

Web-Based Dengue Prevention System
For the
Office of Medical Officer of Health

A.L.Mohamed Mafahir

Registration No: 169320H

Faculty of Information Technology
The University of Moratuwa.

February 2019

Web-Based Dengue Prevention System
For the
Office of Medical Officer of Health

A.L.Mohamed Mafahir

Registration No: 169320H

**Dissertation submitted to the faculty of Information Technology,
University of Moratuwa, Sri Lanka for the Partial Fulfilment of the
requirements for the Master Degree of Master of Science in
Information Technology.**

February 2019

Declaration

I declare that this thesis is my own work and has not been submitted in any form for another degree or diploma at any other universities or other institutions of tertiary education. Information derived from the published or unpublished work of others has been acknowledged in the text and list of references are given.

ALM. Mafahir

(Name of Student)

.....

Signature of student

Date :

Supervised by

Mr. Saminda Premaratne

(Name of Supervisor)

.....

Signature of Supervisor

Date :

Acknowledgments

It is a great honor that I take this opportunity to express my gratitude to everyone who helped me throughout this project and guided me towards the successful completion of the project.

First of all, I would like to extend my sincere thanks to my project supervisor, Mr. Saminda Premaratne, who gave the proper guidance and constant supervision to me despite his busy schedule.

My special thanks go to the staff of the MOH office, Kinniya who was very helpful and supportive during the requirements for the collection phase. Especially, I would like to thank Dr. M.H.M.Ajith, who is a Medical Officer of MOH office Kinniya.

I wish to express my gratitude to the MSC Coordinator of the University of Moratuwa and the project Examination Board of Information Technology Faculty for all the guidance given during the project development throughout the MSC program.

I would also like to thank all the lecturers at the University of Moratuwa for providing valuable knowledge throughout the MSC program, Moreover, I am grateful to my parents for their encouragement and Truly my humble thanks go to my colleagues and friends for supporting and expressing their views on the project and to complete this project successfully.

Also, I take this opportunity to thank the University of Moratuwa who imparted their latest knowledge and technology to students who searched for the knowledge.

Abstract

Office of Medical Officer of Health (MOH) is a government organization which is governed by the Ministry of Health. The primary operation of MOH is protecting people from diseases and uplifting peoples health. Now Dengue fever has become a major threat to our country. Kinniya is also prone to this threat. MOH office is very concerned to protect people from Dengue fever. All PHII of MOH Office goes around their relevant areas to inspect, to give awareness and to provide information. If the Hospital informs the MOH of any diagnoses of dengue patients over the telephone the and MOH officials have to collect information by visiting the hospital. MOH officer needs to analyze dengue-affected area and patients to take important decisions. They use manual systems to carry out their day to day activities. This project focuses on to speed up activities. There are some computerized systems to implement these kinds of activities such as www.dengue.lk, www.ndcu.lk. In these systems, a user can get a current analysis of dengue patients, national events on dengue, important messages, etc.

The "Web-Based Dengue Prevention System" is built to overcome the above-mentioned problems such as inefficiency, time-consuming, redundancy of data and unavailability of timely information of the current manual system. This system includes Patient Management, Complain Management, News update Management, Blog Management, Staff details, Map Management, and Report Modules.

The system has adopted the Model View Controller (MVC) Architecture and Object-Oriented Techniques. Unified Modelling Language (UML) was used for analysis. Server-side scripting language Hypertext Pre-Processor (PHP) has been used to build the system and Apache has been used as the web server. MySQL has been used to handle databases. WBDPS is responsive and browser friendly website. This report presents the outcome of the requirement gathering (system analysis), system design, implementation, testing and evaluation in a structured and logical manner to give the reader clear insight of the work carried out throughout the project duration. At the bottom line, this system will ensure that day to day activities in MOH office are handled efficiently and effectively.

Table of Contents

Chapter 1	1
Introduction.....	1
1.1 Introducing and Motivating problem.....	1
1.2 Background of the Project	2
1.3 Detailing the gap in current knowledge	2
1.4 Describe the Research Problem	2
1.5. Aim and Specific Objectives	4
1.5.2. Main Aim	4
1.5.3 Objectives	4
Chapter 2.....	5
Review of others work	5
2.1 National Dengue Control Unit.....	5
2.2 National Dengue Update by Ministry of Health	6
2.3 Analysing the Current Manual System.....	6
2.4 Summary of Chapter	8
Chapter 3.....	10
3.1 Function of the project	10
3.1.1 User Management Module.....	10
3.1.2 Patients' Registration Module.....	10
3.1.3 Complaints to MOH Module	10
3.1.4 Point out affected place module.....	10
3.1.5 SMS/E-Mail alert handling module	10
3.1.6 Maintain blog to communicate with the public.....	11
3.1.6 News updates regarding dengue fever module	11
3.1.7 FAQ and suggestions on dengue fever	11
3.1.8 Auto-generate reports and charts module	11
3.2. Technology Adapted	11

Chapter 4.....	13
Analysis and Design	13
4.1 System Design	13
4.1.1 High-level use-case of the system.....	13
4.1.1.1 Manage user privileges	14
4.1.1.2 Patients' details module	15
4.1.1.3 Patients' Registration sequence diagram	15
4.1.1.4 Complaints to MOH.....	16
4.1.1.6 SMS alert to MOH and area PHI	17
4.1.1.6 Maintain Blog to communicate with people	17
4.1.1.7 News updates regarding dengue fever	18
4.1.1.8 FAQ regarding dengue fever and suggestions	19
4.1.1.8 Generate reports and charts for decision making	19
4.2 Research components Design and Methods.....	20
4.2.1 MOH	20
4.2.2 Hospital	20
4.2.3 Public	20
Chapter 5.....	24
Implementation	24
5.1 Implementation Environments	24
Software	24
Hardware.....	25
5.2 Reused Model	25
5.3 Network Implementation	26
5.3.1. Network implementation for SMS sending module.....	27
5.3.1. Google Map Developer	27
Chapter 6.....	28
Evaluation	28
6.1 Evaluation and Testing	28
6.1.1 Test plan and test case.....	29
6.1.2 User evaluation	29

6.1.3 Matching and analysis.....	30
Chapter 7.....	32
Conclusion & Further Work	32
7.1 Critical Evaluation	32
7.2 Lessons learned.....	32
7.3 Future improvements	33
References.....	34
Appendix A.....	35
Appendix B	39

List Of Figures

Figure 1: Major Components Involved.....	3
Figure 2: National Dengue Control Unit	5
Figure 3: National Dengue Update by Ministry of Health	6
Figure 4: The Current Manual System.....	7
Figure 5: Use-case of the system	13
Figure 6: Manage user privileges.....	14
Figure 7: Patients' details module.....	15
Figure 8: Registration sequence diagram.....	15
Figure 9: Complaints to MOH	16
Figure 10: Point affected the place	16
Figure 11: SMS alert to MOH and area PHI.....	17
Figure 12: Blog to communicate with people.....	18
Figure 13: News updates regarding dengue fever.....	18
Figure 14: FAQ regarding dengue fever and suggestions	19
Figure 15: Generate reports and charts for decision making	19
Figure 16: Network Implementation.....	26
Figure 17: an SMS sending module.....	27
Figure 18: Map Generator.....	27
Figure 19: User evaluation.....	30
Figure 20: Matching dengue patient year wise	31
Figure 21: User Login Page	39
Figure 22: MOH Home Page	39
Figure 23: User Registration.....	40
Figure 24: Patients Registration.....	40
Figure 25: View Registered Patients.....	41
Figure 26: Add New Doctors and GS Division	41
Figure 27: affected places	43
Figure 28: Reports and Charts	44

Figure 29: Complaint to MOH.....	45
Figure 30: Administrator.....	45
Figure 31: User privileges.....	46

List of Table

Table 1: Common functions.....	35
Table 2: Manage patient details module	36
Table 3: Complaint to MOH	37
Table 4: Point affected place.....	37
Table 5: News update regarding dengue.....	38
Table 6: FAQ and Suggestions	38

Chapter 1

Introduction

1.1 Introducing and Motivating the problem

Today many government institutions are using manual systems to implement their activities. Office of the Medical Officer of Health also uses a manual system to carry out their day to day activities. Now dengue fever is a major problem in our country and government is very keen on this issue. This work has to be done by MOH offices collaboratively with hospitals and the public.

