ASHESI UNIVERSITY

REAL-TIME DISEASE SURVEILLANCE AND ANALYTICS TO IMPROVE DECISION MAKING IN GHANA’S HEALTH SECTOR

APPLIED PROJECT

B.Sc. Computer Science

Ekiyor Toluwani Shammah Odoko

2019
ASHESI UNIVERSITY

REAL-TIME DISEASE SURVEILLANCE AND ANALYTICS TO IMPROVE
DECISION MAKING IN GHANA’S HEALTH SECTOR

APPLIED PROJECT

Applied Project submitted to the Department of Computer Science, Ashesi University in
partial fulfillment of the requirements for the award of Bachelor of Science degree in
Computer Science.

Ekiyor Toluwani Shammah Odoko

2019
DECLARATION

I hereby declare that this applied project is the result of my own original work and that no part of it has been presented for another degree in this university or elsewhere.

Candidate’s Signature:

..................................................................................................................................................

Candidate’s Name:

..................................................................................................................................................

Date:

..................................................................................................................................................

I hereby declare that preparation and presentation of this applied project were supervised in accordance with the guidelines on supervision of applied project laid down by Ashesi University.

Supervisor’s Signature:

..................................................................................................................................................

Supervisor’s Name:

..................................................................................................................................................

Date:

..................................................................................................................................................
Acknowledgements

First and foremost, I would like to acknowledge the almighty of God whose mercies and favour has kept me throughout this journey.

I am sincerely grateful to my parents, Rev. Temple. E. Odoko & Pst. Mrs. Zannie Odoko whose love, support and guidance has molded me to the man I am today. They made sure I get the best of everything when it matters. I will always remain thankful.

Finally, I would like to show appreciation to my supervisor, Mr. David Sampah who has been patient and worked closely with me to see this project to completion.
Abstract

People assume responsibility for their personal health, but on a nation-wide scale, governmental bodies exist to monitor the health status of citizens in the country. This process of monitoring is known as Public health surveillance. Public health surveillance consists of very important sub-categories and one of these categories is disease surveillance. Disease surveillance is concerned with monitoring the spread of diseases to establish trends or patterns.

The common method of disease surveillance in West Africa & Ghana relies on the pen-and-paper approach which is costly and untimely. The ineffectiveness of this approach was evident during the Ebola crisis in 2014. Since then Ghana has acquired a new system, the District Health Information Management System (DHIMS2) to conduct disease surveillance. The DHIMS 2 does a god job in eliminating most of the problems associated with previous implementations for public health surveillance, but it errs on the side of speed and reliability as data collation is not conducted in real-time and hence presents a lapse to the Ghana Health Service.

Therefore, the aim of this project is to design and implement a scalable web-based application which health facilities will use to submit disease reports to health administrators who can in turn monitor this disease cases and perform analysis on them.
Table of Content

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>DECLARATION</td>
<td>i</td>
</tr>
<tr>
<td>Acknowledgement</td>
<td>ii</td>
</tr>
<tr>
<td>Abstract</td>
<td>iii</td>
</tr>
<tr>
<td>Chapter 1: Introduction</td>
<td>1</td>
</tr>
<tr>
<td>1.1 Background</td>
<td>1</td>
</tr>
<tr>
<td>1.1.1 Public health surveillance</td>
<td>1</td>
</tr>
<tr>
<td>1.1.2 EHR</td>
<td>1</td>
</tr>
<tr>
<td>1.1.3 Artificial Intelligence</td>
<td>2</td>
</tr>
<tr>
<td>1.1.4 Disease surveillance</td>
<td>2</td>
</tr>
<tr>
<td>1.1.5 Public health &amp; disease surveillance in Ghana</td>
<td>2</td>
</tr>
<tr>
<td>1.2 Significance of Problem</td>
<td>4</td>
</tr>
<tr>
<td>1.3 Existing Solution: DHMS 2</td>
<td>5</td>
</tr>
<tr>
<td>1.3.1 System overview</td>
<td>5</td>
</tr>
<tr>
<td>1.3.2 Problems DHIMS 2 solves</td>
<td>6</td>
</tr>
<tr>
<td>1.3.3 Challenges with the system</td>
<td>7</td>
</tr>
<tr>
<td>1.4 Objectives</td>
<td>7</td>
</tr>
<tr>
<td>1.5 Scope of project</td>
<td>8</td>
</tr>
<tr>
<td>Chapter 2: Requirements Engineering</td>
<td>10</td>
</tr>
</tbody>
</table>

iv
2.1 Requirement elicitation ........................................................................................................... 10

2.2 Requirement analysis ............................................................................................................. 10

2.3 Users ....................................................................................................................................... 11
   2.3.1 Hospital facility user ......................................................................................................... 11
   2.3.2 Health administrators ..................................................................................................... 12

2.4 Functional requirements ........................................................................................................ 12
   2.4.1 Use Case Diagram .......................................................................................................... 14

2.5 Non-functional requirements ................................................................................................. 15

2.6 Organizational requirements ................................................................................................. 15

Chapter 3: Architecture & Design ............................................................................................... 16

