on the browser without the need of any additional processing.

5.6.5 User Register / User Deletion

An administrator user is given the ability to register new users. In addition to that there is a single superuser in the system that has the ability to delete already registered users. This is done through two separate views the User Registration one and the User Deletion view.

User Registration

An administrator can access the user registration page. The page consists of some text entries where the administrator needs to enter the email of the user to be registered, a name for the user to be identified by, a password that can be changed afterwards and a flag that specifies if the new user will be a regular user or an administrator. When the values are entered and the user submits the form, an HTTP request is made to the web server where after checking if the user that made the request is in fact an administrator, goes on to add the new user into the database. This is done only after checking that the user isn't already registered as the the email column in the users table is unique. On success or error, the appropriate notification gets send to the user.

User Deletion

The system allows a single super user to delete registered users. The user deletion view requests a list of all non-superuser users that are available to be deleted via an HTTP request through the web server API. The front end then populates the view with all the available users; including their name, email and access rights.

The superuser can use a delete button to delete a specific user, when this is done the client sends an HTTP request to the server to delete the specific user who is identified by a unique identifier in the database. When the server receives the request it checks to see if the user that made the request is a super user, and then checks if the email of the user to be deleted and the unique identifier match with the one in the database. The server then removes the entry out of the SQLite user table.

The user deletion view asks the user multiple times if they want to delete the selected user in order to make sure mistake deletions do not happen.

5.7 Final Product

The website and the system as a whole at its final stages provides a user friendly and fully functional platform for rabies researchers. A user can generate and interact with data visualisations seamlessly through the data

explorer page. There is a plethora of static content for the user to get familiarised with the overall fight against rabies in Tanzania and with more specific aspects of the overall project. Administrators can register new users and registered users can log in to get access to protected areas of the web site. Registered users can download automatically generated PDF and CSV files or send them straight to their personal email. A fully functional news view is implemented where users can get information about the latest noteworthy events. Administrators can create/edit/delete new rich posts in the news page. Registered users can reset their password and change their password while logged in. A single super user can also delete registered users.

All this done in a simplistic user friendly manner, through a seamless and minimalistic interface which responds to a plethora of different and unique devices. In addition all this functionality is supplied by a system that is simple, highly expandable, fast and easy to develop. Furthermore all the functionalities take into consideration unstable Internet connections and outdated devices, by making sure that the bandwidth and processing load is at its minimum by transferring any heavy load to the system's back end.

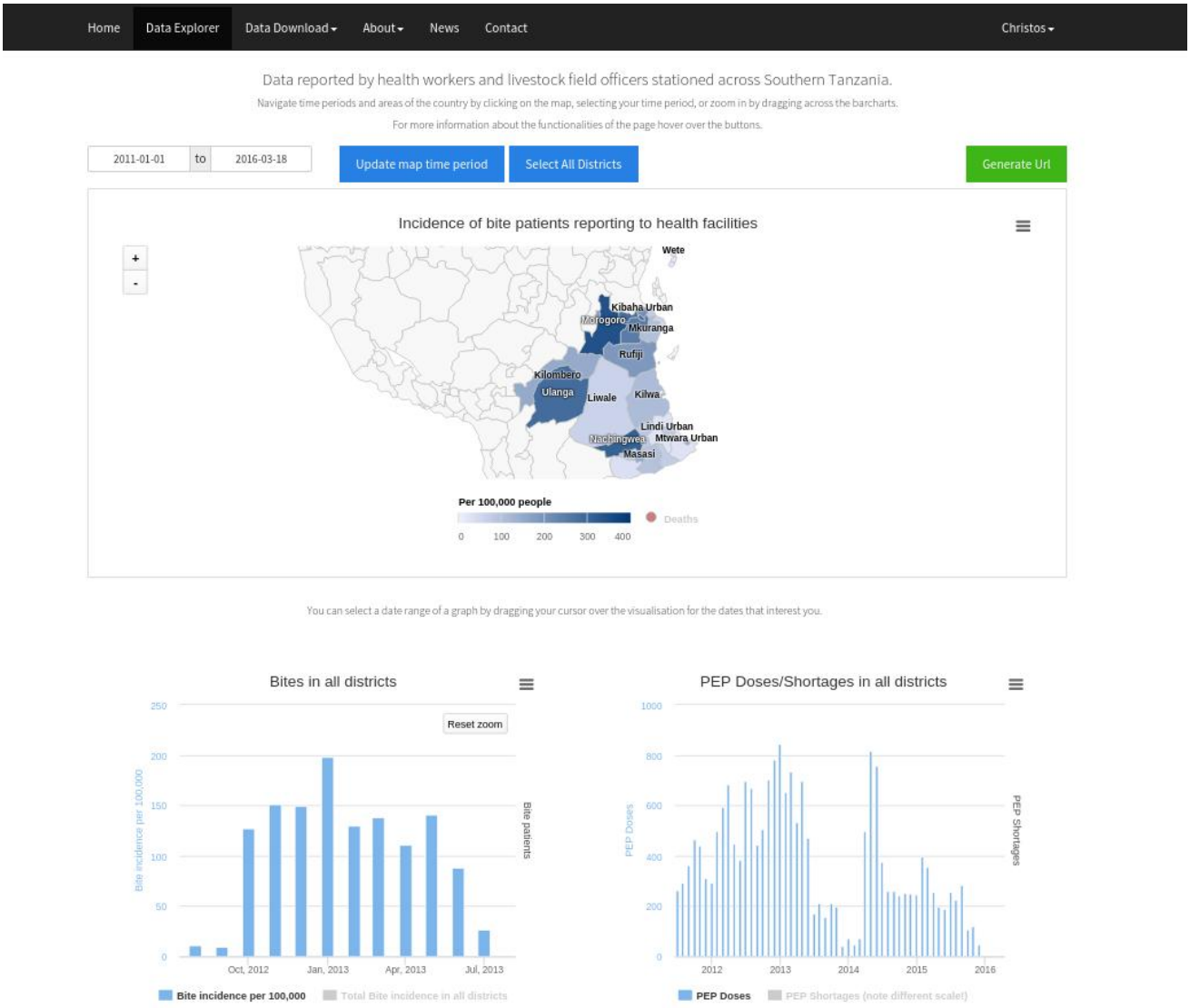


Figure 5.12: Final View of Data Explorer

5.8 Documentation

Due to the fact the system will continue to exist and be used after the end of the project; proper documentation had to be written.

Code Base

Inside the code base there are many comments explaining the functionality of each code snippet. In addition, easy to understand variable names were used for everything in order to make each variable and method self explanatory. Furthermore due to the fact that the code is separated into specific files such as the API file, a future developer will know where everything is located.

Overall System Documentation

Due to the size of the project, a documentation file was created that explains into detail the entire system. The document consists of details about the following :

- Back End/ Middleware
  - How to start/stop and deploy the server
  - How to install/Update modules in the server
  - Where every functionality of the system is located
  - Where all the static and non static content is located
  - How to write and modify API's
  - How to change the authedication strategy
  - How to modify and move the SQLite embedded database
- Client/Front End/Web Application
  - How to install and update libraries/modules
  - Where all the functionality code is located
  - How to change the application configuration
  - How to add new pages in the web application
  - How to change the HTML/JADE
  - How to use and modify HighCharts/HighMaps

5.9 Conclusion

In this chapter an in depth overview was given of all the technologies that were chosen for the development of the system. In addition descriptions were provided about the logic behind the main features that the final web site provides to its users.

Chapter 6

Evaluation and Testing

In this chapter the evaluation and testing strategies including their results will be described. The system had to be tested in multiple ways by taking into account the functional and non functional requirements. The main ways the testing and evaluation was done was, through heuristics and manual system testing, platform testing across multiple devices and technologies; and through an in depth user evaluation with real users to test all the aspects of the system.

6.1 System Testing

The overall system reliability and robustness was tested manually. That was done via structured real life tasks that had as an aim to test the systems expected responses and error handling. The real life tasks consisted of pushing the system to its limits and testing edge cases. This was done throughout the development process in order to discover errors and bugs. Things like trying to access protected pages without being logged in, SQL injections, passing illegal characters and variables in entries were tested often.

In addition more structured scenarios were made, these scenarios would label actions and their expected outputs. These scenarios covered all the functional requirements of the system. Some examples include :

- Test Scenario 1
    1. Go to the login page
    2. Enter email as username with a wrong password
    - Expected Response : Get an error message describing that a wrong email/password was entered
- Test Scenario 2
    1. Go to the home page
    2. Try to access the data download page without loging in
    - Expected Response : Get redirected to the login page

The system testing was done during and after the development of the system. The website was also put through stress and performance testing by third party software (loadimpact, pingdom and gtmetrix), in order to make sure that the server could handle large workloads without any problems and check if good web development practises were followed[44, 45, 46].The final results of these tests are available in the appendix.

6.2 Platform Testing

One of the non functional requirements of the front end application was to make it responsive to a vast array of new and outdated devices. These devices include desktops, laptops, smart phones and tablets.

Research was conducted about the most common devices used by potential users in Tanzania, where it was found that most people use android based smart phone devices marketed towards developing nations. The research was done via inspecting the most popular phone retailers in Tanzania, and interviews with people that lived there. In addition different screen sizes had to be taken under consideration. Due Europe's non commercial supply of mobile phones used in Tanzania; an unorthodox strategy had to be used.

The main differences that were identified between all devices were, the screen size, the operating system and the web browser used to access the website. For non mobile devises most of the resources for the testing were available, but for mobile devices various software tools had to be used as well as a limited but diverse supply of mobile devices.

6.2.1 Screen Size Testing

The Screen size of each individual device affects the way website is rendered on it. Screen sizes today can vary between a small smart phone to large monitor screens. As it can be expected the difference is substantial but its usually the small screens that get affected negatively. The website has been developed with responsiveness in mind thus it is expected to be rendered appropriately and automatically according to the device that is accessing it.

For the testing of screen sizes Google Chrome's built in developer tools were used, these tools allow a developer to render the website visited in a specific manner by specifying the device it wants to emulate or provide specific screen dimensions. This tests pure purpose was to find if the website renders acceptably on a plethora of different screen sizes and it did not test the actual functionality of the web site.

By testing the website on many screen sizes it is expected to gain feedback on how well the interface was developed and how to improve it further, in order to produce a truly responsive web site that will be available to everyone.

6.2.2 Cross Platform Functionality Testing

The website needs to be functional when it gets accessed from different devices. The main things that affect the functionality of a web application are the following :

- Input Method (Touch Screen/Mouse and keyboard)
- Operating System used (Android/Ios/Windows etc)
- Web browser used (Chrome/Firefox etc)

For the functionality testing the use of Google Chrome's developer tools was not enough as they cant replicate the exact device parameters needed. Thus the website was also tested on various devices, operating systems and web browsers. The testing strategy included the completion of some tasks that covered the main functional requirements of the system. The modules that were tested were the following.

- Data Explorer Interaction
- Login/Register/Forgot Password/Change Password
- News page (Read/Create/Edit)
- Downloading CSV/PDF file through the data download pages

The testing was done on the following combinations of devices/technologies. For the mobile devices the testing was done on new and older popular devices. In addition it was possible to get hold of two smart phones similar to the ones used in Tanzania. These two phones were particularly interesting as they were using customised Android OS and Android Browsers which could affect the functionality of the website.

- Personal Computers
  - OS : Windows, OS X, Linux (Arch Linux Distributions, Debian Based distributions)
  - Browsers : Chrome, Firefox, Safari, Opera, Internet Explorer
- Mobile Devices
  - OS : Android, IOs
  - Browsers : Android Browser, Chrome, Firefox, Safari, Opera
  - Special Phone Models : UMI Iron, XIAOMI RedMi Note

Through these tests it was possible to evaluate the web site's functionality when accessed from diverse devices. Testing things like unstable Internet connections though; was not possible from the United Kingdom. It is for these reasons the functionality testing of the system continued in the user evaluation.

6.3 User Evaluation

Due to the nature of the project and its direct involvement with various groups of stakeholders, it was decided to do the user evaluation with potential users of the website. This strategy would not only evaluate the website and its functionalities, but it would also test all the non functional requirements that were not possible to test with any other methods.

The user evaluation aimed to discover if all the functionalities of the website worked as expected, if the website was truly responsive when accessed through different devices, if it was light enough to be used through unstable and slow Internet connections and also identify future improvements and new functionalities. In addition it was expected to discover any potential problems.

In order to get the best results, the evaluation was carried in two separate ways. One was a think aloud strategy done with the main stake holders(administrators) of the project. This method provided insight not only in the functionalities but also the future maintenance and development plans of the system. The second part consisted of distributing a structured set of tasks that tested all the functionalities, to different groups of stakeholders. That was followed by a semi-structured questionnaire which rated all the aspects of the website, and requested future suggestions and improvements while at the same time pointing out any problems encountered in the website. The questionnaire also required the users to supply their operating system, browser and screen size through a third party website.