This system will help MOH to speed up their day to day activities and make a connection between MOH, hospitals, and the public. This system will help them to transfer messages and information quickly among the staff of the MOH.

Office of the Medical Officer of Health (MOH) is a government organization which is governed by the Ministry of Health. MOH's primary role in protecting people from diseases and uplifts their health conditions.

Now Dengue fever has become a challenge to them. In Kinniya also the situation is the same. MOH needs fast processing to implement its activities. Now MOH uses a manual system to implement their activities. This project focussed on to make MOH's activities fast to prevent people from dengue fever.

The problems of the existing manual system are listed as follows:

- MOH uses paperwork to store patients' details
- MOH does not have a proper analysis of dengue patients
- MOH does not have a proper system to transfer important messages
- MOH does not have a proper system to give awareness to the public continuously
- MOH does not have a proper method to get complaints from the public

1.2 Background of the Project

Now many government and non-government organizations are working together to eradicate dengue from this area but they don't have proper IT related systems to implement. MOH and others are randomly choosing places to identify dengue-affected areas. There are some web sites related to dengue problems but they can give only awareness regarding dengue yet they don't have a proper system to reduce dengue. I visited the MOH office to observe their current activity and then I discussed with MOH and his staff about this. I identified they don't have a proper system to identify the affected areas exactly, their activities are lower when compared to the spreading of dengue because of their manual system.

1.3 Detailing the gap in current knowledge

My purpose of developing this system is to increase efficiency and effectiveness of transferring important messages, keeping important records, giving important news on dengue mosquitoes and fever and also my system will observe continuously, conditions of patients, such as, whether they have recovered, still hospitalized, transferred to other hospital or died.

1.4 Describe the Research Problem

Office of Medical Officer of Health (MOH) is a government organization which is governed by the Ministry of Health. The primary operation of MOH is protecting people from disease and uplifting peoples' health. Now Dengue fever has become a major threat to our country. Kinniya area is also prone to this threat. MOH office is very concerned with protecting people from Dengue fever. All PHII of MOH office goes around their relevant areas to inspect, to give awareness and to provide information. The Hospital informs to MOH if they diagnose dengue patients to inform over the telephone and MOH office has to collect information by visiting hospitals. MOH officer needs to analyze dengue-affected areas and patients to take important decisions. They use a manual system to carry out their day to day activities. This project focuses on to speed up activities and to reduce spreading dengue.

In this system, three major components get involved to work together. Their functions and method of implementation are as follows

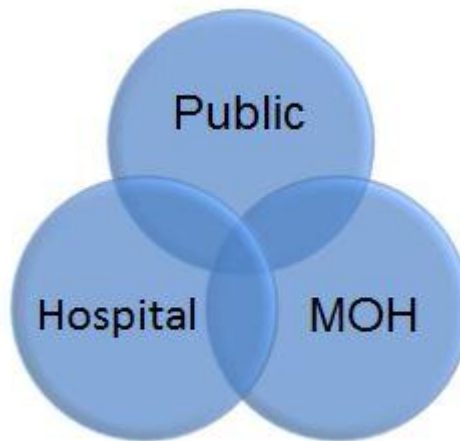


Figure 1: Major Components Involved

MOH

Manage patients' detail coming from the hospital and give awareness to the people.

Its functions are

- Receive and maintain patients' details coming from the hospital
- Receive messages from hospital
- Pass messages to PHII through the system.
- Accept leave request through the system.

Hospital

- In this process, the hospital receives details from patients
- Enter details of the patients to the system.
- Doctor diagnose the fever condition.
- Submit the report to the hospital.

Public

- Visit the hospital to consult the doctor and give full details

1.5. Aim and Specific Objectives

1.5.2. Main Aim

Avoid dengue spreading through identifying dengue-affected areas

1.5.3 Objectives

1. Manage user privileges and patients details
2. SMS / E-Mail alert to MOH and area PHI regarding Dengue patients
3. Maintain Blog to communicate with the public and update news regarding dengue
4. FAQ regarding Dengue Fever and Suggestion regarding Dengue Fever protection
5. Generate report and charts for decision making (Using projected area map)
6. Dengue fever spreading conditions in the project area and Compline to MOH
7. Point out affected patients in the project area using real Google map Developer

Review of others work

2.1 National Dengue Control Unit

National Dengue Control Unit (NDCU) was established in 2005 as a decision taken by the Ministry of Health following a major DF/DHF outbreak occurred in 2004. It is responsible for the coordination of control and prevention activities related to dengue at a central level between different stakeholders. NDCU integrates key functions and processes within the ministry of health into one seamless application. It includes modules, called Components which can run by itself or when integrated with other Components can provide a solution to manage, which are mentioned below. The following figure 2 shows the home page of NDCU

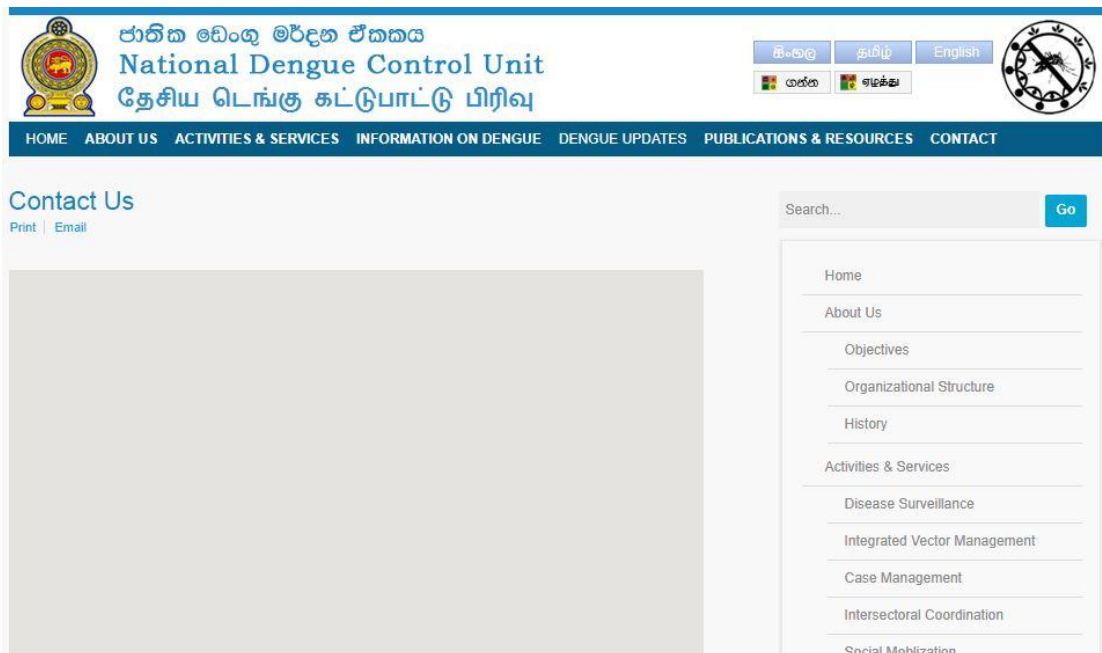


Figure 2: National Dengue Control Unit

2.2 National Dengue Update by Ministry of Health

It is responsible for the coordination of control and prevention activities related to dengue at a central level between different organizations, therefore, government initiate this system to publish national wide events and analysis