3.1 Architecture overview ........................................................................................................... 16
   3.1.1 Database brief ................................................................................................................ 16
   3.1.2 Activity diagram ............................................................................................................ 16

3.2 System architecture ............................................................................................................... 17
   3.2.1 MVC Architecture ........................................................................................................ 18
   3.2.2 Client-server architecture ............................................................................................ 20

3.3 Design Specification ............................................................................................................. 20
   3.3.1 Logical view .................................................................................................................. 20
   3.3.2 Database Architecture .................................................................................................. 21

Chapter 4: Implementation .......................................................................................................... 23
Chapter 1: Introduction

The third goal of the Sustainable Development Goals outlined by the United Nations aim to promote healthy lives and well-being amongst all ages in the global context [1]. This verifies that health is a very important issue worldwide. Not only does improved health increase individual life expectancy, it is also reported to boost rates of economic growth [2].

1.1 Background

1.1.1 Public Health Surveillance

In attempting to promote healthy wellbeing for all, federal and state health agencies ought to monitor the state of health of the public. This monitoring process is referred to as Public Health Surveillance [3]. Public health surveillance is an ongoing process which includes acquisition, analysis, interpretation and dissemination of health data and therefore agencies conducting surveillance must always be informed [3]. This process can directly measure the happenings in the environment and is effective in achieving two purposes; Measuring the need to intervene in a health situation and measuring the effects of such intervention [4].

1.1.2 EHR

Electronic Health Record (EHR) is an example of an innovative approach by the ministry of health in various nations, in which they rely on technology for proper management of health data. Electronic Health Record integrates the use of databases to store data acquired through hospitals, which is an important practice in public health surveillance and makes the
process of data collation easy. This deliberate use of technology has the potential to improve “health system efficiencies and prevent medical errors” [5].

1.1.3 Artificial Intelligence

Also, with the widespread sensitization of the benefits of Artificial Intelligence (AI) during the last decade, private and public research sectors in fast growing economies like China’s do not only acquire and archive data for previewing and future reference. These nations utilize the possibilities of AI to increase their probability of promoting healthy living amongst their populations [6]. One of many examples is probabilistic machine learning models. This is a detection algorithm that can be used to follow long term trends and patterns that exist in the occurrence of diseases amongst the public for early detection [7]. This is a major reason why public health surveillance is conducted, to monitor health cases [3]. Evidently, AI is playing a huge role in public health surveillance.

1.1.4 Disease Surveillance

Under the umbrella of public health surveillance, health sectors are also interested in tracking and monitoring all or specific diseases that are recorded in the country. This process is called disease surveillance and the purpose is to identify existing trends in the occurrence of diseases and provide a medium to know what actions to take in the events of an epidemic.

1.1.5 Public Health & Disease Surveillance in Ghana

In both 2017 and 2018 meetings of the Huawei Ghana Seeds for the Future Program, the Vice President of Ghana, Dr. Mahamudu Bawumia declared that the government would enforce the use of ICT as it has the power to transform Ghana’s economy [8]. This might seem as progress for Ghana in this context, but not quite. Prior to this announcement, Ghana has
already used technology in the health sector. An example is the biometric membership cards issued by the National Health Insurance Authority (NHIA). At the health conference organized by the Association of Chartered Certified Accountants in 2013, delegates in attendance identified that technology could also be used for “monitoring outbreaks of disease; recording and transmitting a patient’s vital signs; appointment booking” [9]. Nonetheless, the current state of technology in Ghana’s health sector does not reflect the projections made in the past. The question is “why?”.

Studies have shown that policies emphasizing on technology-led growth are in futility if policy processes do not facilitate collaboration and cooperation [10]. Now, there are issues in sub-Saharan African countries such as Ghana that make it difficult for technology-oriented solutions to be effective after implementation. These are the issues hindering the progress of policies that has been setup in this regard. In most areas, power infrastructure (electricity supply) is inadequate and inconsistent. This is a major issue in the use of technology to aid public health surveillance as technology is most efficient with uninterrupted power supply. The system being used will always have to be on constant power to monitor public health at every point in time.

Poor ICT background makes it difficult for assigned personnel to use technological systems. Most times the institution using the system would have to organize workshops for their workers before they begin to use it. This can be costly. Therefore, inadequate technical skills to utilize such technological solutions in Ghana is another lingering problem [11].
1.2 Significance of Problem

Africa is behind when it comes to the infusion of technology in the health industry and this was globally exposed by the recent epidemic that hit west Africa, Ebola. Most media bodies and federal officials believe that Ebola was a disaster and could have been handled effectively if the health systems were properly funded and facilitated. In an online publication, the media body Africa Renewal stated that, analysts believed the response to the outbreak was weak and that the health systems in the countries hit by Ebola were “on their knees”, making it difficult to manage the situation [11]. Other factors like equipment, medical staff & infrastructure may also have reduced the casualties of the disease, but much could have been avoided with the aid of technology in the health systems [11].