The structured task evaluation got distributed to numerous future users around the world and more importantly in Tanzania; in order to get the most accurate results. These users consisted of Researchers, Scientists, Government Officials and health workers. The tasks that were included in the evaluation were the following :

- Login Into the Website
- Find the csv download page
  - Download a raw data (csv) file for the period and district that interests the user the most
- Find the PDF report download page
  - Download a PDF report for the period, district and subject that interests the user the most
- Find the Data explorer page
  - Interact with the page and try to gain some information via the map and the charts
- Find the news page
  - Try to read the test news articles that are present
- Have a look around the website and try to gain more information about the project

After the users completed these steps they had the complete the questionnaire, that consisted of questions about the ease they achieved every task on a scale of 0-5(very easy - very hard). In addition questions about the look, speed and responsiveness had to be answered in a scale of 0-5. Further more open ended questions were asked about any problems they encountered, and recommendations they might have. Lastly the user had to answer multiple choice and open ended questions about future additional functionalities that they would like to see in the web site. The questionnaire was made with Google forms and can be found in the appendix.

6.4 Evaluation Results

The goal of the evaluation was to gain feedback on the existing product and its functionalities, while at the same time discovering new and interesting potential future expansions for it. This section will outline the results of the evaluation strategies that were conducted. The results from the user evaluation were gathered through the questionnaire that was provided and via direct emails, while the evaluation with the main stakeholders results were gathered by taking notes during the think aloud interview.

6.4.1 User Evaluation

The were nine users that completed the questionnaire, four of them from Tanzania and four from Scotland. Out of the nine people, six were rabies researchers and the rest were academics. The majority of them used Chrome as a web browser for the completion of the tasks, and their devices ran mainly the Windows Operating System; there was also one user that used OSX. It was interesting to see that the the majority of users were using Windows 7, which was released in 2009. In addition the screens the users used varied a lot, with the majority being around 1366 x 768 which is quite small for today's standards, and ranging up to 1920 x 1080.

All the users except one, managed to complete the tasks successfully without encountering any major problems. One user did not manage to download a PDF report in a proper format, according to his feedback the PDF downloaded had a .txt extension at the end; that was fixed by modifying the response headers. In addition the

users provided mainly positive feedback about how easy it was to use the website as it can be seen bellow, with most users providing the best rating in a scale of 0 to 5.

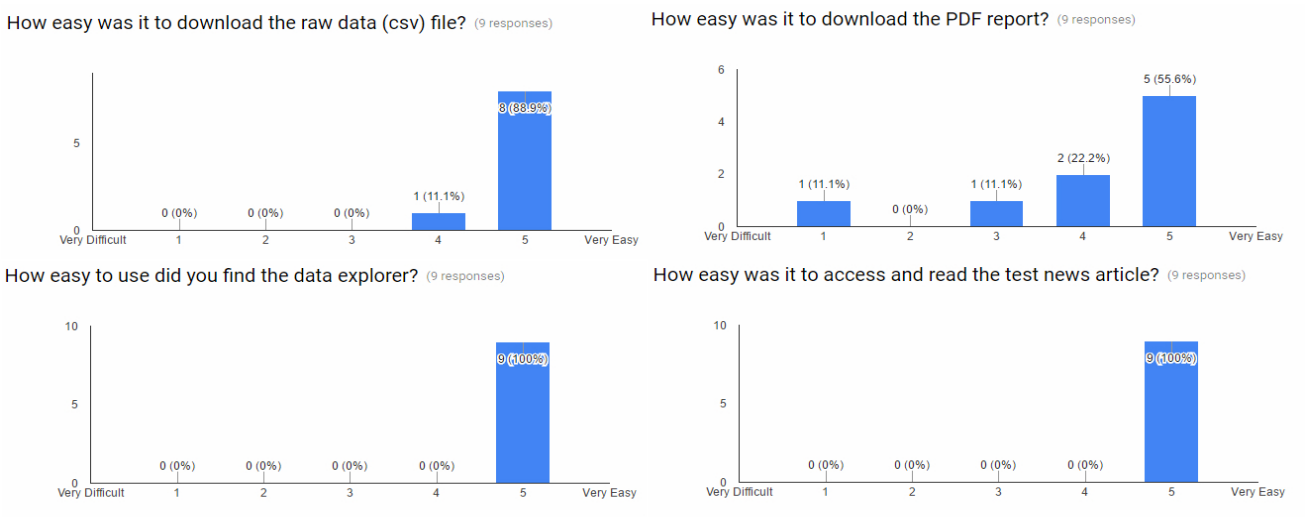


Figure 6.1: First part of the Questionnaire Results

When users were asked how fast they found the website and the response times, the feedback was mainly positive. This was mainly due to the slow Internet connections, and also due to the fact that during the time of the user evaluation the website was hosted on a free server environment in North American. Users were still able to download the files, with one exception, where it took too long for a specific user to download a PDF report and he had to try a second time. The second time the PDF file downloaded successfully.

Users responded averagely when they were asked about the look and the feel of the website, users mentioned that it felt a bit too empty. They suggested the addition of images and other media, which was avoided until now due to the requirement of keeping the website lightweight.

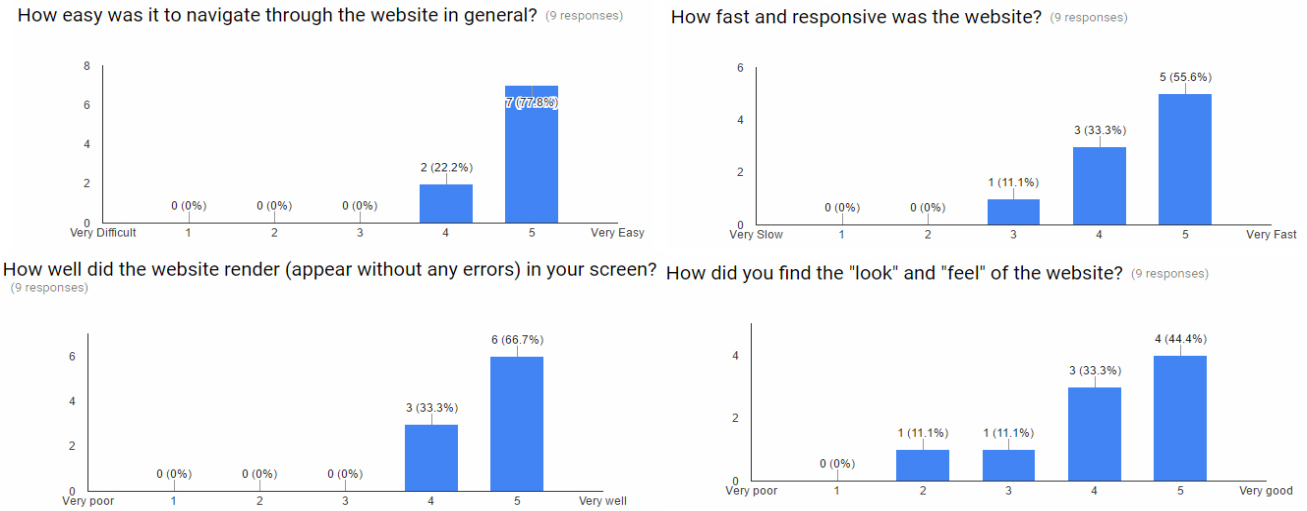


Figure 6.2: Second part of the Questionnaire Results

Users were also asked if the would like any additional future functionalities, and to provide open ended feedback . The majority of users mentioned that they would be interested in automated emails with PDF reports,

and one user was interested in automated SMS messages. The requested automated reports consisted mainly of the ones available through the website. A single user also mentioned that he would like to receive automated alerts for important events, like rabies related deaths. Futher more most users requested that they would like to receive automated information about various data that is presented in the Data Explorer, such as : Bite exposures, PEP shortages, Dog vaccination campaigns, Suspected outbreaks and Samples collected and laboratory results.

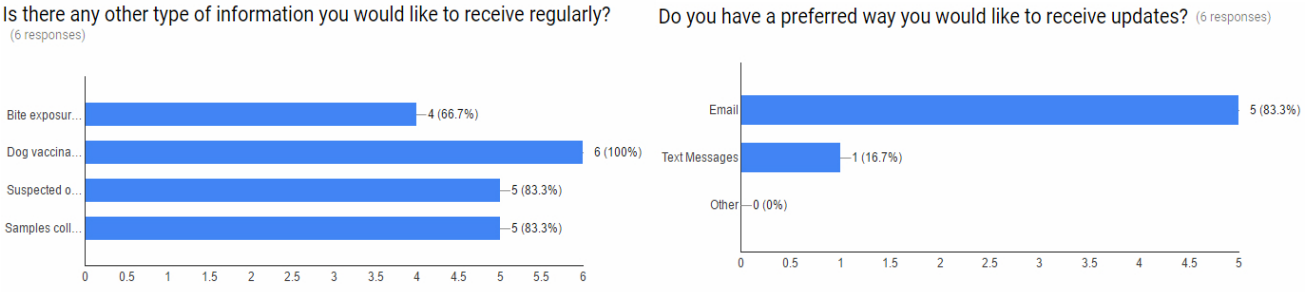


Figure 6.3: Third part of the Questionnaire Results

Overall the feedback received from the user evaluation was mainly positive. The users provided honest answers and described all the problems they encountered clearly. The majority of the problems that were identified were resolved shortly after the end of the user evaluation. In addition multiple suggestions were made regarding the future of the website.

6.4.2 Main Stakeholders Evaluation

An evaluation was carried with the main stakeholders of the project. It consisted of a think aloud interview, the stake holders were given a semi structured set of tasks to achieve, and the stakeholders were encouraged to discuss their thoughts through the whole process. The results from this evaluation were really beneficial. The stakeholders did not encounter any major difficulties during the evaluation and a lot of feedback was provided about small interface improvements. In addition some small changes on the functionality of the website were requested; such as the ability to see a users access levels in the delete user view. These suggestions were implemented after the end of the evaluation. Further more during the evaluation the process of passing the project and the code base to them was discussed along with other administrative issues such as: the creation of an administrative email, creation of a joint GitHub account for future developers, documentation and migration to Amazon AWS.

Both clients felt that the final product met all main expectations that were identified since the start of the project and that it covered all the functionalities that were requested. The overall responses to the product were highly enthusiastic and the main stakeholders were happy about the outcome, (Katie)”I am impressed, I think it looks great”. The overall feedback that was received was positive and both the stakeholders were impressed by the final product.

6.5 Conclusion

This chapter went through all the testing and evaluation that was done, during and after the development of the product. The overall results are positive and the product meets the majority of the requirements successfully. In addition useful feedback and suggestions was gathered that can be used in the future, for the expansion of this project.

Chapter 7

Future

In this chapter, possible future work and improvements are described in detail. This includes partially unfinished functional requirements that weren't able to be completed due to infrastructure or time limitations, additional possible functinalities that were identified during the development process and recommendations from the participants of the user evaluation.

7.1 Automated Email Reports

Automated Emails was a new functionality that was suggested by both users and the main stakeholders. These emails could be similar to the PDF report emails that are dynamically generated in the web server and could be sent automatically to all registered users in specific time intervals such as once a month. This will provide users with an update on the progress that was made in the past month which a lot of users would find invaluable.

Due to the fact that an email sending service is already present in the system, the development of automated emails would be quite simple as it would mainly require the development of a timed invocation of the email service. NodeJs supports multiple ways of implementing timer functions[27]. In addition the email service supports the ability to send HTML as emails, which might be a preferable option than just generating and attaching a PDF file.

In addition, to time based automated emails, another option would be sending emails according to specific important events regardless of time. Things like rabies related human deaths are extremely important and it would be useful to alert health workers and researchers about it immediately through an email[24].

7.2 SMS Service

Another possible future capability that was requested was the ability to send SMS messages through the system. These SMS messages could provide automated short reports like the ones mentioned before. These could consist of the progress that was made in the past month or so and provide some useful data. The messages could also be sent according to important events like a rabies inflicted death in a specific district.

Currently there is no infrastructure that supports SMS sending in the system, there are though multiple ways of developing such service. A possible solution is the use of a third party service such as Twilio which provides an API that the web server would be able to use to send SMS texts. One of the possible implications with Twilio or

other similar software, is the fact that their service is not free[47]. On top of that, the costs of using these services would be extremely high due to the fact that the majority of text messages would be sent to people in Tanzania. Another more complex but much cheaper viable solution would be to use a specialised router device that has a SIM card slot, and place it in Tanzania with a cheap mobile provider SIM card[48]. With that infrastructure in place the web server could connect to that specific device remotely and send SMS messages through it.

7.3 Server based HighCharts/HighMaps visualisations

HighCharts and HighMaps have the ability to be generated on the server side instead of the client. By generating the visualisations in the back end, the server will be able to send automated emails and reports that include the relevant maps and graphs. This would provide more content rich emails and reports which a lot of people would find more useful[37].

In order to develop this functionality the server will have to include PhantomJS which is a Javascript framework that can run independently without the need of a browser (similar to NodeJS)[49, 15, 37]. With the use of PhantomJS the creation of the visualisations would be the same as they are in the front end currently so it would not require any extensive additional development[37].

7.4 Extended Map Visualisation

Currently the map visualisation in the data explorer presents the 28 districts which the project is involved with. A future capability would be to present a more in depth map visualisation by providing a drill down map. A drill down map is a map where if a district gets selected the map ”drill downs” to that district and generates a new visualisation at a lower level, such as counties. This would allow the user to get data about more specific areas that might be of interest, other than a district which consists of a large land area.