Figure 3: National Dengue Update by Ministry of Health

2.3 Analysing the Current Manual System

Requirement analysis sets the stages for gathering information about the current and the existing problem of the system. a Cleared and detailed analysis of the existing system is necessary to develop a new quality information system. The existing manual system contains three areas which are MOH office, Hospital and Public. Each subsystem has been analyzed carefully to understand their functions and how they are related to each other.

In the current manual system dengue patients, details are entered by hospital manually and it will be sent to MOH office as a paper document. The hospital sends alert messages about detected dengue patients through the phone. PHI visits patients location and analyses the situation with MOH officer. The following figure 4 depicts functions of MOH office.

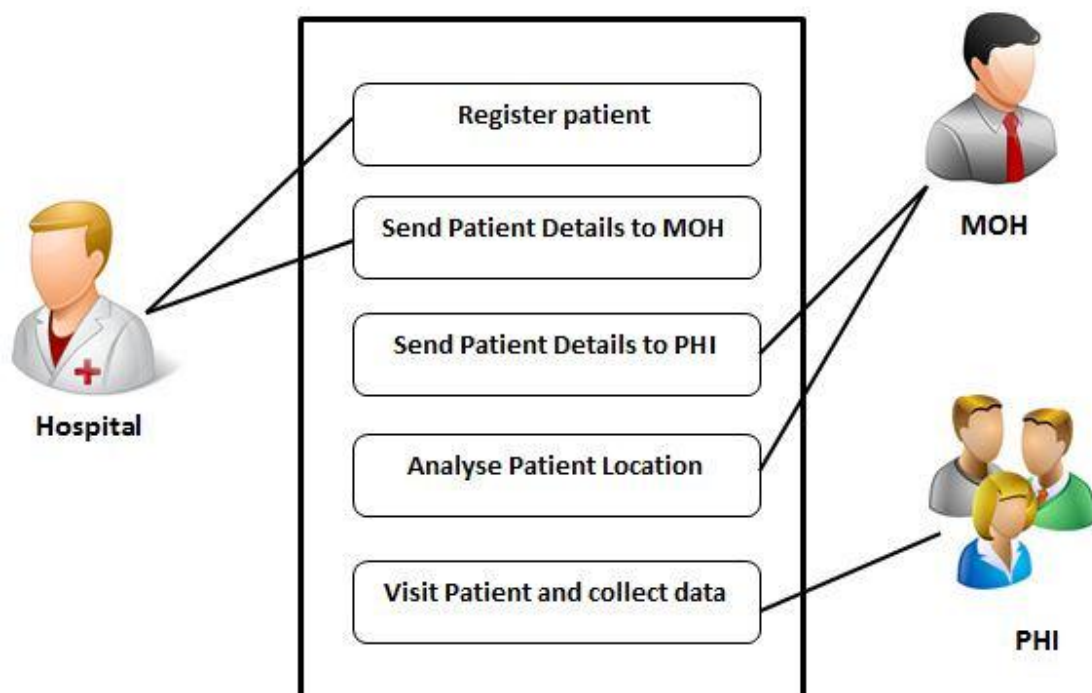


Figure 4: The Current Manual System

Last year this area met approximately 16 deaths and more than 100 of dengue patients who were admitted. This lasted long' for several months' that time caring and providing good treatment were major problems this situation encouraged me to find good IT related solution, therefore, I visited MOH office and Hospital to observe the current process of dengue prevention activity. There' I identified that they don't have a proper system to catch exact affected places and communication between them was very poor and there is no option to get efficiency reports for decision making. Therefore I was pushed to design an IT related system to support and to reduce the complexity.

Dengue is a national issue. It is responsible for the coordination of control and prevention activities related to dengue at a central level between different organizations. Therefore, the government initiates this system to publish national wide events and analysis.

Hence there is the number of websites' national wide regarding dengue control but they can give only awareness for the public but there is no any initiation for collaborative work among hospitals' MOH and public for dengue-related activities. Communication among these people should be entitled to efficiency work because they must work together.

Being considered of all these issues, it is decided to build a new system incorporating all functionalities require to be attended by the client since, by the proposed system is expected to have access from any place irrespective of the location, it was decided to adopt a web-based system solution for this project. Due to the web-based nature of the system, several benefits can be obtained. Reasons for choosing a web-based system solution are

- The client particularly requests a web-based system.
- Transferring important data through the internet.
- The system would be platform independent.
- Easy deployment.
- Can be implemented on client-server. architecture and use a web-browser as the client interface.

2.4 Summary of the Chapter

According to my literature review, I found several dengue-related systems which are only providing certain information not providing efficient information to take decision or action

Some of the dengue-related systems are providing only awareness to the public not to consider regarding the prevention method of dengue

Also, these systems are only used by the departments, not public such as DCU, Ministry Health, etc. because of that public are not take much consideration about those systems.

Advantages of the System

- Instruct the public regarding dengue fever using article, pictures
- It's publishing dengue-related tips to follow-up to the public regarding dengue
- Instruct the public to clean their areas to prevent dengue
- Action to be taken if the dengue patients are identified

Disadvantages of the System

- There is not a collaborative dengue-related system like MOH, Hospital, Public
- No User-Friendly
- There are no fully computer-based activities for dengue prevention
- Need computer knowledge to use

Technology Adapted and Approach

3.1 Function of the project

MOH's list of functional requirements was identified during the fact-finding process. For defining identified requirements clearly, they have been categorized into modules according to their dependencies.

3.1.1 User Management Module

The proposed system has a different access level. This system should allow users to access the system according to their user level

3.1.2 Patients' Registration Module

The system should allow to register patients' details and managing the patient list including updating, removal of the existing patient and discharged patients. The system should also have the capability to send important notifications to the MOH through various means such as email, SMS. Hospital initiates and maintains this module.

3.1.3 Complaints to MOH Module

The system should allow the public to send complaints through the system to MOH regarding environmental issues that cause to spread dengue mosquitoes and sender will receive a reply for their relevant complaints

3.1.4 Point out affected place module

The system should allow viewing an affected place in a map with the number of patients affected according to the area. This feature permitting the officers to trace the location of the affected place in a map mode

3.1.5 SMS/E-Mail alert handling module

The System should allow sending important notifications from hospital to MOH officer regarding dengue patients through E-Mail and SMS.