Most health facilities; Hospitals, Health-centers, Labs etc. that are not properly facilitated still record valuable data using the traditional paper-based approach, which makes analysis and further usage of such data close to impossible. While the few, that manage to afford an EHR system perform minimal functions with it: such as inputting data into the system, updating the information, viewing it later and deleting it. These operations are referred to as CRUD (create, read, update, delete) operations.

The general problem prevalent is that data available can’t be used and the ones which can be used are not used. This problem must be solved so that data can be made available. Once data is available, it can be analyzed to get useful information and then the health agency can respond immediately to threats such as Ebola.
1.3 Existing Solution: DHIMS 2

1.3.1 System Overview

The District Health Information System 2 (DHIS2) is a free and open-source web-based application developed by the University of Oslo in Norway. This platform is currently being used by 60 countries around the world. In 2012, Ghana’s ministry of health decided to integrate this system into their health sector. So, the University of Oslo and health Information Officers at the Center for Health Information management (CHIM) worked on customizing the DHIS 2 to be used in Ghana. The resultant system known as District Health Information Management System 2 (DHIMS 2) is currently used across 216 districts in Ghana.

Figure 1.1: Front-end interface of the DHIMS 2
1.3.2 Problems DHIMS 2 solves

Integration of DHIMS 2 into Ghanaian health sector provides all district health administrations in Ghana with an information management system which they could easily use to collate data from health facilities around each respective district and supply the information to the higher levels in the hierarchy as shown in figure 1.2.

DHIMS 2 also allow the users create their own custom entry fields to replicate the original hardcopy format. With this feature, the Ghana Health Service (GHS) can assure they are still receiving the same type of information they formally received using the paper-based approach. The district health administration can also generate various analysis on the data that has been collated over time.

Finally, because the system is a web-based application, it is independent of any desktop or PC that will be used. This also enhances portability because the DHIMS 2 platform can be accessed on tablets and smartphones anytime and anywhere.

![Diagram of how Ghana health activities are represented in DHIMS2]

Figure 1.2: Flow of data collated via DHIMS 2
1.3.3 Challenges with the system

The DHIMS 2 has automated most of the processes required in public health surveillance but its implementation is not a foolproof solution to the problem. Three major challenges identified with the DHIMS 2 are stated below:

- The DHIMS used by the district health administrators is not integrated into the health facilities (hospitals, clinics, maternity home etc.). This requires that the various facilities would collate their data (via in-house Management Information System or Hardcopy record) and send it to their district health administration to input into the DHIMS 2. This causes a lapse in the relay of information; hence diseases cannot be monitored in real-time.

- The DHIMS 2 is a web-based application and therefore relies on internet connectivity to input data and view stored data. If there is no internet access, they cannot view or input any information at all.

- The DHIMS 2 requires personnel to conduct statistical analytics and data visualizations on the platform. This is a drawback in a disease outbreak scenario, because the system relies on a human to conduct analysis before it can identify that an outbreak has occurred. This time that is spent on analysis of the disease cases can be reduced or even eliminated.

1.4 Objectives

The objective of this paper is to document the development of a new system, without the drawbacks of the DHIMS 2.
Therefore, the proposed solution is a Disease Surveillance System (DSS). A scalable web-based application that health facilities would use to provide their reports for it to be accessible by health administrators at the district, regional and country level. Behind the scenes, the system would use this data to constantly detect possible outbreaks given a certain threshold with the use of a probabilistic model. After detecting the threat, the system would alert the health administrator at the level (district, regional, or national) where the outbreak occurs. With this system, the health administration can easily acquire and analyze data, which would aid in making informed decisions when responding to disease cases.

1.5 Scope of Project

Ideally, the system is designed to be used by a health administrative agency like the GHS to assist the task of Disease surveillance.

Figure 1.3: High-level representation of the project
The figure above shows that the data provided by the district health facilities into a shared database. As this data is fed to the database, the second system has mutual privilege to extract information & disease-related records such as; patient’s symptoms, diagnosis, time of diagnosis, medication/prescription etc. from the hospitals’ database. This means that the district, regional & national level administrator can all see the information as it is being recorded in real-time, bridging the lapse that the DHIMS 2 creates.
Chapter 2: Requirement Engineering

This chapter provides more detail about the solution, its users and how it would be used in a typical scenario.

2.1 Requirement elicitation

The design requirements for the system were elicited from a random sampling of hospital administrators and secretaries. In an interview setting, they explained how they would regularly file reports to their district health administration, at monthly intervals, irregular intervals and sometimes on request. The hospital administrators were chosen to conduct this elicitation process because they represent the hospital enterprise and handle the records and reports.

The random sample was not substantial enough to draw out all necessary requirements, so the elicitation process continued through secondary research. The researcher reviewed technical publications on public health surveillance and disease surveillance in Ghana to extract information on how the DSS ought to function.

2.2 Requirements analysis

After analyzing the data collected from both primary and secondary research, a major theme identified was the need to have accurate information at every point in time. During disease surveillance, the health administrators are dependent on the accuracy of the data they receive to be able to make the right decision during an emergency. An emergency in this context refers to a case of epidemic outbreak, or numerous counts of unidentifiable disease cases. Health
administrators also count on the timeliness of the data. Therefore, it is a major requirement that the system always maintain the most current information at every point in time.