HighMaps supports the implementation of drill down maps and is configured similarly to the already existing map visualisation. One of the obstacles of this new functionality is the supply of appropriately made Maps for the visualisation. HighMap requires a separate in depth map file for each district, and these map files will require to have the same format as the one used currently[37].

7.5 Additional Visualisations and File Types

Another possible addition that could be made to the website, and was partially requested by users; was the implementation of additional visualisations in the data explorer. The addition of new visualisations would be simple to implement, as they would not be much different than the ones already existing, except from the data supplied and the HighCharts configuration[37].

Further more some different types of files were requested, such as the ability to download documents in a .odt format (Word format). For this an additional technology for the generation of Microsoft Office file formats will have to be used. The most popular technologies that meet this functionality is ”officegen” and ”officegen-2”, both are available in NPM which means installation will be simple. The implementation of this functionality would mainly consist of the configuration needed for the generating the Word document, as the rest of the logic for downloading and querying the data is already present in the web server[18, 50].

Chapter 8

Outcome and Conclusion

8.1 Summary

The objective of this project was to develop an interactive data visualisation platform for rabies researchers, while at the same time creating a system with expandability and long term maintainability in mind. The final product met all the project requirements that were identified in the second chapter of this dissertation. Through the expansive testing and evaluation that the product went through it was demonstrated that the application is simple and user friendly, while at the same time it provides rich and informative content. A lot of possible future work was also identified which is possible to develop with the current state of the overall system. The website has reached the point were its entirely usable and it is accessible for users to gather, share and visualise data for the support of the fight against rabies in Southern Tanzania.

8.2 Evaluation Analysis

The analysis of the user evaluation provided a major insight on the state of the system, and how it could be further expanded to cover additional needs of researchers. It was revealed that the current functionalities of the web site work really well, while at the same time it was proved that it is made in such a way to support unstable connections and diverse devices acceptably. A lot of future work was also provided, such as the induction of automated emails and text messages through the server for important events. In addition through the interview evaluation it was found that the main stakeholders of the project were impressed with the final product. Overall the feedback received was highly positive and extremely useful in terms of additional improvements.

8.3 Reflection

The project was an extremely rewarding experience. The opportunity to develop a web site from scratch and the support provided by the project supervisor and the main stakeholders, provided me enough freedom to make the main design choices and experiment with new technologies. The constant interaction with the stakeholders and the fact that the product was of such importance to real people, made this project feel truly significant. Through the development I was exposed to technologies that I've never encountered before, and I developed my skills in every aspect of software engineering. Currently I am confident in my web development skills and I think that I could continue doing it for non academic purposes. After all this, I hope that this website will be truly helpful

in the fight against rabies and that it will end up accelerating the complete elimination of the disease in Southern Tanzania.

8.4 Acknowledgements

I would like to thank:

- My project supervisor Simon Rogers for his guidance and for providing so much time to the project with all the longer than expected meetings.
- The main stakeholders of the project Katie Hampson and Rebecca Mancy who were present at every weekly meeting and for responding to my countless emails.
- All the participants of the user evaluation for providing constructive feedback and extremely useful recommendations.

8.5 Website Link

Interactive Website For Rabies Researchers

Bibliography

[1] rabies. http://www.nhs.uk/Conditions/Rabies/.

[2] Who rabies. http://www.who.int/rabies/en/.

[3] Maganga Sambo, Sarah Cleaveland, Heather Ferguson, Tiziana Lembo, Cleophas Simon, Honorati Urassa, and Katie Hampson. The burden of rabies in tanzania and its impact on local communities. 2013.

[4] Louise H. Taylor, Katie Hampson, Anna Fahrion, Bernadette Abela-Ridder, and Louis H. Nel. Difficulties in estimating the human burden of canine rabies. 2015.

[5] Jo Halliday, Chris Daborn, Harriet Auty, Zacharia Mtema, Tiziana Lembo, Barend M. deC. Bronsvoort, Ian Handel, Darryn Knobel, Katie Hampson, and Sarah Cleaveland. Bringing together emerging and endemic zoonoses surveillance: shared challenges and a common solution. 2012.

[6] Who bill & melinda gates foundation fund who-coordinated project to control and eventually eliminate rabies in low-income countries. http://www.who.int/rabies/bmgf_who_project/en/.

[7] Kevin Bardosh, Maganga Sambo, Lwitiko Sikana, Katie Hampson, and Susan C. Welburn. Eliminating rabies in tanzania? local understandings and responses to mass dog vaccination in kilombero and ulanga districts. 2014.

[8] Zacharia Mtema, Joel Changalucha, Sarah Cleaveland, Martin Elias, Heather M Ferguson, Jo E B Halliday, Daniel T Haydon, Gurdeep Jaswant, Rudovick Kazwala, Gerry Killeen, Tiziana Lembo, Kennedy Lushasi, Alpha David Maligana, Rebecca Mancy, Matthew Maziku, Eberhard M Mbunda, Geofrey Mchau, Rod Murray-Smith, Kristyna Rysava, Khadija Said, Maganga Sambo, Elizabeth Shayo, and Lwitiko Sikana. Mobile phones as surveillance tools: implementing and evaluating a large-scale intersectoral surveillance system for rabies in tanzania. 2016.

[9] Prosper P Chak, Yeromin Mlacha, Daniel Msellemu, Athuman Muhili, Alpha D Malishee, Zacharia J Mtema, Samson S Kiware, Ying Zhou, Neil F Lobo, Tanya L Russell, Stefan Dongus, Nicodem J Govella, , and Gerry F Killeen. An affordable, quality-assured community-based system for high-resolution entomological surveillance of vector mosquitoes that reflects human malaria infection risk patterns. 2012.

[10] Git wiki. https://git.wiki.kernel.org/index.php/GitSvnComparison.

[11] Github. https://github.com/.

[12] Heroku. https://www.heroku.com/.

[13] Openshift. https://www.openshift.com/.

[14] Amazon aws. https://aws.amazon.com/?nc2=h_lg.

[15] Nodejs. https://nodejs.org/en/.

[16] Expressjs. http://expressjs.com/.

[17] Django. https://www.djangoproject.com/.

[18] Npm. https://www.npmjs.com/.

[19] Sqlite. https://www.sqlite.org/.

[20] Mongodb. https://www.mongodb.com/.

[21] Passport. http://passportjs.org/.

[22] Stormpath. https://stormpath.com/.

[23] Express-mailer. https://github.com/RGBboy/express-mailer.

[24] Nodemailer. https://github.com/nodemailer/nodemailer.

[25] Pdfkit. http://pdfkit.org/.

[26] Pdfmake. http://pdfmake.org.

[27] Node-csv. https://github.com/wdavidw/node-csv.

[28] Jsontocsv. https://github.com/mrodrig/json-2-csv.

[29] Node-mysql. https://github.com/felixge/node-mysql.

[30] Node-sqlite3. https://github.com/mapbox/node-sqlite3.

[31] Jade. http://expressjs.com/.

[32] Angularjs. https://angularjs.org/.

[33] Developer economics. http://www.developereconomics.com/comparison-4-popular-javascript-mv-frameworks-part-2/.

[34] Angular modules. http://ngmodules.org/.

[35] Jquery. https://jquery.com/.

[36] Bootstrap. http://getbootstrap.com/.

[37] Highcharts. http://www.highcharts.com/.

[38] D3. https://d3js.org/.

[39] Textangular. https://github.com/fraywing/textAngular.

[40] Bootstrap datepicker. https://bootstrap-datepicker.readthedocs.org/en/latest/.

[41] Bootstrap3-dialog. https://github.com/nakupanda/bootstrap3-dialog.

[42] Urs Ramer. An iterative procedure for the polygonal approximation of plane curves. 1972.

[43] Visvalingam Maheswari and J.D. Whyatt. Line generalisation by repeated elimination of the smallest area. 1992.

[44] Pingdom. http://tools.pingdom.com/fpt/.

[45] Gtmetrix. https://gtmetrix.com/.

[46] Loadimpact. https://loadimpact.com/.

[47] Twillio. https://www.twilio.com/.

[48] Developers home. http://www.developershome.com/sms/howToSendSMSFromPC.asp.

[49] Phantomjs. http://phantomjs.org/.

[50] Officegen. https://github.com/Ziv-Barber/officegen.

Appendices

Appendix A

API Endpoints

File Name	Type
/api/districtDeathsPerPop/	GET
/api/districtIncidentsByMonth/	GET
/api/districtIncidents/	GET
/api/PEPDosesByMonth/	GET
/api/PEPShortagesByMonth/	GET
/api/totalDogsVaccinated/	GET
/api/dogsVaccinatedPerDistrict/	GET
/api/villageVaccinationCampaigns/	GET
/api/villageVaccinationCampaignsPerDistrict/	GET
/api/suspectAnimalsPerDistrict/	GET
/api/sampleResultsPerDistrict/	GET
/api/totalSampleResults/	GET
/api/csvDownload/	GET
/api/PDFReport/	GET
/api/posts/	GET
/api/post/:id	GET
/api/post/	POST
/api/post/:id	PUT
/api/post/:id	DELETE

/register/	POST
/resetPassword/	POST
/api/getUsers/	GET
/api/getUser/:id	GET
/api/deleteUser/:id	DELETE
/forgot/	POST
/reset/:token	GET
/resetPage/:resetPasswordToken/:resetPasswordExpires	POST

Table A.1: API Endpoints

Appendix B

User Evaluation Form

Rabies Surveillance in southern Tanzania website evaluation

Hello and thank you for participating in the test user evaluation. This website has been designed as a platform to share and visualise data and progress from the rabies surveillance project in Southern Tanzania. It has been designed to work effectively in areas with limited internet connectivity.

The evaluation will consist of you (the test user) doing the following:

- Complete a structured set of tasks in order to test some of the core functionalities of the website
- Answer a short questionnaire about the tasks you tried to achieve in order to provide feedback about the functionalities the content and the quality of the website

Tasks to achieve

1. Go to the website http://rabieswebsite-rabiesresearch1.rhcloud.com/
2. Log into the website with the following credentials
    a. Username: ***test@test.com***
    b. Password: ***test***
3. Find the csv download page
    a. Download a raw data (csv) file for the period and district that interests you the most (Please do not click the "Send Email" button)
4. Find the PDF report download page
    a. Download a PDF report for the period, district and subject that interests you the most (Please do not click the "Send Email" button)
5. Find the Data explorer page
    a. Interact with the page and try to gain some information via the map and the charts
6. Find the news page
    a. Try to read one of the test news articles that are present
7. Have a look around the website and try to gain more information about the project

Answer the questionnaire

1. Go to http://www.whatsmybrowser.org/ and copy the link provided as you will need it for the questionaire. The link is in the box displayed below

2. Go to http://goo.gl/forms/yNM8bNyzmr and answer the questionaire

Thank you for your participation

If you have any questions regarding the evaluation please contact:
chr.kitsos@gmail.com

Appendix C

User Evaluation Questionnaire

Website Evaluation Form

Evaluation form for the Rabies Surveillance in southern Tanzania website

* Required

1. **Please supply your personal email address** *

.....

2. **Which group of users are you part of?** *
Mark only one oval.

- ◯ Health Worker
- ◯ Government Official
- ◯ Veterinarian
- ◯ Research Scientist
- ◯ Other

3. **Please provide the link with the details of your browser** *
Visit http://www.whatsmybrowser.org/ and copy the URL link provided

.....

4. **Did you manage to log in into the website?** *
Mark only one oval.

- ◯ Yes
- ◯ No

5. **How easy was it to download the raw data (csv) file?** *
Mark only one oval.

	1	2	3	4	5	
Very Difficult	◯	◯	◯	◯	◯	Very Easy

6. **Is there any other type of data you would like to be able to download?**

.....

.....

.....

.....

.....

7. **How easy was it to download the PDF report?** *
Mark only one oval.

	1	2	3	4	5	
Very Difficult	◯	◯	◯	◯	◯	Very Easy

8. **Is there any other type of PDF reports you would like to be able to download?**

.....

.....

.....

.....

.....

9. **How easy to use did you find the data explorer?** *
Mark only one oval.

	1	2	3	4	5	
Very Difficult	◯	◯	◯	◯	◯	Very Easy

10. **Is there any other type of maps/charts you would like to be able to explore?**

.....

.....

.....

.....

.....

11. **How easy was it to access and read the test news article?** *
Mark only one oval.

	1	2	3	4	5	
Very Difficult	◯	◯	◯	◯	◯	Very Easy

12. **How easy was it to navigate through the website in general?** *
Mark only one oval.

	1	2	3	4	5	
Very Difficult	◯	◯	◯	◯	◯	Very Easy

13. **How fast and responsive was the website?** *
Mark only one oval.

	1	2	3	4	5	
Very Slow	◯	◯	◯	◯	◯	Very Fast

14. **How well did the website render (appear without any errors) in your screen?** *
Mark only one oval.

	1	2	3	4	5	
Very poor	◯	◯	◯	◯	◯	Very well

15. **How did you find the "look" and "feel" of the website?** *
Mark only one oval.