3.1.6 Maintain blog to communicate with the public

The System should allow the administrator to post important messages, news, and events about dengue. The administrator can update, edit and delete the post. The public can view blog post and it gives facilities to leave a comment on the post

3.1.6 News updates regarding dengue fever module

The System should allow the administrator to post important news to the public regarding dengue fever, environmental problems, national events, symptoms, government announcement, etc. The administrator can update, edit and delete news and make links to useful sites and newsletters

3.1.7 FAQ and suggestions on dengue fever

The System should allow the public to learn some solutions for common problems and frequently ask questions. Administrator uploads this FAQ to clarify problems of the public regarding dengue fever and its problems

3.1.8 Auto-generate reports and charts module

The system should provide timely and accurate reports needed by MOH and the other users about the dengue patients. The MOH office and hospitals have many resources that have to be managed appropriately to gain maximum efficiency. The system should allow allocating resources owned by the MOH for tasks like providing medicines, patients details, summary reports provided by the system can be used to take decision making.

3.2. Technology Adapted

Here is the technology adapted system that should have, in order to achieve the outcomes expected by the MOH office.

Recoverability – The ability to restore the system after a failure has occurred.

Extensibility – The system should be extensible for future improvements.

Privacy – Appropriate measures have to be taken to protect personnel details from unauthorized access.

Reliability – The system should be trustworthy to its users.

- Robustness – The system should have the ability to cope with errors arising while it is in a production environment.
- Response – The system should work fast enough for its users.
- Usability – The system should be easy to use and learn for its target user.
- Accessibility – The system should be accessible by using all major web browsers.
- Availability – Most of the time the system should be in functioning condition.
- Backup – The system should allow making copies of data, which may be used to restore the original after a data loss.

My purpose of developing this system is to increase efficiency and effectiveness of transferring important messages, keeping important records, giving important news on dengue mosquito and fever and also my system will observe continuously conditions of patients such as whether they have recovered, still hospitalized, transferred to other hospitals or died

Analysis and Design

4.1 System Design

4.1.1 High-level use-case of the system

The following figure 5 depicts the high-level Use Case Diagram of the proposed WBDPS system

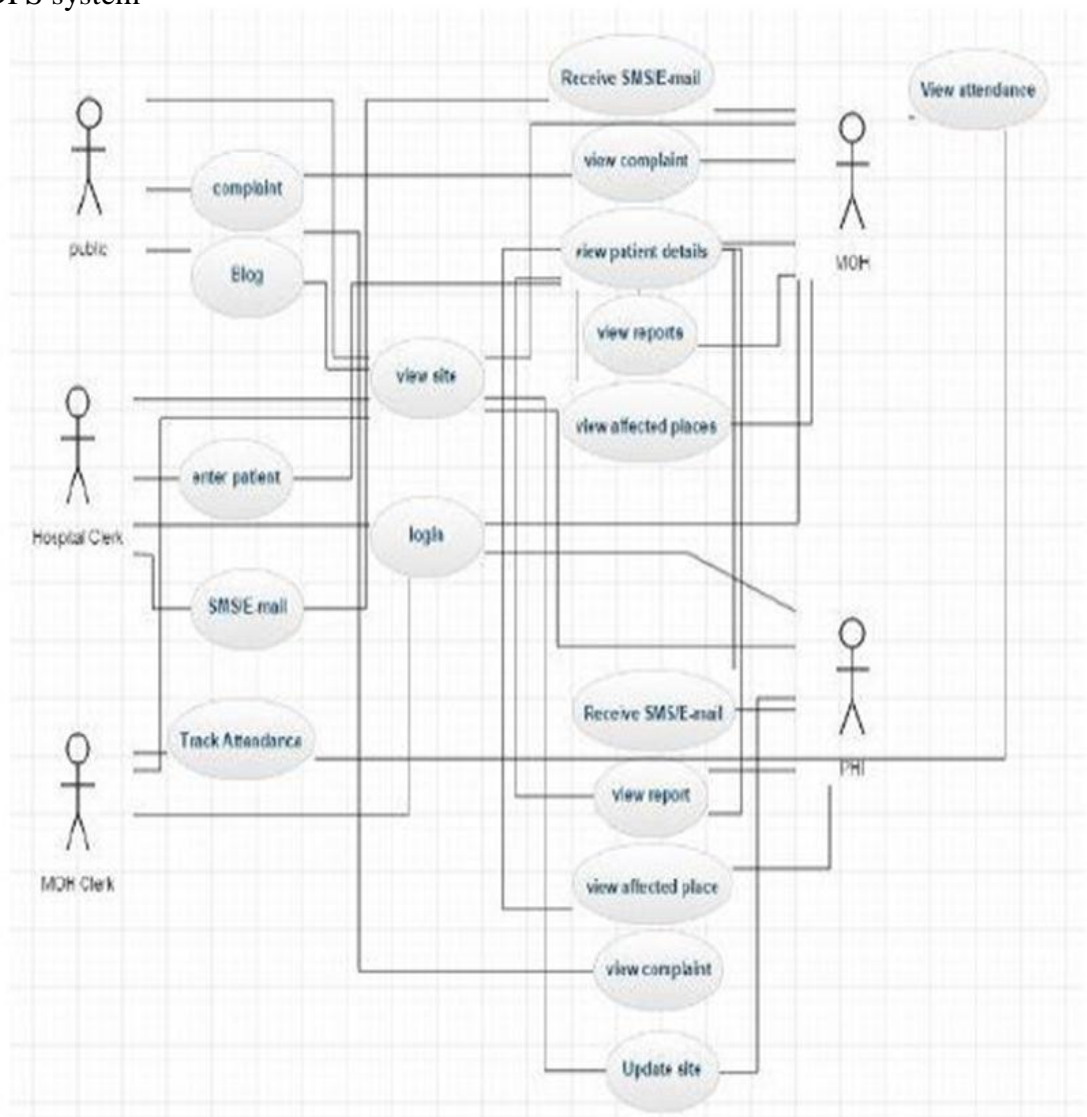


Figure 5: Use-case of the system

The web-based system is introduced to help the users of the Dengue Prevention System to do the tasks easier by passing important messages quickly and safely and reducing the clerical support to operate the processes and also to help in decision making and concentrate in core activities of the organization.

The suggested solution has been divided into nine modules in order to make development and comprehension easier. The modules are as follows,

- Manage User Privileges
- Manage Patients Details
- Complaints to MOH
- Point Out Affected areas Using Map
- SMS and email Alert to MOH and Area PHI Regarding Dengue Patient
- Maintain Blog to Communicate with Public
- News Updates Regarding Dengue Fever
- FAQ Regarding Dengue Fever and Suggestion
- Generate Reports and Charts for Decision Marking

4.1.1.1 Manage user privileges

This section handles the user privileges of users of the system. MOH, hospital, and PHII have different kinds of access level and can log in into the system. Following figure 6 illustrates the procedure of login.