Ghana Health Service, the body which oversees the public health & disease surveillance in Ghana requires that the system should also be easy to implement and use. Existing systems such as the DHIMS 2 required the district health personnel to be given an extensive training on the usage of the software by external instructors from the University of Oslo in Norway. Anytime a new person is on-boarded, he/she must receive training, and this incurs cost. Therefore, a system that is intuitive, easy to navigate and maintain in-house is essential.

2.3 Users

The requirements analysis process aided in distinguishing the users of the DSS into two categories. This means that the DSS will have two different interfaces; One for the district facilities and another for the administrators at various levels of the hierarchy in fig 1.2.

2.3.1 Hospital facility user

The facility refers to the following institutions; Regional Hospitals, Teaching hospitals, District Hospitals, Hospitals, Polyclinic, Health clinic, CHPS, Maternity Home etc. While the user represents the personnel operating the software in any of the facilities mentioned above. Hospital facility user (HFU) in this category record the disease cases in-house. After recording cases, they are to compile reports of different kinds dependent on the disease case they wish to report. The DSS will offer the facilities a platform to submit all necessary and timely report to a database which can only be accessed by health administrators, in this case, the GHS.
2.3.2 Health administrators

The health administrators exist at four different levels as declared by the GHS; Sub-district level, District level, Regional level and National level. As stated in the scope of this project, researcher will only consider the district, regional and national level.

The duty of the health administrator during disease surveillance is to monitor and detect patterns and trends in the occurrence of disease.

- District Level (DL) Administrators: Administrators of a district will only view disease reports from facilities in their district. This report is uploaded by the district facilities.
- Regional Level (RL) Administrators: The administrators at this level will be able to view reports uploaded by all facilities in the region they preside.
- National Level (NL) Administrators: This is the highest level, and the administrator at this level will be able to view all disease cases and reports from all facilities in the nation.

2.4 Functional requirements

The functional requirements for the Health Facility Users (HFU) include:

- Apply: The HFU should be able to send an application to the Ghana Health Service (GHS) which will process the application and respond with an approval status.
- Register: The HFU should be able to register their institution on the DSS platform once application has been approved.
- Login: The HFU should be able to access the DSS platform by logging in with their registration credential.
• Upload case-based surveillance reports: The HFU should be able to report specific diseases as they occur or when they desire to.

• Upload weekly notifiable disease reports: The HFU should be able to file or upload weekly disease case reports.

• Upload monthly communicable disease surveillance reports: The HFU should be able to upload their monthly disease case reports to the platform.

• View disease case reports: The HFU should be able to view previous submissions of disease case reports

The functional requirements for the health administrators include:

• Login: The health administrators must log in with their Admin ID and password before being able to use the application. Admin accounts would be created by the Ghana Health Service for the health administrators which gives them access to the system.

• View disease records: The health administrators should be able to view general disease records from different facilities in the format they were uploaded.

• Generate visualizations: The health administrators should be able to represent their data in any visual form, ranging from simple visualizations like line graphs to more complex ones like circumplex charts.

• Save/Exporting visualizations: The health administrators should be able to save or export the output of the visualizations to their local device.

• Choose threshold for disease alert: The health administrators should be able to select an appropriate threshold which would determine on what condition he is to receive an alert regarding a disease outbreak.
• Receive outbreak alert: The district administrators must receive an alert in the presence of any possible outbreak threat for disease cases recorded by the health facilities.

• Track specific disease: The district administrators should be able to select a specific disease e.g. Malaria and see various occurrences in the district he is conducting disease surveillance.

2.4.1 Use Case Diagram

![Use Case Diagram for the DSS](image)

Figure 2.1: Use case diagram for the DSS
2.5 Non-functional Requirements

The non-functional requirements of the DSS includes:

- Availability: The DSS should be available 24/7 if user’s have internet connectivity.
- Timeliness of data: The DSS must always contain up-to-date information, meaning that health administrators should always have the recent information uploaded by the district facilities.
- Ease of use: The DSS must be intuitive to use so that new personnel can easily take control of the system without needing much training.

2.5.1 Organizational Requirements

Before a HFU can be registered on the DSS, an application will be submitted to the GHS. The purpose of this process is to ensure that the HFU registering onto the system is a legally existing and recognized facility. If this application is approved, the HFU will have access to the DSS to submit disease case reports. If application is declined, the HFU will not have access to the system.

Lastly, the various levels in the hierarchy illustrated in figure 1.2 requires that DL admins will only have access to disease occurrences recorded in their respective districts. The RL admins will also be restricted to diseases recorded by districts in their own region. And the NL admins will have access to all diseases recorded in the country.
Chapter 3: Architecture and Design

This chapter describes and illustrates the various architecture types the DSS requires.

3.1 Architecture overview

The previous chapter states there are two main users, health facility and the health administrators. The health facilities will be restricted to their own platform where they provide the disease records. After the health facility provides their registration credentials, they would be able to view their previous submissions and make new uploads and submissions.

The health administrators would also have a platform where they can view records, process data, run analytics and receive alerts from the system in the presence of possible disease outbreak threat. The system is designed to restrict the access of health administrators at the to only disease records from health facilities that are available to them.