	1	2	3	4	5	
Very poor	◯	◯	◯	◯	◯	Very good

16. **Please list bellow any problems you encountered during the evaluation of the website.**

.....

.....

.....

.....

.....

17. **Would you be interested to receive automatic updates about the information reported through the surveillance system**
Check all that apply.

- [ ] Emails summarising numbers of persons bitten in a specific area, number of death cases, and numbers of dogs vaccinated during the last month?
- [ ] Text messages summarising numbers of persons bitten in a specific area, number of death cases, and numbers of dogs vaccinated during the last month?

18. **Is there any other type of information you would like to receive regularly?**
Check all that apply.

- [ ] Bite exposures and PEP shortages
- [ ] Dog vaccination campaigns
- [ ] Suspected outbreaks
- [ ] Samples collected and laboratory results

19. **Do you have a preferred way you would like to receive updates?**
Check all that apply.

- [ ] Email
- [ ] Text Messages
- [ ] Other

20. **Please list your preferred ways of receiving updates. (Answer only if you chose "other" in the previous question)**

.....

21. **Is there any other type of information you would like to receive regularly that haven't been mentioned? Please elaborate.**

.....

.....

.....

.....

.....

22. **Please provide any general feedback/suggestions on how to improve the website**

.....

.....

.....

.....

.....

Appendix D

Consent Form

Department of Computing Science
University of Glasgow

Ethics checklist form for 3rd/4th/5th year, MSc IT/CS/ACS projects

This form is only applicable for projects that use other people ('participants') for the collection of information, typically in getting comments about a system or a system design, getting information about how a system could be used, or evaluating a working system.

If no other people have been involved in the collection of information, then you do not need to complete this form.

If your evaluation does not comply with any one or more of the points below, please submit an ethics approval form to the Department Ethics Committee.

If your evaluation does comply with all the points below, please sign this form and submit it with your project.

---

1. Participants were not exposed to any risks greater than those encountered in their normal working life.
   Investigators have a responsibility to protect participants from physical and mental harm during the investigation. The risk of harm must be no greater than in ordinary life. Areas of potential risk that require ethical approval include, but are not limited to, investigations that occur outside usual laboratory areas, or that require participant mobility (e.g. walking, running, use of public transport), unusual or repetitive activity or movement, that use sensory deprivation (e.g. ear plugs or blindfolds), bright or flashing lights, loud or disorienting noises, smell, taste, vibration, or force feedback

2. The experimental materials were paper-based, or comprised software running on standard hardware.
   Participants should not be exposed to any risks associated with the use of non-standard equipment: anything other than pen-and-paper, standard PCs, mobile phones, and PDAs is considered non-standard.

3. All participants explicitly stated that they agreed to take part, and that their data could be used in the project.
   If the results of the evaluation are likely to be used beyond the term of the project (for example, the software is to be deployed, or the data is to be published), then signed consent is necessary. A separate consent form should be signed by each participant.

   Otherwise, verbal consent is sufficient, and should be explicitly requested in the introductory script.

4. No incentives were offered to the participants.
   The payment of participants must not be used to induce them to risk harm beyond that which they risk without payment in their normal lifestyle.

5. No information about the evaluation or materials was intentionally withheld from the participants.
   Withholding information or misleading participants is unacceptable if participants are likely to object or show unease when debriefed.

6. No participant was under the age of 16.
   Parental consent is required for participants under the age of 16.

7. No participant has an impairment that may limit their understanding or communication.
   Additional consent is required for participants with impairments.

8. Neither I nor my supervisor is in a position of authority or influence over any of the participants.
   A position of authority or influence over any participant must not be allowed to pressurise participants to take part in, or remain in, any experiment.

9. All participants were informed that they could withdraw at any time.
   All participants have the right to withdraw at any time during the investigation. They should be told this in the introductory script.

10. All participants have been informed of my contact details.
    All participants must be able to contact the investigator after the investigation. They should be given the details of both student and module co-ordinator or supervisor as part of the debriefing.

11. The evaluation was discussed with all the participants at the end of the session, and all participants had the opportunity to ask questions.
    The student must provide the participants with sufficient information in the debriefing to enable them to understand the nature of the investigation.

12. All the data collected from the participants is stored in an anonymous form.
    All participant data (hard-copy and soft-copy) should be stored securely, and in anonymous form.