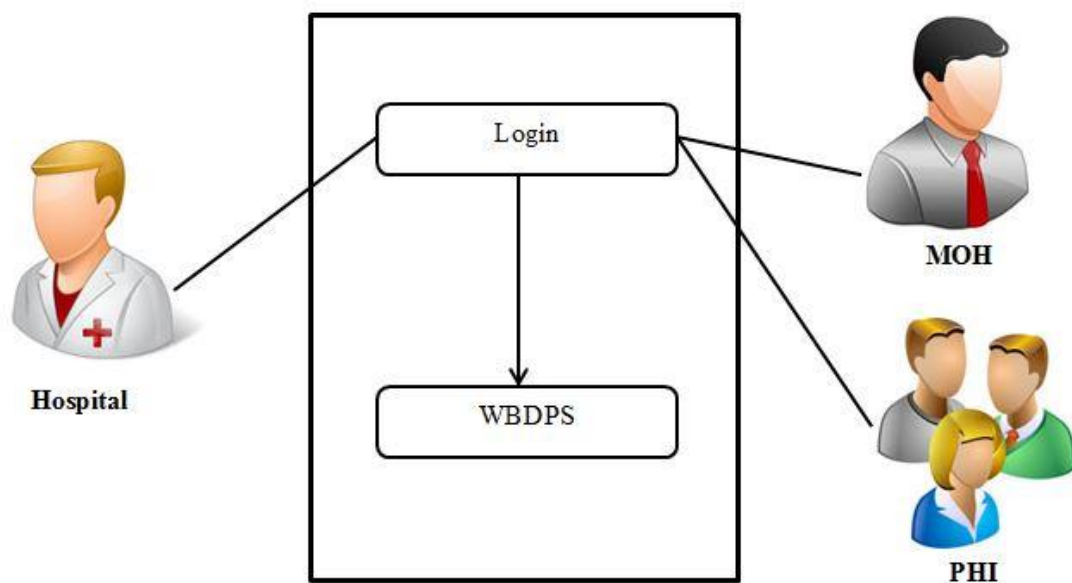


Figure 6: Manage user privileges

4.1.1.2 Patients' details module

Patient details module handle collecting patient details from the patient, the following diagram shows patient details entered by "Hospital" into the WBDPS and details can be accessed "MOH". The following figure 7 illustrates patient details module

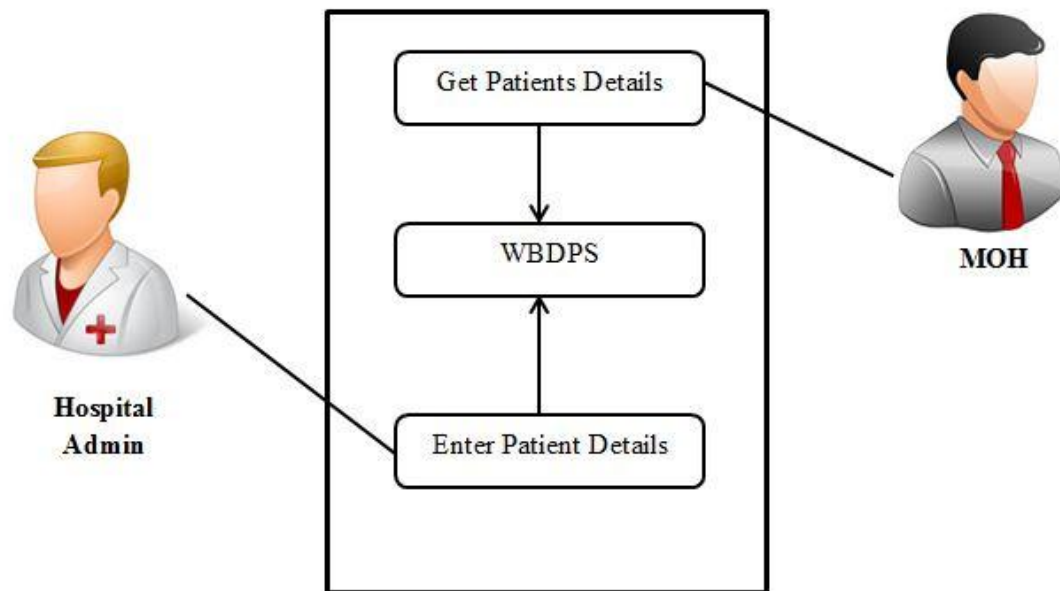


Figure 7: Patients' details module

4.1.1.3 Patients' Registration sequence diagram

Following figure 8 is the Sequence diagram of a patient registration module. It shows that public visit hospital for treatment then the hospital enters detail into the system and it is saved in a database. This can be accessed by MOH.

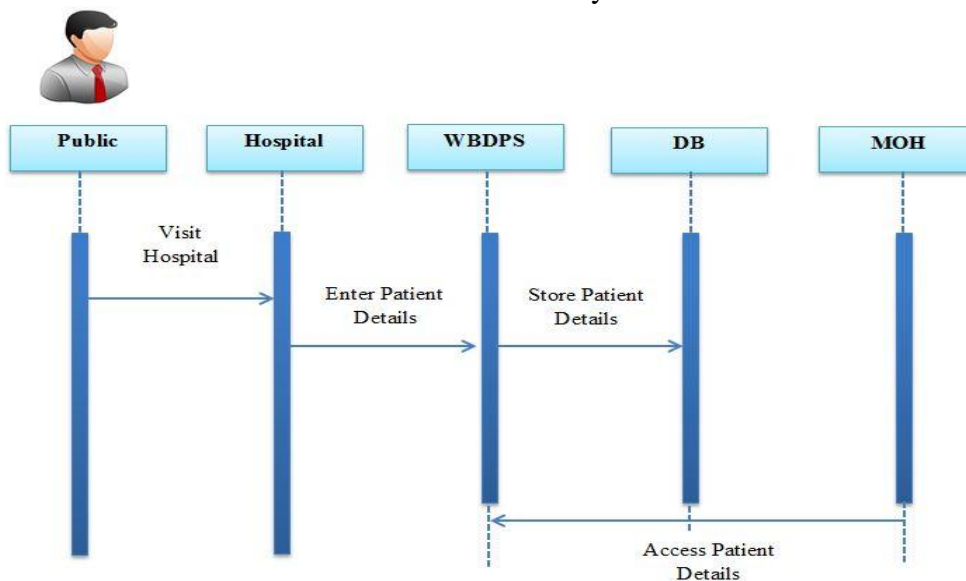


Figure 8: Registration sequence diagram

4.1.1.4 Complaints to MOH

Complaints to the MOH module handles complaints send from public to MOH officer. Complaints can be viewed by MOH and he is the responsible person to send a reply and all PHII can be viewed of the complaints. The following figure 9 depicts the processing of sending a complaint to MOH through WBDPS.