3.1.1 Database brief

The platform for the health administrators and the health facilities share one database. All disease cases reported by the health facilities are stored in this database. The system retrieves the information from the database and presents it in the view of the health administrators.

3.1.2 Activity diagram

The activity diagrams show in fig 3.1 and fig 3.2 illustrate the processes the two categories of users will take while interacting with the system. It will also highlight how the user activities interact with each other.
3.2 System Architecture

The platform provided for the health facilities to upload disease records and the platform for the health administrators to view and perform analytics on the record constitutes the major
functions of the DSS. Another key function of the DSS is to give alerts to the health administrator in the presence of any outbreak threat. It is important to understand this because these functions define the architectural pattern that would be implemented. Therefore, the DSS will use both the Model-View-Controller (MVC) and client-server architecture.

### 3.2.1 MVC Architecture

The DSS is a web application and can be accessed from any device via the web. Information recorded in the system such as the disease records uploaded by the health facilities should not only be viewed by the health administrators in one format but should be able to be presented in other formats. This can be done with the use of an application programming interface (API), but it requires the data, views and actions of the system to be separate components. In this case, the DSS data needs to be separated from every other aspect of the system. The MVC architectural pattern separates presentation and interactions from the system data. This characteristic of the MVC architecture makes it a perfect fit for the DSS. The interactions between the three components of the DSS are shown in Fig 3.2.

**Model:** The model in an MVC architecture refers to domain specific data and business logic. They are responsible for the state of a data during retrieval and storage in a database. The model for the DSS is implemented using mongoose, a JavaScript framework for MongoDB. MongoDB is a document-like model database, whose model is structured as JSON documents that will consist of data provided and viewed by the users of the application. Mongoose requires that each model has a schema which is mapped to a collection. Collections in MongoDB are like tables in a Standard Query Language (SQL) database such as MySQL or MSSQL. The schema determines the shape of the data (document) that will be stored. This document is recorded in the collection.
View: The view in this context refers to the interface of the DSS. This interface is generated via the Embedded JavaScript (ejs) template. This templating engine offers fast compilation and rendering and offers both server JS and browser support. The data contained in the model of the DSS will be structured with ejs’ template tags and HTML to be viewed by the user and rendered using the ejs template. Styling would be implemented with Cascading Style Sheets (CSS) with addition of a few styling frameworks like bootstrap and materialize.

Controller: The controller consists of vanilla (plain) JavaScript and Express (a JavaScript framework for backend web development) functions. The controller also includes an
Application Programming Interface that will give other systems to interface with the data contained in the model of the DSS. Key functions of the controller include:

- Receive input from the view of the DSS
- Process requests such as GET, POST, PUT & DELETE
- Retrieve data from the model and determine the appropriate presentation for the view component
- Handle the routing operations to different endpoints in the API

3.2.2 Client-Server Architecture

In tandem, the system will constantly run analysis on the disease records provided by the health facilities behind the scenes to detect possible outbreak threats. Once the system detects a threat, an alert is sent to either the DL, RL or NL administrator depending on the geographical area which the threat was detected, with details of the threat. This functionality will be achieved with a client-server architecture.

The disease surveillance platform for the health administrators will act as the client component in this architecture while a dedicated server continuously running a machine learning program will be responsible for detecting any outbreak and alerting the user of the threat. This server would analyze data from the DSS database, therefore would constantly retrieve data from it.

3.3 Design Specification

3.3.1 Logical view
3.3.2 Database Architecture

The database will serve as a storage to all the data. As identified in section 3.2.1 the model component of the MVC architecture specifies the structure and logic of data stored in the database.
Figure 3.4: Database design of a segment of DSS
Chapter 4: Implementation

This chapter will focus on the implementation of the DSS including the technical processes and design decisions taken in developing the system. This chapter will build upon the architecture and show how various technologies are used to implement all key functionalities of the system. As explained in Chapter 3, the system has two different architecture to achieve the generalized functions of the DSS. The MVC architecture to support the system which HFUs will use to report disease cases and the Client-server architecture for the system health admins to conduct disease surveillance.

4.1 Web application tools & setup

4.1.1 Web application

The DSS as has been mentioned frequently is an application that runs on the web. The backbone technology of this application is JavaScript, a high-level programming language that runs on the browser. Plain JavaScript alone cannot be used to develop a large project such as the DSS, so JavaScript frameworks and libraries were employed.

4.1.2 NodeJS & Express

NodeJS or Node is an open-source JavaScript run-time environment that can execute JavaScript code outside of a browser. Plain JavaScript cannot run on its own, it requires another application such as a browser like Google Chrome or Mozilla Firefox to run. This is problematic because with plain JavaScript operations can be performed only on the client side and not the server side which extensively means dynamic web content cannot be produced.
NodeJS is open-source and has a community of developers always looking to improve it and make development easier with NodeJS. Express is a NodeJS framework developed by members of this community which builds upon NodeJS and offers more tools and functionalities. The DSS was created using the Express framework making it an Express application.