Project title Interactive Website for Rabies Researchers

Student's Name Christos Kitsos

Student's Registration Number 2023761

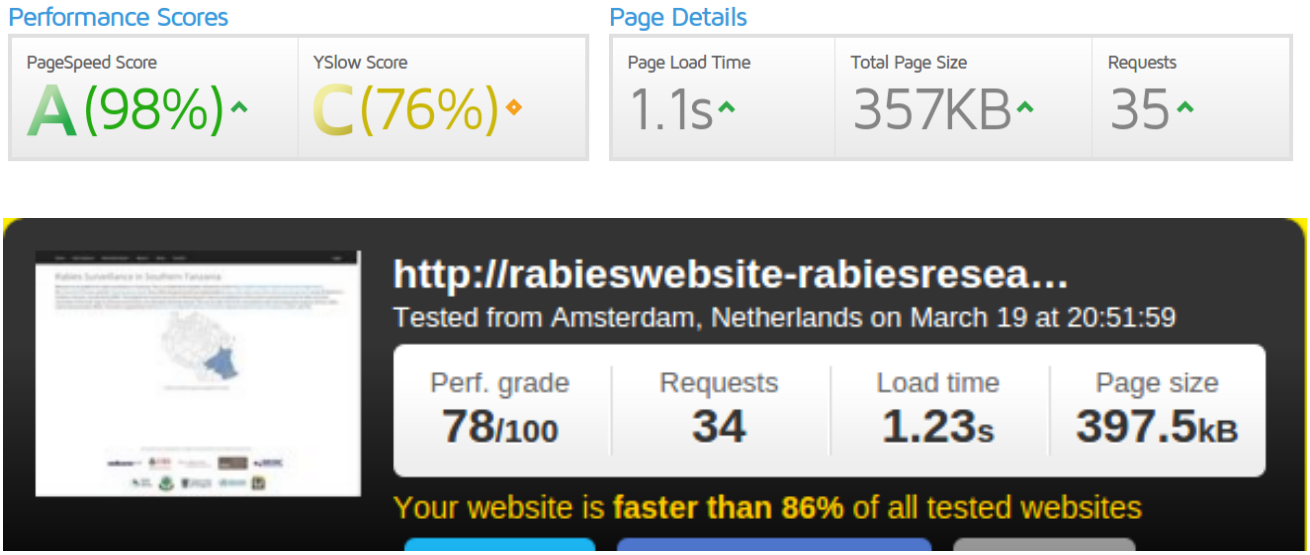
Student's Signature 

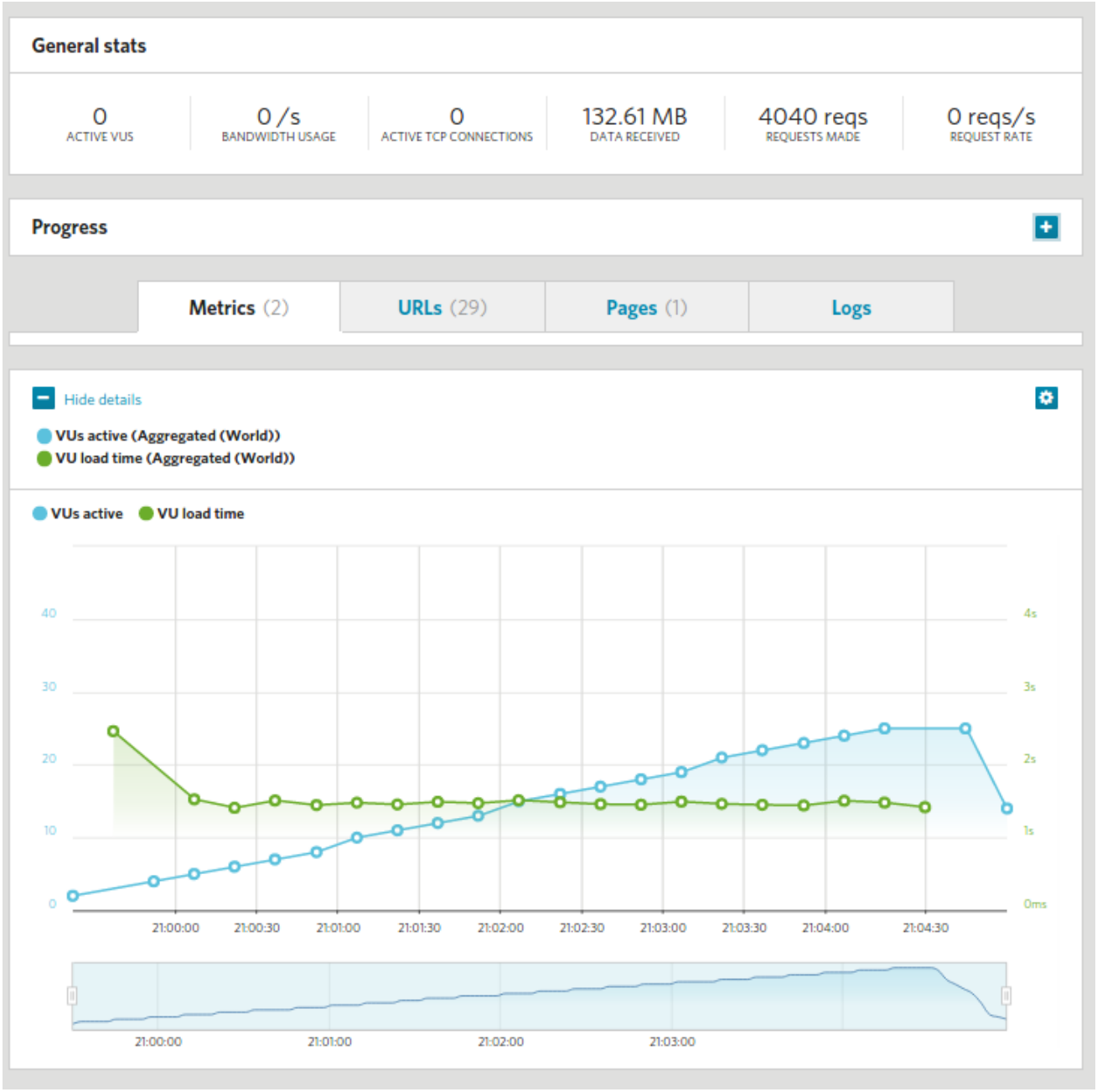
Supervisor's Signature

Date 29/3/16

Appendix E

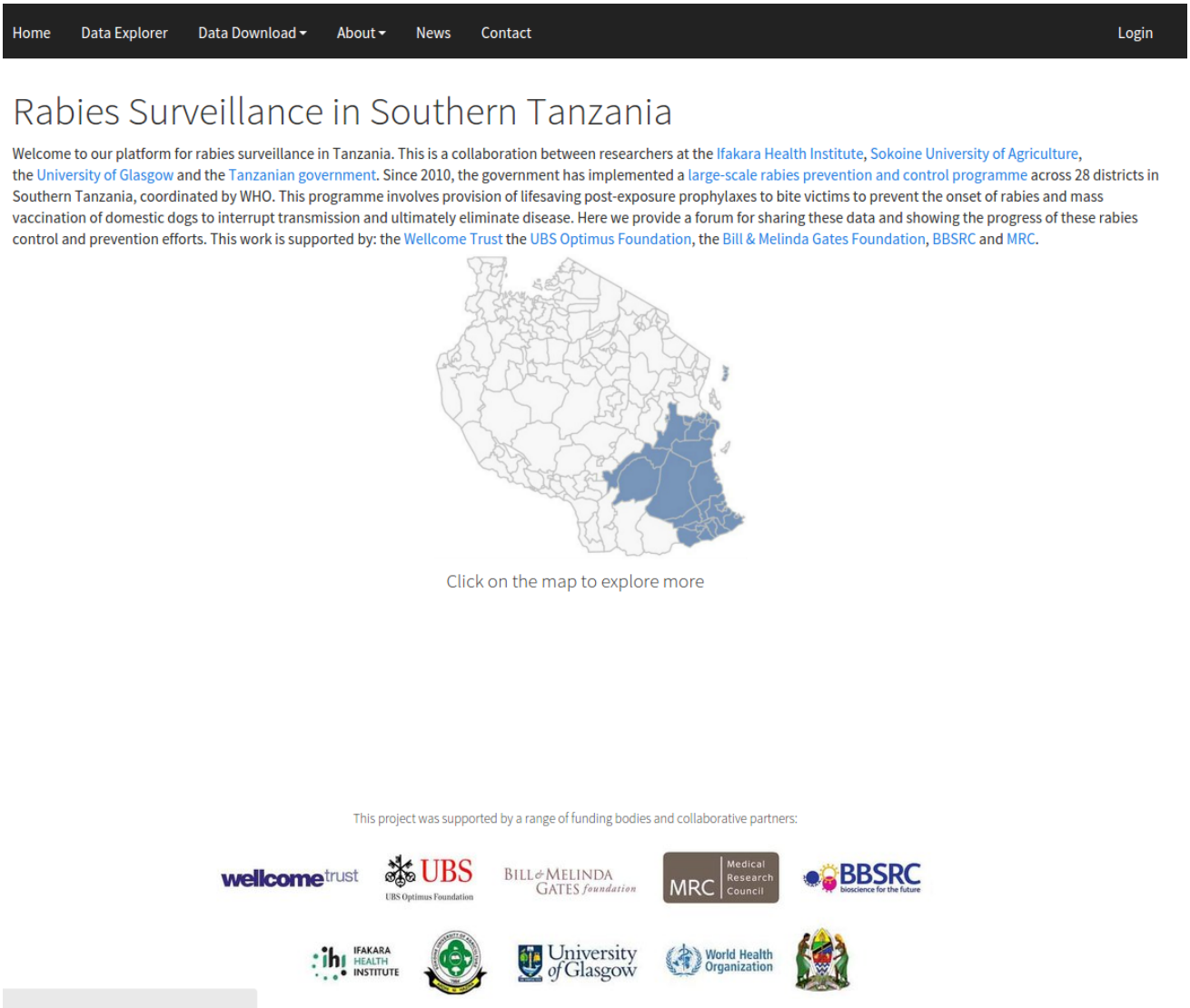
Speed and Performance Test Results

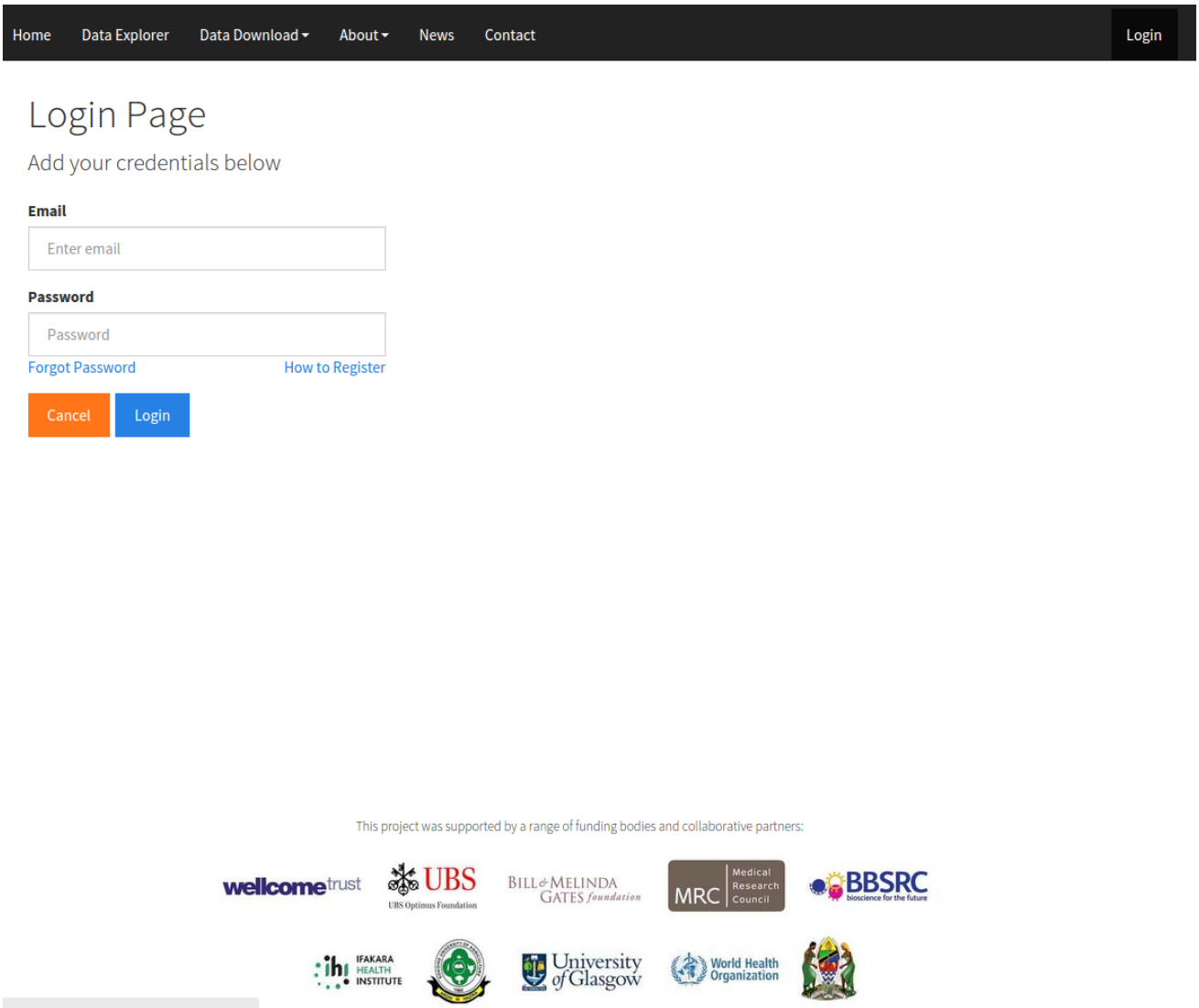



General stats
0
ACTIVE VUS
0 /s
BANDWIDTH USAGE
0
ACTIVE TCP CONNECTIONS
132.61 MB
DATA RECEIVED
4040 reqs
REQUESTS MADE
0 reqs/s
REQUEST RATE
Progress
Metrics (2)
URLs (29)
Pages (1)
Logs
Hide details
VUs active (Aggregated (World))
VU load time (Aggregated (World))
VUs active
VU load time
40
30
20
10
0
4s
3s
2s
1s
0ms
21:00:00
21:00:30
21:01:00
21:01:30
21:02:00
21:02:30
21:03:00
21:03:30
21:04:00
21:04:30
21:00:00
21:01:00
21:02:00
21:03:00

Appendix F

Interface Screenshots




Home
Data Explorer
Data Download
About
News
Contact
Login
Login Page
Add your credentials below
Email
Enter email
Password
Password
Forgot Password
How to Register
Cancel
Login
This project was supported by a range of funding bodies and collaborative partners:
wellcome trust
UBS
UBS Optimus Foundation
BILL & MELINDA GATES foundation
MRC
Medical Research Council
BBSRC
bioscience for the future
IFAKARA HEALTH INSTITUTE
University of Glasgow
World Health Organization

Home Data Explorer Data Download About News Contact Christos

Partners

The project partners include:

- IFAKARA HEALTH INSTITUTE
- University of Glasgow
- World Health Organization
- Sokoine University of Agriculture
- Government of Tanzania / Tanzania Ministry for Health and Social Welfare / Tanzania Ministry for Livestock and Fisheries Development

Funding

The project is supported by:

- wellcome trust
- BILL & MELINDA GATES foundation
- UBS Optimus Foundation
- BBSRC bioscience for the future
- MRC Medical Research Council

This project was supported by a range of funding bodies and collaborative partners:

wellcome trust UBS Optimus Foundation BILL & MELINDA GATES foundation MRC Medical Research Council BBSRC bioscience for the future

International projects category: award winner and runners up

A University of Glasgow research team has developed a system that allows health and veterinary workers to upload reports of rabies through mobile phones

Rabies is the world's most deadly disease: transmitted by domestic dog bites, it is inevitably fatal once symptoms occur and kills 59,000 people every year in developing countries. It circulates in regions with weak health infrastructure and limited resources, affecting the poorest and most vulnerable communities. Yet, all of the tools needed to control rabies exist – a post-exposure vaccine for bite victims is effective if delivered expediently, and mass dog vaccination can eliminate rabies entirely. However, prevention and control depends on effective surveillance that enables timely detection of, and coordinated responses to, outbreaks. In Tanzania, such infrastructure is generally poor and unreliable.

Frustrated by the unreliable surveillance and tragic experiences of families striving but failing to obtain life-saving vaccines, a University of Glasgow team developed a user-friendly mobile phone-based system. The system uses the widespread reach of mobile technologies, offering a user-friendly platform that enables frontline health and veterinary workers to upload reports from basic mobile phones via a mobile network. Key stakeholders are able to access and share the data, which is stored in a central server. This has empowered health workers and communities across southern Tanzania to gain better access to vaccines, demonstrate the rabies burden and need for action, and support the government to prevent the disease through mass dog vaccinations.

The system is an example of affordable mobile technologies adopted at scale in low-income settings. It has gained considerable momentum: accelerating the collaboration between health and veterinary sectors required to manage zoonotic diseases such as rabies, and supporting the targets set by major international organisations to eliminate rabies globally.

Runner up: Bath Spa Univerity

Bath Spa University's Global Academy of Liberal Arts (GALA) is an international network of liberal arts institutions, faculties and programmes that facilitates collaboration in teaching and research. It also supports student and staff mobility, and promotes liberal arts education in a transnational context. BSU's 17 partners come from across Europe, Asia, North and Central America, with African and South American partners pending.

GALA focuses on the liberal arts in the broadest sense – including the humanities, social sciences, sciences, and arts – with a particular interest in creative entrepreneurship and civic

the disease through mass dog vaccinations.

The system is an example of affordable mobile technologies adopted at scale in low-income settings. It has gained considerable momentum: accelerating the collaboration between health and veterinary sectors required to manage zoonotic diseases such as rabies, and supporting the targets set by major international organisations to eliminate rabies globally.

Runner up: Bath Spa Univerity

Bath Spa University's Global Academy of Liberal Arts (GALA) is an international network of liberal arts institutions, faculties and programmes that facilitates collaboration in teaching and research. It also supports student and staff mobility, and promotes liberal arts education in a transnational context. BSU's 17 partners come from across Europe, Asia, North and Central America, with African and South American partners pending.

GALA focuses on the liberal arts in the broadest sense – including the humanities, social sciences, sciences, and arts – with a particular interest in creative entrepreneurship and civic engagement. Its activities include shared teaching, transnational research, artworks, conferences both in the UK and abroad, and public events. It has also been involved with the university's own recent restructuring, contributing directly to the establishment of a College of Liberal Arts, one of the first in the UK.

Runner up: University of the West of Scotland

UWS' dementia palliare project creates interprofessional learning resources to equip qualified health and social care workers to deliver the best possible care to people with advanced dementia and to support family caring.

The project is creating the impetus for new positive approaches to this often neglected area of practice. Initially dementia palliare has focused on creating a shared understanding about best palliare practice, by encouraging an understanding of the experience of this stage of dementia from an individual and family perspective.

A European best practice statement (BPS), informed by in-depth evidence, policy review and original international case studies, has since been endorsed by Alzheimer Europe. This is inspiring new positive ways to develop advanced dementia practice, helping practitioners (doctors, nurses, allied health and social care) to imagine new and better approaches to practice and lead reforms.

UWS is the lead partner in the project, working with six universities from Sweden, Finland, Czech Republic, Slovenia, Portugal and Spain.

Edit -Delete -Read

Thu, 17 Mar 2016 19:11:30 GMT

Islamic State committed genocide, says US

Edit -Delete -Read

Thu, 17 Mar 2016 19:08:58 GMT

This project was supported by a range of funding bodies and collaborative partners:

"Naming these crimes is important," Mr Kerry said, "but what is essential is to stop them."

Mr Kerry also said IS was responsible for crimes against humanity and ethnic cleansing in areas it controls in Syria and Iraq.

He said his conclusions had been based on a wealth of evidence provided by the US state department, intelligence teams and other sources.

They include well-documented accounts of IS attacks on the Yazidi community in Iraq, which led to the deaths of hundreds of men and boys and the abduction of thousands of women.

Tens of thousands of Yazidis became stranded on an exposed mountain, and Mr Kerry said "without our intervention, it is clear those people would have been slaughtered".

He also highlighted the killings of Christians in northern Iraq and Libya, and of Shia Turkmen in Iraq.

"The fact is that Daesh [IS] kills Christians because they are Christians, Yazidis because they are Yazidis, Shia because they are Shia," he said, using an Arabic acronym for the group.

"This is the message it conveys to children under its control. Its entire world view is based on eliminating those who do not subscribe to its perverse ideology."

Mr Kerry admitted that a lack of access to IS areas meant the US did not have a "complete picture" of the atrocities that had been carried out, and said he was "neither judge, nor prosecutor, nor jury".

But he said he hoped its victims would take comfort in the fact that "the United States recognises and confirms the despicable nature of the crimes committed against them".

How the UN defines genocide

Article II of the 1948 UN Genocide Convention says genocide means any of the following acts committed "with intent to destroy, in whole or in part, a national, ethnical, racial or religious group, as such". They are:

- Killing members of the group.
- Causing serious bodily or mental harm to members of the group.
- Deliberately inflicting on the group conditions of life calculated to bring about its physical destruction in whole or in part.
- Imposing measures intended to prevent births within the group.
- Forcibly transferring children of the group to another group.

What is genocide?

It is only the second time the US administration has declared a genocide during a conflict. The previous time was in 2004 when then-Secretary of State Colin Powell used it to describe the killings in Darfur.

Such a declaration is a powerful signal, says the BBC's diplomatic correspondent Jonathan Marcus, and Mr Kerry may hope that it bolsters the fight against the IS and possibly opens the way to action at the UN Security Council.