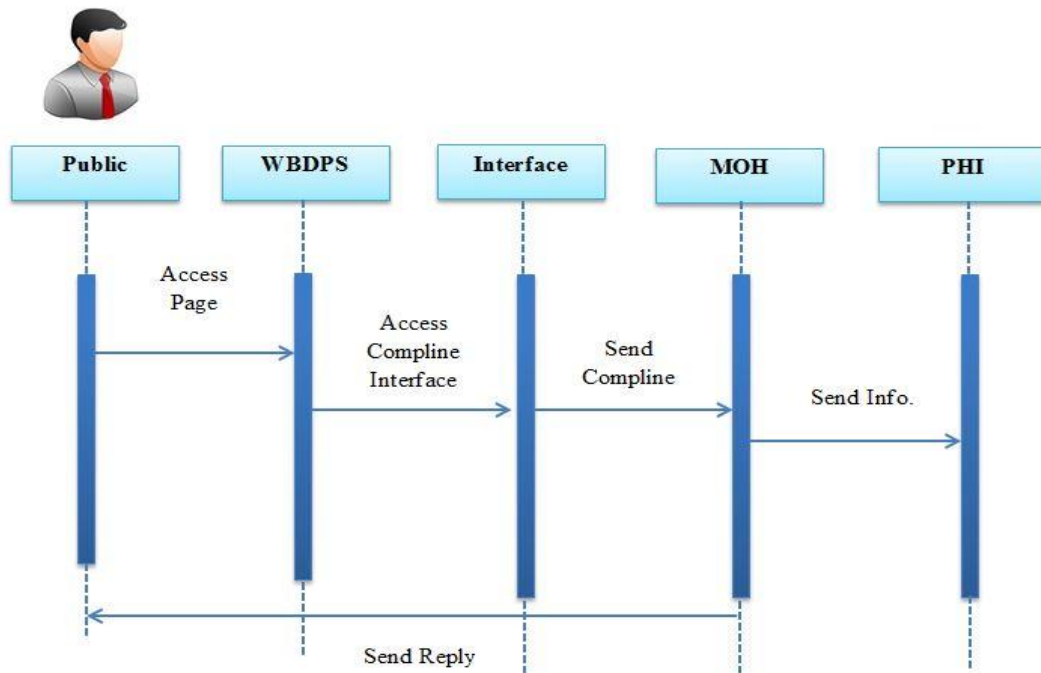


Figure 9: Complaints to MOH

4.1.1.5 Point affected the place

The "Point Affected Place" module handles generating a map from the system of the area covered by Kinniya MOH office with a number of patients relevant to each area. This diagram depicts the way of generating a map.

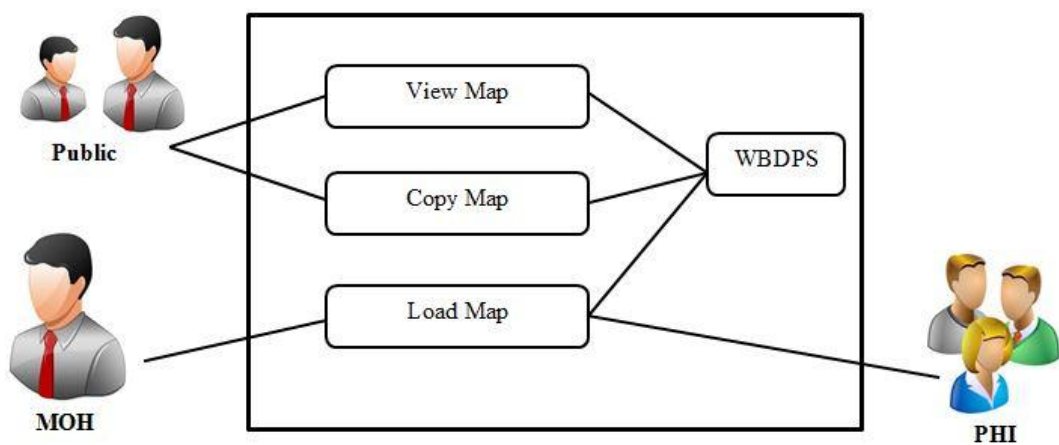


Figure 10: Point affected the place

4.1.1.6 SMS alert to MOH and area PHI

The “SMS alert to MOH and area PHI” module handle sending SMS to MOH immediately when hospital detects dengue patients. The following figure 11 depicts, how WBDPS sends alert to MOH and PHI.

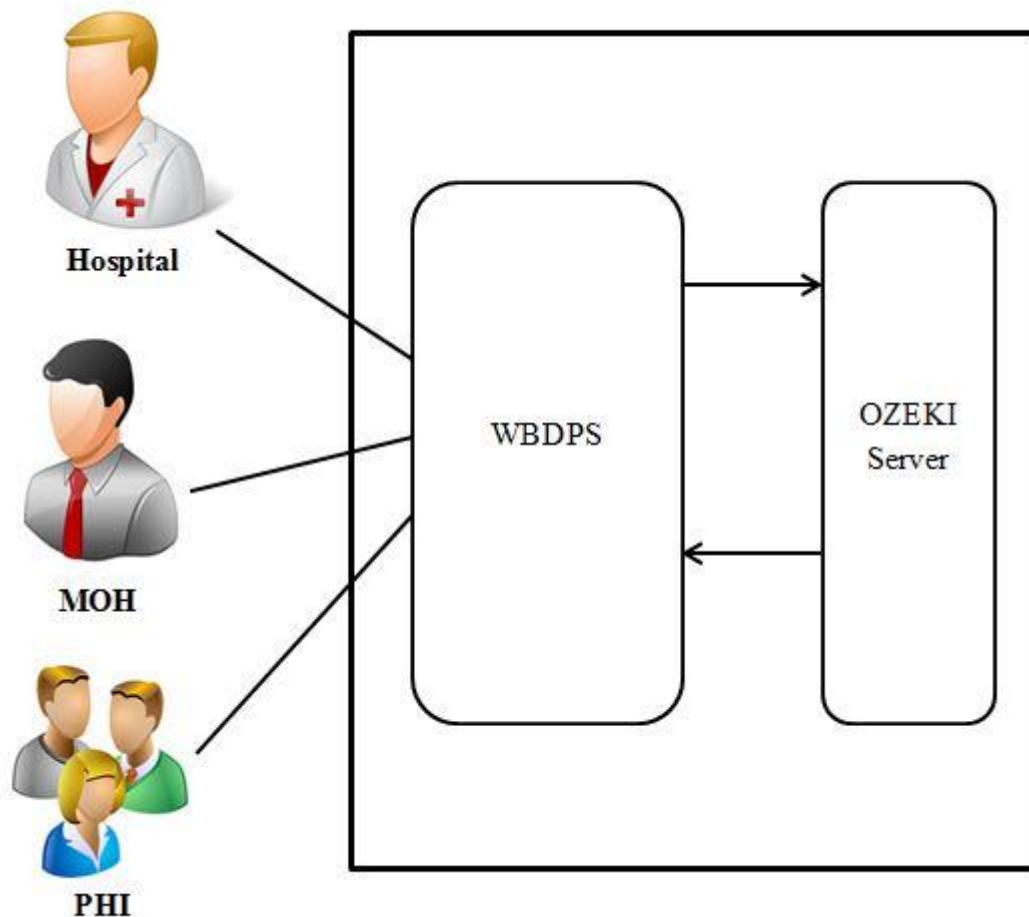


Figure 11: SMS alert to MOH and area PHI

4.1.1.6 Maintain Blog to communicate with people

The "Maintain Blog to communicate with people" module handle to display the blog posted by the administrator, this can be accessed by public and public can leave a comment on uploaded post. This diagram figure 12 depicts the processing of posting blog and how the blog generates from WBDPS.