4.2 DSS - Report Platform

The report platform of the DSS facilitates uploads and recording of disease cases by HFUs in facilities all over the country.

4.2.1 Setting up the Express application

The application was first created as a development package and not a production package. The app was setup on a localhost server 3000 for development and testing purposes. Express offers a range of functionalities for back-end to front-end development. The frameworks and tools used in setting up the application included:

- Express – The backbone of the Disease report platform
- Mongoose – A JavaScript framework for MongoDB to handle server-side storage requirements such as storing and retrieving persistent data. The mongoose framework is connected to a MongoDB hosting platform, mLab where the documents (data) would be stored in collections.
- BodyParser – A middleware that allows Express easily access the contents of a response to API requests such as POST.
- Embedded JavaScript (ejs) – A JavaScript template engine that lets you write JavaScript embedded in HTML that makes producing dynamic content easy.
- Path – Allows for working with file directories and paths.

4.2.2 User application process

Only registered health facilities can have access to the DSS report platform. Before the HFU can be successfully registered, they’ll have to apply. The information required in this application process includes details of the hospital and the appropriate legal document showing proof of the facility’s existence in Ghana. The application platform is made available to them when they visit the application for the first time.

Figure 4.1: Application/Registration page for new HFU
A model and schema are created for the information the user provides. After the user has filled the input fields and clicks the submit button, a user controller, which is a JavaScript program retrieves the data from the input form, creates a key-value pair with the Schema representation in JSON format and makes a post to the MongoDB hosting platform, mLab. It is important to note that any data stored in MongoDB is referred to as a document. This specific document that is posted will have additional data that represents if it has been approved, denied, or pending (i.e. awaiting decision) the GHS.

The application details are sent to an admin of GHS who will review the documents and application details. The admin can either select approve or decline. If declined, a mail is sent to the applicant’s e-mail address stating the application has been declined. If approved, the application status data in the document is changed.

Also, an e-mail is sent to the applicant’s address containing a unique log in link with a generated login ID and default password that would be changed later.

**4.2.3 Upload/Create Disease Case Report**

Once logged in, the user can choose one out of four report formats

- Case-based surveillance reports -
- Case-based report for several cases which have occurred during a short period
- Weekly notifiable disease reports
- Monthly communicable disease surveillance reports
Figure 4.2: Report format selection page

After a desired format is selected, the user can either choose to create a new report and input data into the fields manually or upload an excel workbook (.xlsx). Although the reports are different, the model and schema are the same for all four report types. If a user decides to create report manually or upload an .xlsx file, the Express application fires the diseaseCaseController.js script which contains a program which extracts information from the submitted report that will match the data required by the disease model and schema.

The output of the diseaseCaseController.js is used to create a MongoDB document in the structure of the schema and is stored in the MongoDB host database in a DiseaseCase collection.
4.2.4 View Submitted Reports

In addition to creating and uploading reports, as stated in the financial requirements, the HFU must be able to view disease case reports that have been submitted previously. When the HFU clicks the records tab on his dashboard, a diseaseControllerRouter.js program will send a get request to an endpoint to retrieve all reports that have been stored by the user making the request in the MongoDB database. The middleware, BodyParser which was discussed about in the beginning of this chapter will be useful here. When the request is made with or without search parameters, there is a response and the data in the response will be parsed as a JSON document. The BodyParser which has already been instantiated during setup makes the content of the response easy to access and work with.

4.3 DSS – Surveillance Platform

The health administrators under the GHS will use the surveillance platform of the DSS to monitor the disease cases that have been recorded by health facilities around Ghana. The surveillance platform was developed as a client-server architecture and thus was implemented differently than the report platform of the DSS.

4.3.1 Surveillance platform Setup

Like the report platform, the surveillance platform makes use of Express, BodyParser, ejs, Path & Mongoose and is setup the same way.
4.3.2 Health Administrator Access Control

Health administrators using the surveillance platform exists at 3 levels; District Level (DL), Regional Level (RL) and National Level (NL). As explained in chapter 2, these levels provide restriction to the disease records which the health administrators at the various levels can access.

Personnel assigned to the levels will be provided with the credentials that give them access to monitor disease cases recorded by health facilities that fall under their jurisdiction. This access control is achieved with some Express functionalities. The AdminUser schema is defined with certain properties. The properties responsible for facilitating access control are assignmentLevel, and assignmentLocation. The assignmentLevel property contains what level the user has been assigned to; National, Regional or District, and the assignmentLocation property contains the name of the location the admin user is assigned to. For e.g. {assignmentLevel: ‘Regional’, assignmentArea: ‘Takoradi’}. Both schema property values are used to query the DiseaseReport collection to retrieve only the disease reports they are to monitor. When the admin user logs on to the surveillance platform of the DSS, the adminUserController.js get’s the id of the user that is signed in and queries the AdminUser Collection to retrieve the assignmentLevel and assignmentArea values. These values are used as extra parameters in making a GET request to the DiseaseReport collection to make a cross-reference. The response contains a JSON document of all DiseaseReport documents which the logged in admin user should access.
4.3.3 Admin User Dashboard

After an admin user is successfully logged in, the first interface he encounters is his dashboard. The dashboard contains navigations to Disease Reports, & Visualizations, Map view & Settings as shown in figure 4.3.