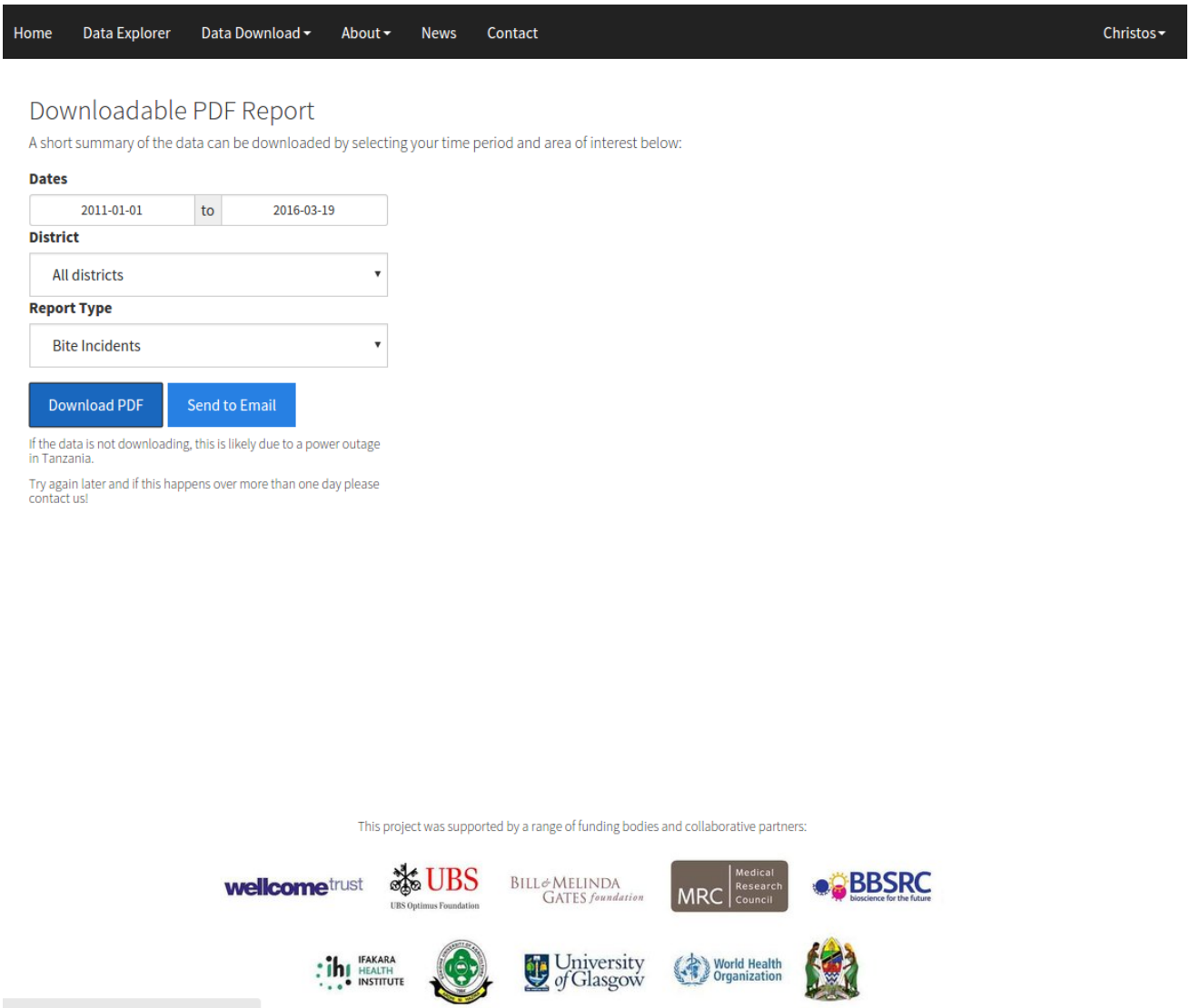
But in every other sense its practical impact will be limited, our correspondent adds, as the US and its allies are already engaged in a war against IS and the struggle is likely to continue for months and probably years yet.

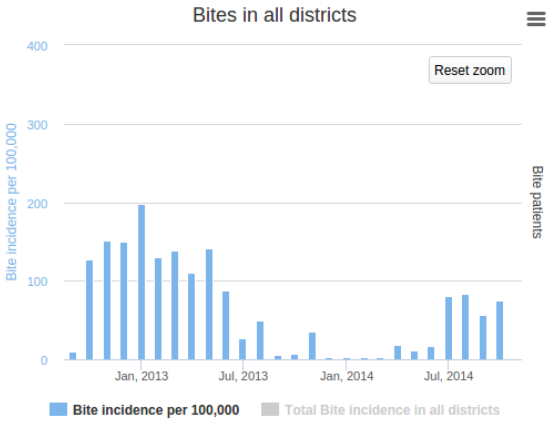
Mr Kerry had been given a congressional deadline of 17 March to announce whether IS's actions in areas under its control constituted genocide.

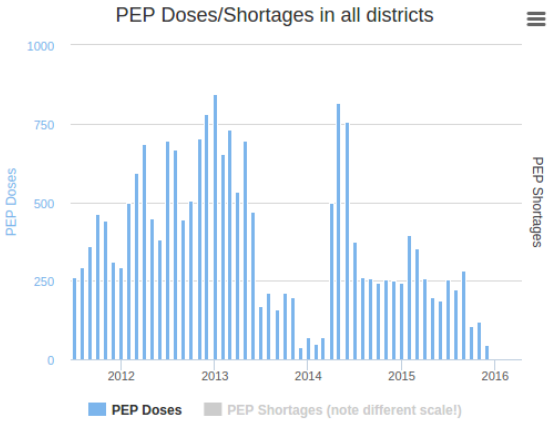
Earlier this week, the US House of Representatives voted 393-0 to designate the crimes as genocide.

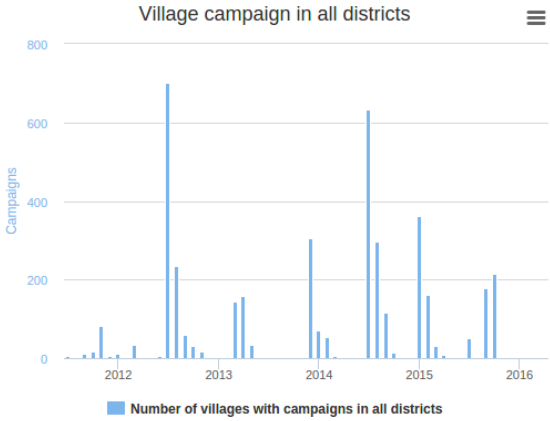
Some critics have accused the Obama administration of not speaking out forcefully enough about the treatment of minority groups by IS.

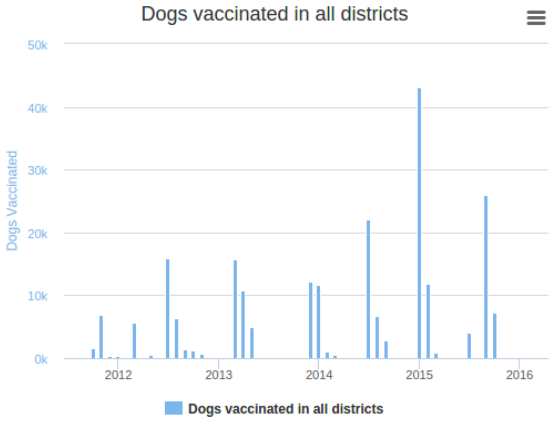
What do we know about the groups targeted by IS?

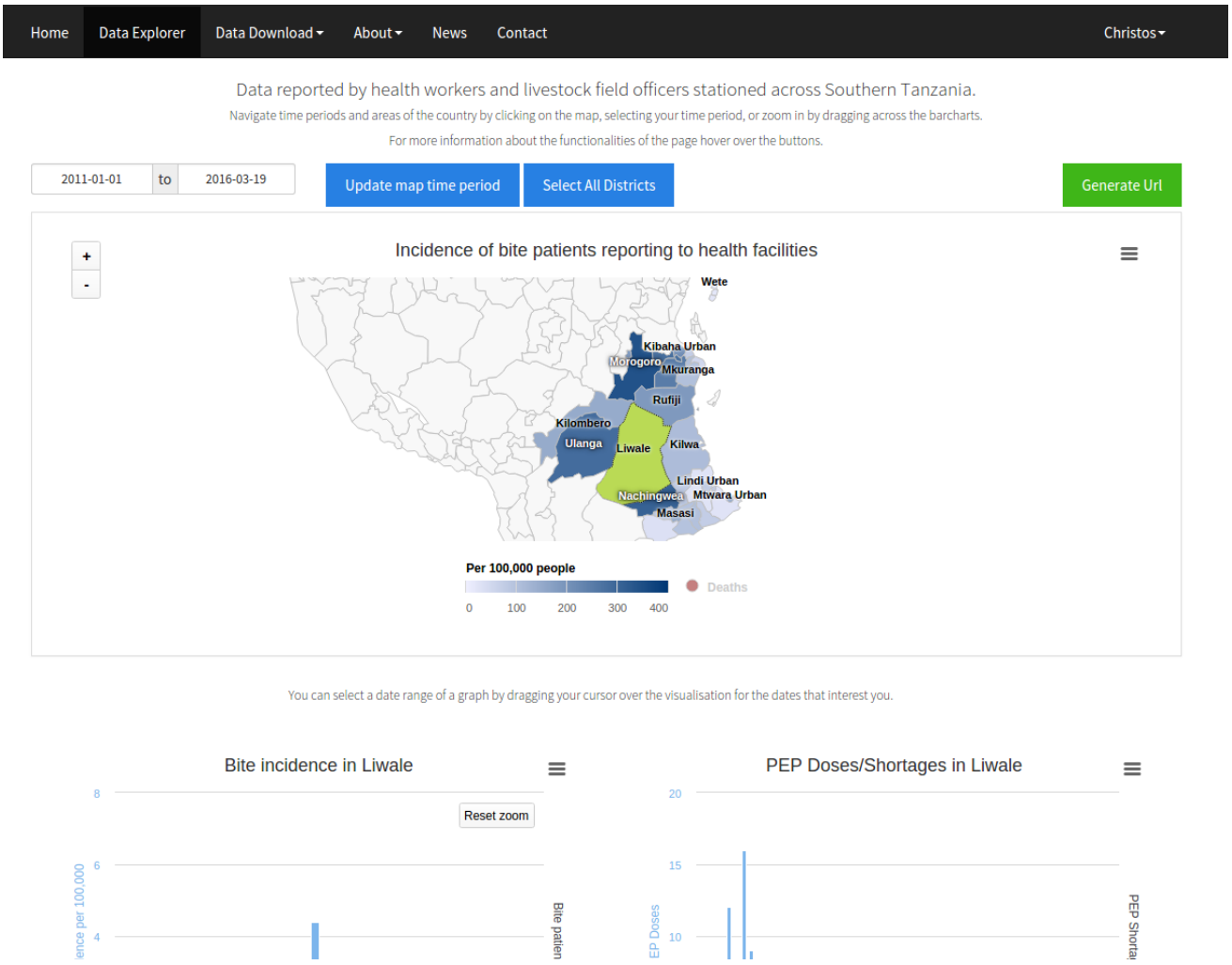

Home
Data Explorer
Data Download
About
News
Contact
Christos
Downloadable PDF Report
A short summary of the data can be downloaded by selecting your time period and area of interest below:
Dates
2011-01-01
to
2016-03-19
District
All districts
Report Type
Bite Incidents
Download PDF
Send to Email
If the data is not downloading, this is likely due to a power outage in Tanzania.
Try again later and if this happens over more than one day please contact us!
This project was supported by a range of funding bodies and collaborative partners:
wellcome trust
UBS
UBS Optimus Foundation
BILL & MELINDA GATES foundation
MRC
Medical Research Council
BBSRC
bioscience for the future
IFAKARA HEALTH INSTITUTE
University of Glasgow
World Health Organization