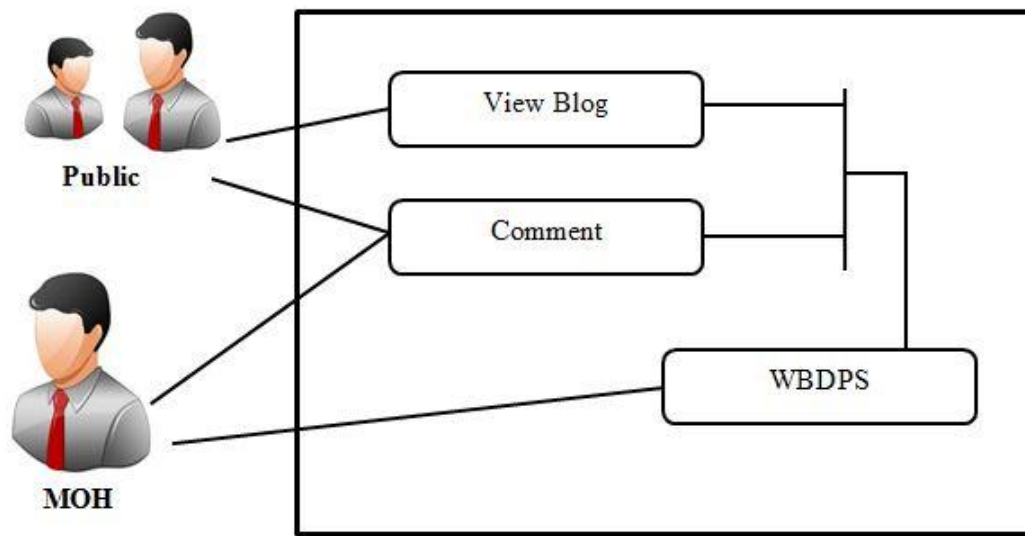


Figure 12: Blog to communicate with people

4.1.1.7 News updates regarding dengue fever

The “News updates regarding the dengue fever” module handle news updating to the system by Administrator. This diagram depicts the processing of updating news to the system and it can be viewed by the public.

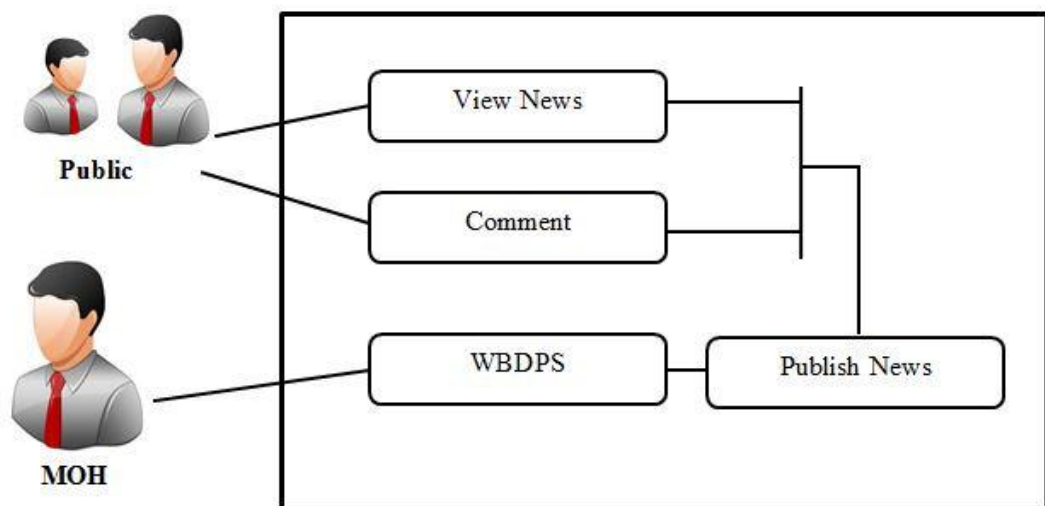


Figure 13: News updates regarding dengue fever

4.1.1.8 FAQ regarding dengue fever and suggestions

The "FAQ regarding dengue fever and suggestions" module handle uploading and updating FAQ to the system. The following figure 14 depicts the processing of MOH upload FAQs to WBDPS and the public can view the FAQ.

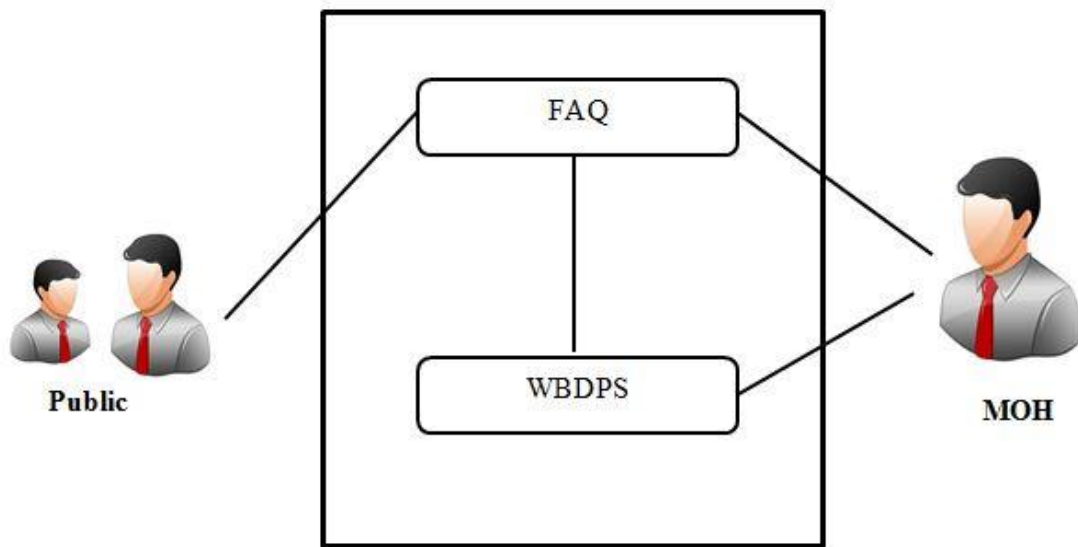


Figure 14: FAQ regarding dengue fever and suggestions

4.1.1.8 Generate reports and charts for decision making

The “Generate reports and charts for decision making” module sends an SMS alert to MOH and relevant area PHI regarding dengue patients admitted to hospital. This will be carried out by the hospital. The following figure 15 shows who can access reports and chart from WBDPS.

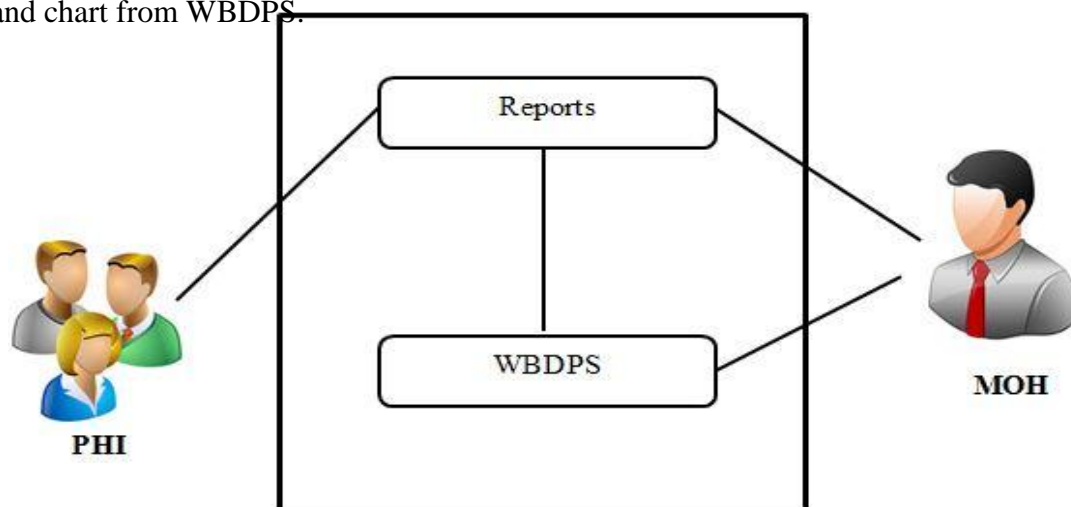


Figure 15: Generate reports and charts for decision making

4.2 Research components Design and Methods

In this system, three major components involved to work together. Their functions and

method of implementation are as follows

4.2.1 MOH

Manage patient details coming from the hospital and give awareness to the people. Its functions are

- Receive and maintain patient details coming from a hospital
- Receive messages from hospital
- Pass messages to PHII through the system.
- Accept leave request through the system.