The ejs template engine is used to create a fixed navigation bar at the top of the page and a side-bar at the left side of the page. The body of the web application is a dynamically created layout. In 4.3.2 an explanation is given on how the system retrieves the disease report. After the JSON document is retrieved, each disease report is summarized and displayed dynamically in a table as shown in figure 4.3 above.
4.3.4 Monitor Specific Disease Cases

In the navigation bar at the top of the dashboard, there’s a dropdown at right where the admin user can either choose to monitor all diseases or a specific disease. The adminReportsController.js has an event listener listening for a change in event state of the select list (dropdown) element. When an admin user clicks the dropdown and selects a disease like ‘malaria’ as shown in the figure below, the diseaseCaseController.js the event listener is informed of this change and fires a function that makes a query to the DiseaseReport Collection and responds with all disease cases related to malaria.

When the admin user switches to the Map view page, points appear on the map representing every disease case for malaria that has been reported by HFUs in the region the logged in admin user is monitoring. The maps and map functionalities are generated with Mapbox GL JS, a JavaScript library for rendering maps. A mapViewController.js script uses the id of the logged in admin user to get the assignmentLocation value. It creates a map canvas for this area. And uses the latitude and longitude properties of each disease report document to create points on the map.

4.3.5 Visualizations

The admin users are also able to create visualizations from the disease reports to derive more meaning from the data. Chart.js, an open source HTML5 based JavaScript chart was used to implement the visualizations. The admin user can create a limited number of visualizations and charts. The chartjs library is simple and easy to use and didn’t require concrete implementation from.
4.3.6 Epidemic/Outbreak detection

The admin user can create visualizations from the disease reports, by using charts and plots the user may be able to identify patterns and trends at face-value. The user can also export the data in csv format and use other statistical software to analyze this data. Per the functional requirements, the admin user must receive notifications from the system in the presence of an outbreak threat/possibility. This means that the system must be able to detect outbreaks. Over the years, research has been conducted on finding outbreak detection algorithms. Outbreak detection algorithms looks for differences between expected value and the observed value of a surveillance time series. If the difference between the values exceeds a threshold, the algorithm may have detected an outbreak [7]. One of the globally used algorithms is the Non-adaptive log-linear regression algorithm. This algorithm could not be implemented for this project as it requires several years of historic data which was not available. In place of this algorithm, the C-family of outbreak detection algorithms. There are three algorithms in the C-family, C1, C2 & C3. Amongst these three algorithms, C2 is commonly used because of its moderate sensitivity [7].

If the diseaseMonitorController.js gets a test statistic of above 3, it will immediately send a notification to the logged in admin user, who can inspect the details of the reported threat. 3 is the threshold used in the Early Aberration Reporting System (EARS) and this value was derived empirically [7].

In addition to being notified of threats using the outbreak detection algorithm, the admin users can also choose when to be notified and what reason to be notified for. In the settings tab, the admin user can choose to receive threat notifications via the C2 algorithm or put their custom disease case threshold and be alerted when that threshold is exceeded.
4.4 Summary

This chapter has highlighted the tools, technologies and open source resources used to implement the system. The next chapter documents the testing process and results.
Chapter 5: Testing

5.1 Overview

This chapter documents the process of testing the DSS as units, components and a whole integrated system. It also documents the details of a user acceptance testing as well as all the results of the test.

The objectives of the test include checking for:

- System performance – speed and efficiency
- Ease of use
- Security
- Correctness of data

5.2 Development Testing

Development tests are conducted by the developer(s) of the system. During development testing, the system was tested in units, then as components and finally as a fully integrated system.

5.2.1 Unit Testing

During the unit tests, functions in the JavaScript programs were tested individually such as individual API routes, models and controllers.

Below is a table documenting some of the tests.
Table 5.1: Showing a portion of unit tests conducted

<table>
<thead>
<tr>
<th>Unit Test</th>
<th>Testing input</th>
<th>Expected Results</th>
<th>Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. mongoose.connect()</td>
<td>MongoURI: “mongodb://&lt;username&gt;:&lt;password&gt;@ds115595.mlab.com:15595/drs-users”</td>
<td>Connected</td>
<td>MongoDb connected…</td>
</tr>
<tr>
<td>Description</td>
<td>The connect function is provided a unique URI to connect to the mLab hosting database. Test passed succesfully</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2. router.post() -&gt;</td>
<td>{</td>
<td>Status code: 200</td>
<td>Status: 200 OK</td>
</tr>
<tr>
<td>localhost:3000/api/users</td>
<td>k</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Description</td>
<td>This test is an api POST request to a route that stores the JSON document in a collection. The test was passed successfully</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

5.2.2 Component Testing

Component testing requires that units or entities are categorized and tested as a component. In this section, four components were tested.