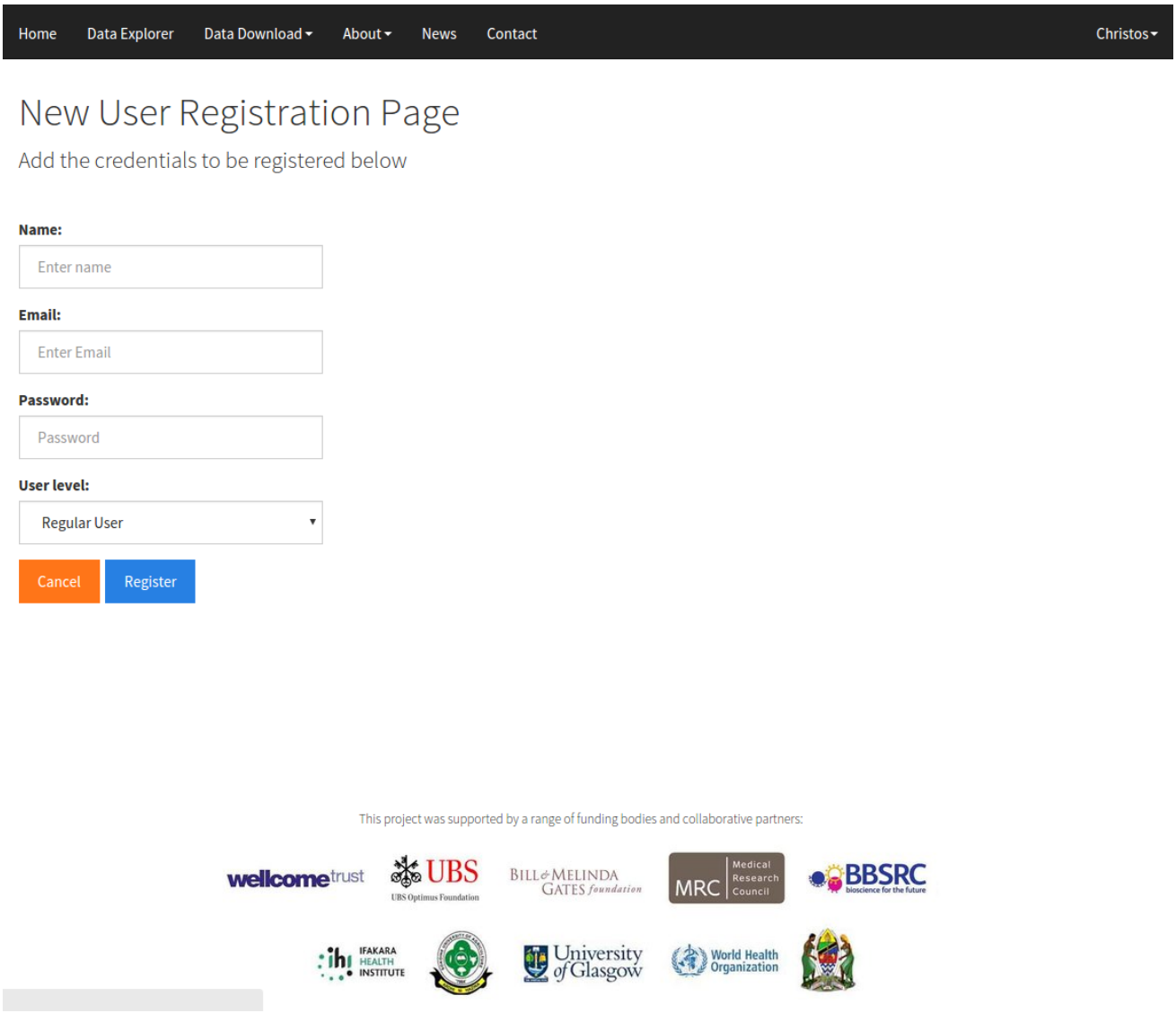

Bites in all districts
Reset zoom
Bite incidence per 100,000
Bite patients
400
300
200
100
0
Jan, 2013
Jul, 2013
Jan, 2014
Jul, 2014
Bite incidence per 100,000
Total Bite incidence in all districts

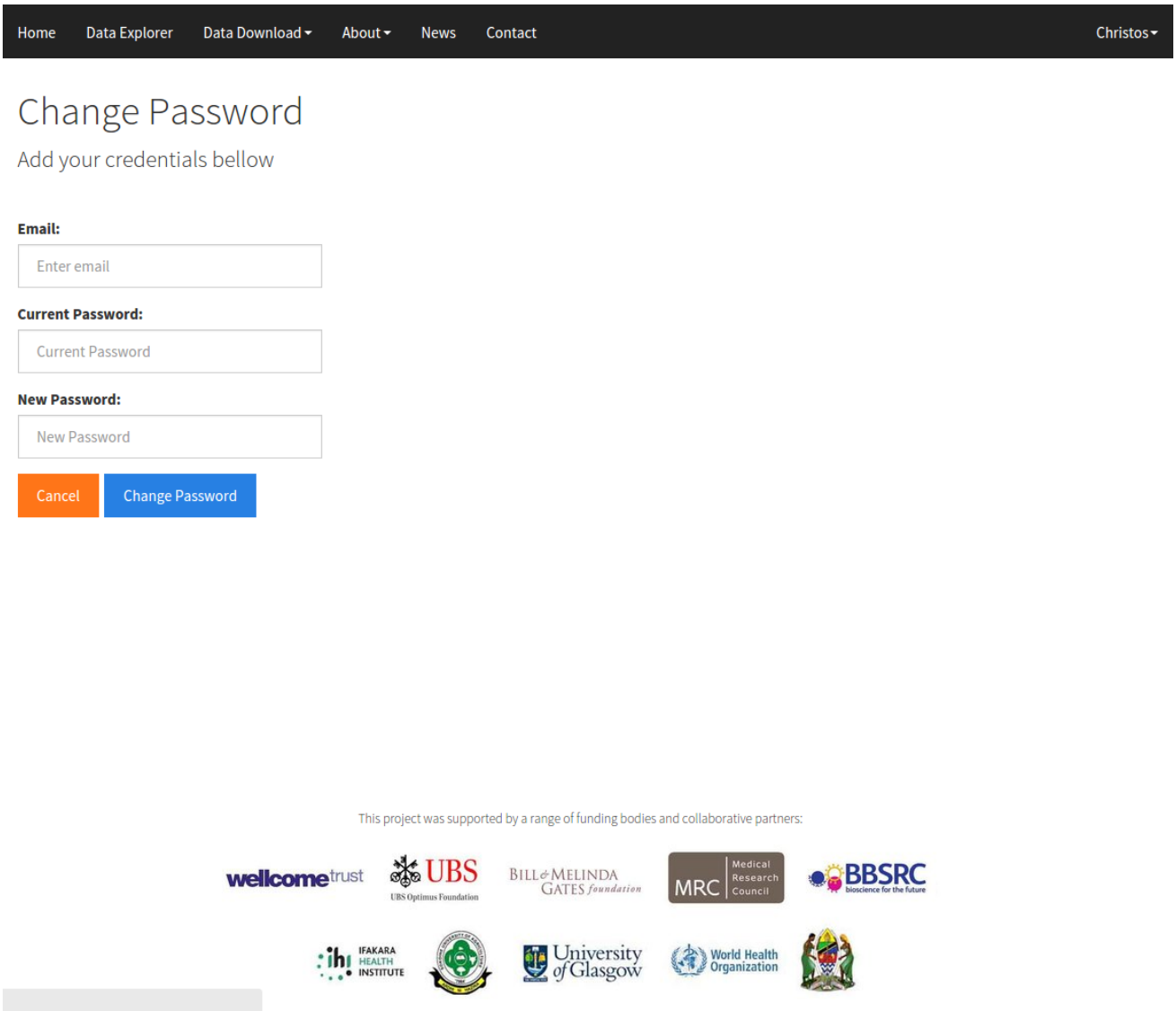

PEP Doses/Shortages in all districts
PEP Doses
PEP Shortages
1000
750
500
250
0
2012
2013
2014
2015
2016
PEP Doses
PEP Shortages (note different scale!)


Village campaign in all districts
Campaigns
800
600
400
200
0
2012
2013
2014
2015
2016
Number of villages with campaigns in all districts


Dogs vaccinated in all districts
Dogs Vaccinated
50k
40k
30k
20k
10k
0k
2012
2013
2014
2015
2016
Dogs vaccinated in all districts


Home
Data Explorer
Data Download
About
News
Contact
Christos
Data reported by health workers and livestock field officers stationed across Southern Tanzania.
Navigate time periods and areas of the country by clicking on the map, selecting your time period, or zoom in by dragging across the barcharts.
For more information about the functionalities of the page hover over the buttons.
2011-01-01
to
2016-03-19
Update map time period
Select All Districts
Generate Url
Incidence of bite patients reporting to health facilities
Wete
Kibaha Urban
Morogoro
Mkuranga
Rufiji
Kilombero
Ulanga
Liwale
Kilwa
Lindi Urban
Nachingwea
Mtwara Urban
Masasi
Per 100,000 people
Deaths
0
100
200
300
400
You can select a date range of a graph by dragging your cursor over the visualisation for the dates that interest you.
Bite incidence in Liwale
Reset zoom
8
6
4
Bite patien
PEP Doses/Shortages in Liwale
20
15
10
EP Doses
PEP Shortag


Home
Data Explorer
Data Download
About
News
Contact
Christos
New User Registration Page
Add the credentials to be registered below
Name:
Enter name
Email:
Enter Email
Password:
Password
User level:
Regular User
Cancel
Register
This project was supported by a range of funding bodies and collaborative partners:
wellcome trust
UBS
UBS Optimus Foundation
BILL & MELINDA GATES foundation
MRC
Medical Research Council
BBSRC
bioscience for the future
IFAKARA HEALTH INSTITUTE
University of Glasgow
World Health Organization


Home
Data Explorer
Data Download
About
News
Contact
Christos
Change Password
Add your credentials bellow
Email:
Enter email
Current Password:
Current Password
New Password:
New Password
Cancel
Change Password
This project was supported by a range of funding bodies and collaborative partners:
wellcome trust
UBS
UBS Optimus Foundation
BILL & MELINDA GATES foundation
MRC
Medical Research Council
BBSRC
bioscience for the future
IFAKARA HEALTH INSTITUTE
University of Glasgow
World Health Organization

Rabies

Rabies is a fatal disease that can infect all mammals, but is primarily spread by domestic dogs. Following a bite, rabies can be prevented through prompt administration of post-exposure prophylaxis. This involves a course of vaccinations administered over several weeks, together with immunoglobulins for high-risk exposures. A major challenge in low-income countries is ensuring these vaccines are available and affordable to bite victims. The risk of exposure can be reduced and rabies can be controlled at source through mass dog vaccination. Rabies has been eliminated in industrialized countries through mass dog vaccination, however in most low-income countries there has been little investment in dog vaccination and rabies continues to kill thousands of people every year.

The momentum for the global elimination of rabies spread by domestic dogs is growing and targets have been set for 2030. For more information about rabies generally visit the Global Alliance for Rabies Control.

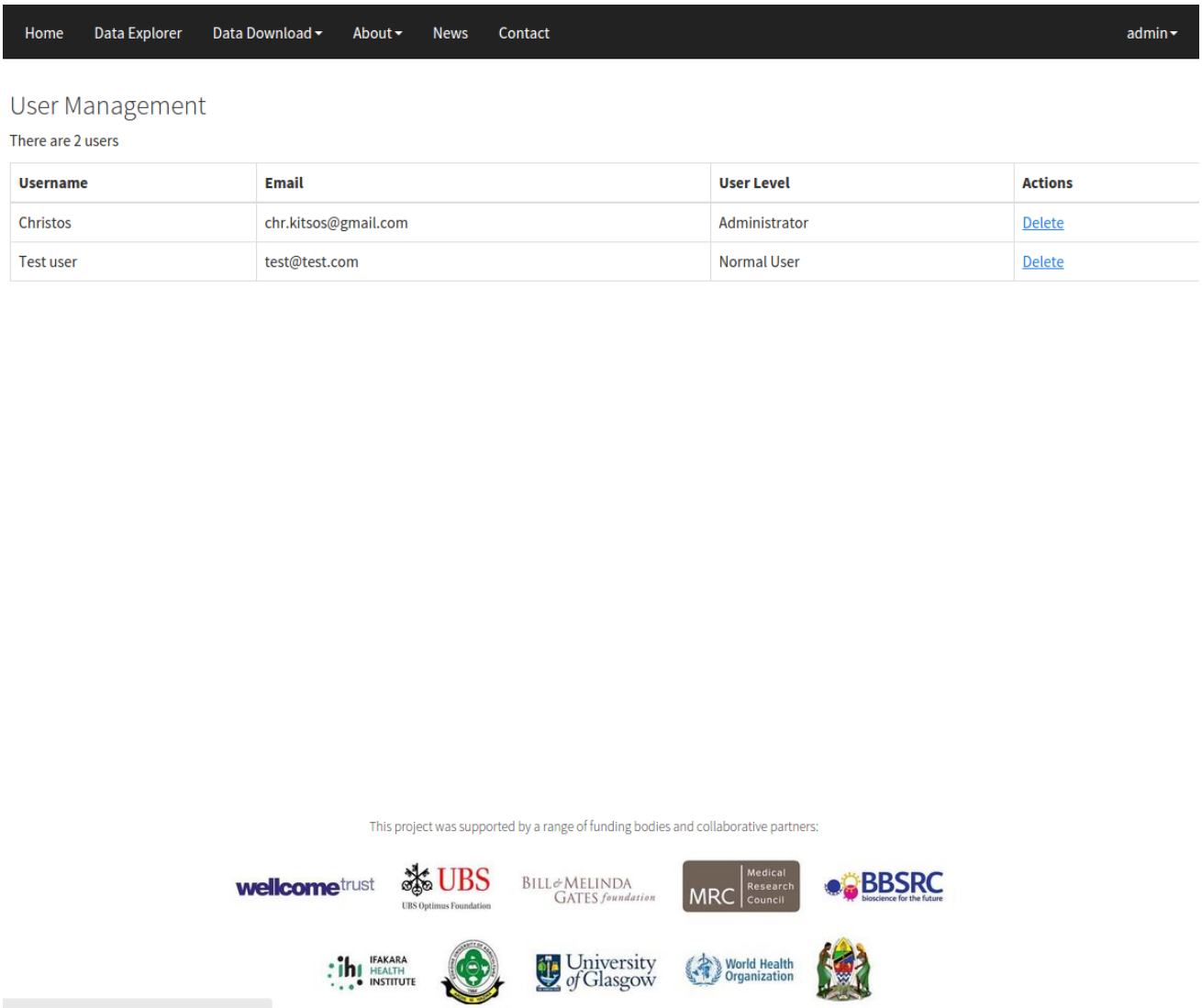
Rabies control and prevention project in southern Tanzania

In 2010, the Tanzanian government and the World Health Organization secured funding from the Bill and Melinda Gates Foundation for a large-scale rabies control programme across southern Tanzania, as part of a multi-country initiative. The overarching aim was to eliminate human rabies deaths through establishing annual mass dog vaccination campaigns across the region and improving the provision of post-exposure prophylaxis to bite victims. In Tanzania this project operates across 28 districts, including Pemba Island and covers a 150,000km2 catchment area serving around 10 million inhabitants.

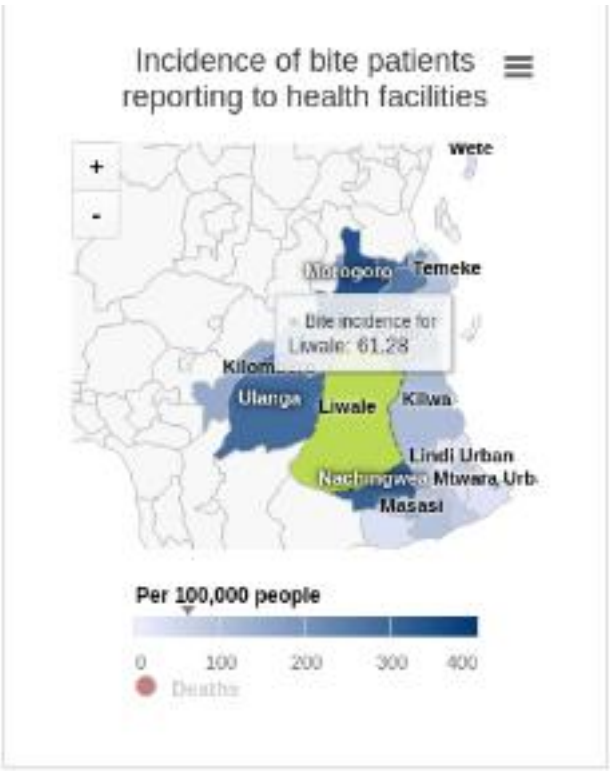
Mobile phone-based surveillance system

Surveillance is critical for managing preventative health services and controlling infectious diseases. Surveillance involves the routine collection, analysis and dissemination of data to guide health policy and practice. But paper-based surveillance is slow and often incomplete, therefore does not allow effective monitoring or timely responses. Surveillance for zoonotic diseases (spread from animals to humans) requires intersectoral collaboration between the health and veterinary sectors. For rabies, health workers need to report animal bites to veterinary officers to trigger outbreak investigations, and vets need to alert medical authorities to exposure risks from animal cases. Our mobile phone-based surveillance system for rabies supports around 300 healthworkers and veterinary officers to record information needed to monitor the progress of rabies control and prevention efforts in Southern Tanzania.

This project was supported by a range of funding bodies and collaborative partners:


Home
Data Explorer
Data Download
About
News
Contact
admin
User Management
There are 2 users
Username
Email
User Level
Actions
Christos
chr.kitsos@gmail.com
Administrator
Delete
Test user
test@test.com
Normal User
Delete
This project was supported by a range of funding bodies and collaborative partners:
wellcome trust
UBS
UBS Optimus Foundation
BILL & MELINDA GATES foundation
MRC
Medical Research Council
BBSRC
bioscience for the future
IFAKARA HEALTH INSTITUTE
University of Glasgow
World Health Organization


Home
Data Explorer
Data Download
About
News
Contact
Christos
Rabies Surveillance in Southern Tanzania
Welcome to our platform for rabies surveillance in Tanzania. This is a collaboration between researchers at the Ifakara Health Institute, Sokoine University of Agriculture, the University of Glasgow and the Tanzanian government. Since 2010, the government has implemented a large-scale rabies prevention and control programme across 28 districts in Southern Tanzania, coordinated by WHO. This programme involves provision of lifesaving post-exposure prophylaxes to bite victims to prevent the onset of rabies and mass vaccination of domestic dogs to interrupt transmission and ultimately eliminate disease.



You can select a date range of a graph by dragging your cursor over the visualisation for the dates that interest you.

Rabies Surveillance in southern Tanzania website evaluation

Hello and thank you for participating in the test user evaluation. This website has been designed as a platform to share and visualise data and progress from the rabies surveillance project in Southern Tanzania. It has been designed to work effectively in areas with limited internet connectivity.

The evaluation will consist of you (the test user) doing the following:

- Complete a structured set of tasks in order to test some of the core functionalities of the website
- Answer a short questionnaire about the tasks you tried to achieve in order to provide feedback about the functionalities the content and the quality of the website

Tasks to achieve

1. Go to the website http://rabieswebsite-rabiesresearch1.rhcloud.com/
2. Log into the website with the following credentials
    1. Username: *Available in the pdf sent to you*
    2. Password: *Available in the pdf sent to*

Appendix G

Documentation

System Documentation

General Information

The system consists of one package of source code files. The Server and Client are connected so there are no 2 seperate folders. The web server uses NodeJS + ExpressJS

There are 3 “programming languages” used in the package:

- Javascript - Responsible for the Client and Web Server Logic
- JADE(HTML) - Is an alternative way of writing HTML. It has the same functionalities
- CSS - Responsible for all the objects rendered in the front end

System Structure

The plain source code package without any modules installed consists of the following files/folders

- public - All the client static content is located here (Javascript, CSS, Images, Maps)
- routes - Contains the API and the routing logic of the web server
- views - All the HTML/JADE that gets rendered is contained here
- deplist.txt - This document is deprecated and is only there for backwards compatibility
- package.json - All the web server installed modules are listed there, this allows the installation and updating of all the modules automatically
- README.md - Small documentation for the Git repository
- server.js - The configuration of the expressJS server, This is the main server file
- website.db - This is a SQLite3 embedded database, preferably it should be moved elsewhere

How to Run

The package can be run locally and in a server environment. For both actions “node” must be installed to that specific device/server. In order to run the package you need to go to the root folder and do the following :

- ~$ npm install (Installs all the modules needed to run the server)
- ~$ node server.js (Starts the server)

The server should start running at the specified IP address and PORT. These values can be modified inside the server.js file close to the end of the file.

Server

The server is written in NodeJS + ExpressJS, that means that all the back end code is in Javascript. The server logic consists of 3 main files : server.js, api.js and index.js. The server uses the JADE template(which is ExpressJS default) to parse HTML files and it has an embedded SQLite database for user authentication, logging and storing news posts. The authentication is done through the Passport package.

How to install/update packages

In order to install new packages into the web server, NPM will have to be used, and by going to the folder where server.js and package.json exists. When packages are installed a node_modules folder gets created with the installed packages inside, this folder should be ignored by the repository as it gets automatically created when “~$ npm instal” is run. Once there the following need to be typed through the terminal for the following actions.

- ~$ npm install -S “package name” - This installs the specified package, the -S flag makes sure that the package gets listed into the package.json
- ~$ npm install - This installs all the packages listed into package.json automatically, it generates a node_modules folder which contains all the packages.
- ~$ npm update - This updates all the specified packages to their latest version

How to run the server

The server can be run simply by going to the folder where server.js is located and typing :
“~$ node server.js”
An automated script can be written so this is done automatically every time the github repository is updated.

How to modify/expand the server logic

API

The majority of the API endpoints are located in the /routes/api.js file. There mostl the implemented API functionalities can be found. If it is wished to add an additional endpoint or modify an existing one, it should be done in there.

Keep in mind that if you develop a new API endpoint you will have to declare it inside the server.js file in the root folder.

Server configuration

All the configuration of the server is done in server.js located in the root folder of the package. There you can modify

- PORT/IP addresses
- General expressjs configuration
- Include additional packages installed
- Passport Authentication strategies

Adding extra JADE/HTML pages

If it is required to add an additional page into the web site a JADE file will have to be created first inside /views/partials. When a URL requests a specific HTML template the server renders one with the same name of the requested from that folder (You will have to add the new JADE page routing logic in the /public/app/app.js too so angular can render it correctly).

Changing/Modifying the Passport Authentication Strategy

For modifying or changing the authentication strategy you will have to go to /server.js as all of it is located there. Comments clearly identify which part of the code is responsible for this strategy. For any modifications please consult *http://passportjs.org/* first as a lot of information can be found there

Client

The client can be mainly found inside the /public/ folder, there all the Javascript, CSS and miscellaneous files can be found. (Keep in mind the HTML/JADE is located in /views/ and not in the /public/ folder). The client consists of 2 sub folders /public/app which has all the application logic and implementation and /public/assets which contains all the libraries and technologies used by the web application.

The client users AngularJS for the implementation of the project and thus it is structured in a specific way :

- App.js - Here the configuration of the web application can be found. This includes declaring technologies and setting up the routing logic. (If it is decided to add a new page in the application you need to configure it in here as well as add the HTML/JADE in the /views/partials/ folder
- Controller.js - Here you will find all the implementation of the web application. This includes generation of maps, API calls etc

How to modify/expand the client

Installing Updating Technologies

All the technologies and libraries are located inside the /public/assets/ folder. You can install a technology by placing the required folders in there and then going to /views/index.jade and pointing to their location. Sometimes you might have to include the name of the library inside the app variable declaration which can be found in the /public/app/app.js file.

Adding new web page

In order to add a new web page to the web application it is needed to first add the JADE/HTML file inside the /public/partials folder. Then the routing logic needs to be inserted inside the app.js. Keep in mind that if the web page to be added is nonstatic you will have to create a new controller for it to serve its implementation, the controller will have to be added in the /public/app/controllers.js file

Modifying an existing web page

Each individual web page consists of two things: The JADE file and the implementation. In order to modify the JADE/HTML file you will have to go into the /views/partials/ folder and find the file required. Then the file can be simply changed. If it is the implementation that needs to be modified you will have to find the controller that is linked to that page in the /public/app/controllers.js and modify the code or add new one. The controller set for every page can be found inside the /public/app/app.js file

Adding/Modifying Visualisations

All the visualisations can be found in the dataExplorer page. Thus all the implementation is located inside the *dynamicDataExplorerCtrl* controller. There all the configurations for the already developed visualisations can be found. In order to add new ones or modify the existing ones please consult http://www.highcharts.com/. The website contains a list of all the configuration options. In addition it contains a large variety of already made examples.

Adding/Modifying the CSS

The web application uses Bootstrap as the default library for CSS. That means that you should not have to create your own CSS as Bootstrap provides a huge amount of functionalities. If you wish to modify already existing elements of Bootstrap or create your own you will have to do it in the following file : /public/assets/css/custom.css Everything that is written there overwrites

Bootstrap so it can be used for extending Bootstrap elements. In addition any new CSS files that get added to the web application will have to be imported in the /views/layout.jade file.

Migration to AWS

KEEP IN MIND THE FOLLOWING ARE JUST GUIDELINES AND NOT SPECIFIC INSTRUCTIONS! FOR PROPER INSTRUCTIONS CONSULT https://aws.amazon.com/ec2/

In order to migrate the system to AWS you will need to create an EC2 Ubuntu instance in AWS. You will need to install "NODE" on that instance by typing the following terminal command

- `curl -sL https://deb.nodesource.com/setup_4.x | sudo -E bash - sudo apt-get install -y nodejs`

Then you will have to move the source code to that EC2 instance and place in a desired folder. You will have to modify the IP ADDRESS and PORT in the server.js file to match with the ones provided by the instance.
The server can be started by typing simply "~$ node server.js" inside the root folder of the source code. It is reccomended to deploy the project through github to EC2 please consult http://docs.aws.amazon.com/codedeploy/latest/userguide/github-integ-tutorial.html

Support Material

It is recommended to get familiar with the main technologies used in this system before modifying or adding anything. The following websites provide rich information about all the complex aspects that you might encounter.

- NodeJS - https://nodejs.org/en/
- ExpressJS - http://expressjs.com/
- AngularJS - https://angularjs.org/
- NPM - https://www.npmjs.com/
- Highcharts - http://www.highcharts.com/
- Bootstrap - http://getbootstrap.com/
- Passport - http://passportjs.org/
- JADE - http://jade-lang.com/
- HTML to JADE converter : http://html2jade.org/
- PDFMake - http://pdfmake.org
- JsonToCsv - https://github.com/mrodrig/json-2-csv
- SQLite3 - https://github.com/mapbox/node-sqlite3
- MySQL - https://github.com/felixge/node-mysql
- NodeMailer - https://github.com/nodemailer/nodemailer
- TextAngular - https://github.com/fraywing/textAngular
- Bootstrap Datepicker - https://bootstrap-datepicker.readthedocs.org/en/latest/