5.2.2.1 User application process

Table 5.2: Table showing component test of user application process

<table>
<thead>
<tr>
<th>Step</th>
<th>Process description</th>
<th>Input/Action</th>
<th>System response</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application</td>
<td>HFU account attempted application</td>
<td>Input – Health facility details and document Action – Submit</td>
<td>System receives application and redirects to admin. (Approval of</td>
</tr>
</tbody>
</table>
application was simulated.) Sends unique log in link and credentials to HFU

<table>
<thead>
<tr>
<th>Step</th>
<th>Process description</th>
<th>Input/Action</th>
<th>System response</th>
</tr>
</thead>
<tbody>
<tr>
<td>Log in</td>
<td>Attempted log in with unique log in URL received</td>
<td>Input – log in credentials received; HFU id &amp; password</td>
<td>Log in Attempt successful</td>
</tr>
<tr>
<td>Test Result</td>
<td>Component test was successful</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

5.2.2.2 Disease case report process

<table>
<thead>
<tr>
<th>Step</th>
<th>Process description</th>
<th>Input/Action</th>
<th>System response</th>
</tr>
</thead>
<tbody>
<tr>
<td>Select disease report format</td>
<td>The Logged in HFU selected disease report format</td>
<td>Action – Click event</td>
<td>System acknowledges click event and loads disease report page.</td>
</tr>
<tr>
<td>Select format type</td>
<td>HFU must either select the option to upload .xlsx file or create report manually. User selected file upload</td>
<td>Action – click event</td>
<td>System acknowledges click event and</td>
</tr>
<tr>
<td>Upload file</td>
<td>User uploaded .xlsx file whose fields cannot be watched with the disease schema property.</td>
<td>Input – File upload Action – submit event</td>
<td>Upload failed: .xlsx rows and column do not correlate with fields.</td>
</tr>
<tr>
<td>Test Result</td>
<td>Partially failed</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

5.2.3 System Testing

System testing requires all components to be integrated together and tested. Dummy accounts are first created for all possible users of the systems. 4 HFU accounts are created, 2 DL admins are created, 1 RL admin is created and 1 NL admin is created as well. The users were made to perform all possible functionalities including intentionally wrong actions to see how the system will react. The test showed that the system was able to manage data flow, access control and multiple requests.
Chapter 6: Conclusion

6.1 Introduction

The goal of this project was to develop a system that will facilitate real-time analytics and disease surveillance in Ghana. This system provides health facilities with a platform to submit disease case reports to the Ghana Health Service administrators responsible for disease surveillance. It also provides the health administrators with a platform to monitor disease occurrences that are recorded by the health facility as at when it is reported through the DSS.

6.2 Limitations

6.2.1 Database Attacks

The DSS is designed to be used in the health industry, therefore whatever data is stored is delicate. The DSS uses MongoDB, a NoSQL database. Although the DSS can manage basic security concerns such as data flow control and unauthorized user access, it may not fare well against attacks targeted to the database such as automated NoSQL database injection attacks. It will take a broader knowledge of NoSQL technology to develop a more secure system.

6.2.2 Analytics & Limited Data Visualizations

It is through visualization and analysis of disease reports that health administrators who monitor disease can detect and identify patterns and trends. The surveillance personnel conducts analysis such as complex statistical tests to derive meaning from the data they retrieve from the health facilities. The DSS does not provide such options for analysis. The DSS also has few visualizations as compared to what disease surveillance personnel do with other statistical software such as Excel, R and Strata.


6.2.3 Device and Internet connectivity constraint

Arguably this is the most substantially challenging of all limitations. The DSS provides a solution to the problem of inefficient disease surveillance, but most facilities in Ghana can neither have nor can afford the resources required to use the DSS. The DSS is a web application, therefore it can be launched from any device with a browser application. In most cases, devices such as Desktops or laptops. Another constraint is the need for constant internet connectivity which can be costly. This constraint has the potential to discourage health facilities from using the DSS and revert to the traditional method of reporting the diseases.

6.3 Future works

In addition to the challenges and limitations described above there are other areas that require improvements and presents an opportunity for addition of more functionalities in the future.

6.3.1 Offline availability

Like most web applications, the DSS requires constant internet connectivity to always stay online. Constant internet connectivity comes at a heavy cost, as well as fast and reliable internet connection. In situations like this, the DSS may be able to store files on the user’s local storage while not connected to the internet. The user will still be able to use the application, but with limited functionality till internet connection is restored.

6.3.2 Improved analytic and visualization

The surveillance platform of the DSS will include a much more substantial analysis options and more data visualizations.
6.4 Conclusion

The DSS provides eliminates the traditional pen-paper approach to storing information and allows health facilities store their disease report records digitally. The health facilities can also submit this report to the health administration at any time. The DSS also eliminates the delay for the health administrators when it comes to receiving the reports. With the surveillance platform of the DSS, health administrators do not have to worry about organization data and delay, they receive the report immediately it has been uploaded by health facilities. In an emergency, the health administrators will have the luxury of time to make informed decisions.

The DSS was developed with limited time and resources yet it shows it is very promising and has the potential to change the way disease surveillance is conducted in Ghana, and possibly West Africa. It gives agencies, both public and private an opportunity to invest in technologies such as this which can give us a fighting chance if we ever must deal with another tremor such as Ebola.
References