4.2.2 Hospital

- In this process, the hospital receive details from patients
- Enter details of the patients to the system.
- Doctor diagnose the fever condition.
- Submit a report to the hospital.

4.2.3 Public

- Visit the hospital to consult the doctor and give full details

User management module

This system should allow users to access according to their user level

Patients' registration module

The system should allow registering patient details and managing the patient list including updating, removal of an existing patient, discharged and dying patients. The system should also have Web capability to send important notifications to the MOH through various means such as email, SMS. Hospital initiates and maintains this module.

Complaints to MOH module

The system should allow the public to send complaints through the system to MOH regarding environmental issues that cause to spread dengue mosquitoes and sender will receive a reply for their relevant complaints

Point out affected place module

The system should allow viewing affected areas on a map with the number of patients affected according to the area. This feature permits the officers to trace the location of the affected areas in a map mode

SMS/E-Mail alert handling module

The System should allow sending important notifications from hospital to MOH officer regarding dengue patients through E-Mail and SMS.

Maintain a blog to communicate with the public

The System should allow the administrator to post important messages, news, and events about dengue. The Administrator can update, edit and delete the post. The public can view blog post and it gives facilities to leave comments on the post

News updates regarding dengue fever module

The System should allow the administrator to post important news to the public regarding dengue fever, environmental problems, national events, symptoms, government announcement, etc. The administrator can update, edit and delete news and make a link to useful sites and newsletters

FAQ and suggestions on dengue fever

The System should allow the public to learn some solutions for common problems and frequently ask questions. The administrator uploads this FAQ to clarify problems of the public regarding dengue fever and its problems

Auto-generate reports and charts module

The system should provide timely and accurate reports needed by MOH and the other users about the dengue patients. The MOH office and hospitals have many resources that have to be managed appropriately to gain maximum efficiency. The system should allow allocating resources owned by the MOH for tasks like providing medicines, patient's details, summary reports provided by the system can be used to take decision making.

The system has adopted MVC as its architecture. MVC is a software architecture or design pattern for developing web applications whose fundamental principle is based on the idea that the logic of an application should be separated from its presentation. In other words, the main emphasis is for the separation of the user interface layer or the view logic from the business logic. Adhering to the MVC pattern brings modularity and it enables code reusability. Moreover, the adaptation of the MVC architecture makes the system flexible. That is, it has the ability to adapt to changing needs and future enhancements which results in the maintenance processes an effortless task. The MVC design pattern paradigm is a way of assigning objects in an application in one of the three distinct roles as:

Model – Represents the underlying data and interactions with the database is done through here. It serves as a data access layer where data is fetched.

View – Includes the interface files which are presented to the users with which the user interacts with manages the display of information. Provides a visualization or presentation of the model.

Controller – To handling events that affect the Model or View. The middleware that translates the Model to the View and vice versa. Offers facilities to change the state of the model or to update the data in the model.

5.3 Expected Outcome

- Manage patients' details
- Complaints to MOH

- Point affected areas
- Send SMS / E-Mail alert to MOH and area PHII regarding Dengue patients
- Provide Blog to communicate with the public
- News will be updated regarding Dengue fever
- FAQ option regarding Dengue Fever and Suggestions
- Manage Attendance and leave of Dengue-related staff
- Generate reports and charts for decision making
- Observing and reporting dengue patients conditions continuously.
- Dengue fever spreading conditions in the project area.
- Point out the number of affected patients in the project area map

Implementation

Implementation is the process of converting the system specification into an executable system. Design and implementation specification processes are transforming into an executable program. The ultimate goal of the system implementation is to build a correctly working system, the system was developed using familiarized language and modern development tools. The codes were written, readable and easily understandable with comments. The system will be maintainable in the future.

5.1 Implementation Environments

Software

- CSS (Cascading style sheets) is a style sheet language used for describing the presentation semantics of a document written in a markup language. CSS was to use for design the interfaces more attractive and user-friendly.
- HTML (Hypertext markup language) is the main markup language for creating web pages and other information that can be displayed on the web.
- PHP 5.4 (Hypertext Preprocessor) is widely used general-purpose scripting language that is especially suited for web development and can be embedded into HTML.
- JQuery is the most popular JavaScript library. It is used to implement a precoded date picker model, according to the menu and image transition.
- AJAX which is used to get a date from the database without refreshing the page.
- JavaScript was used for client-side validation.
- Windows 8 operating system was used for the development of the system.
- MySQL 5.5 was used to implement the database.
- XAMPP version 1.8.0, it is including PHP 5.4.4.
- Apache server 2.4.2.
- Netbeans IDE 7.2/Dreamviewer was used for coding.
- Photoshop used for image creations.
- Easy Imagemap Generator

Hardware

- Intel Core i5 2.5GHz processor.
- 4GB RAM.
- 460GB Hard disk.

5.2 Reused Model

Several Modules where existing modules freely available from the web in order to speed the development and get modules that are time tested.

- **Jquery DatePicker** – A jQuery plugin which provides a simple, highly configurable date picker to easily navigate through months and years.
- **PHPMailer** – PHPMailer is utilized to add email functionality to the system. It is a full-featured Email Transfer Class for PHP that provides a package of functions to send an email. The two primary features are sending HTML Email and Emails with attachments.
- **Face box** – Firebox is a jQuery based lightbox which is used to reduce the navigation overhead between pages and to add more interactivity.
- **Data Tables** – It is a plug-in for the jQuery JavaScript library which is equipped with advanced interaction controls over HTML tables. Inbuilt pagination facility and advanced searching ability make this a highly flexible Tool.
- **Am charts** - Amcharts Chart is used as the charting component for this system and is very useful in creating management reports. These are delightful JavaScript charts that work across a wide variety of devices and browsers which helps to create animated, interactive and customizable charts.
- **OZEKI Message Server 6** – It is a powerful, flexible SMS Gateway application that enables to send SMS messages to mobile devices from the computer.
- **Htm2pdf** - The system uses Htm2pdf for reports creation process. It is an HTML to PDF converter.
- **Bootstrap** – Used to design attractive interfaces.
- **Accordion** – Navigation menu for attractive interface.

- **Jquery AutoComplete** – Enables users to quickly find and select from a pre-populated list of values as they type.

5.3 Network Implementation

Since WBDPS is a web-based system, it can be installed in a web server equipped with any operating system due to its platform independent nature. A dedicated web server and a database server can be set up to gain optimal performance. Yet even a single server could be set up for the same purpose too. The system can be accessed via a web browser and an active internet connection should be made available.

The following Figure 16 shows the network architecture of the proposed WBDPS system to implement

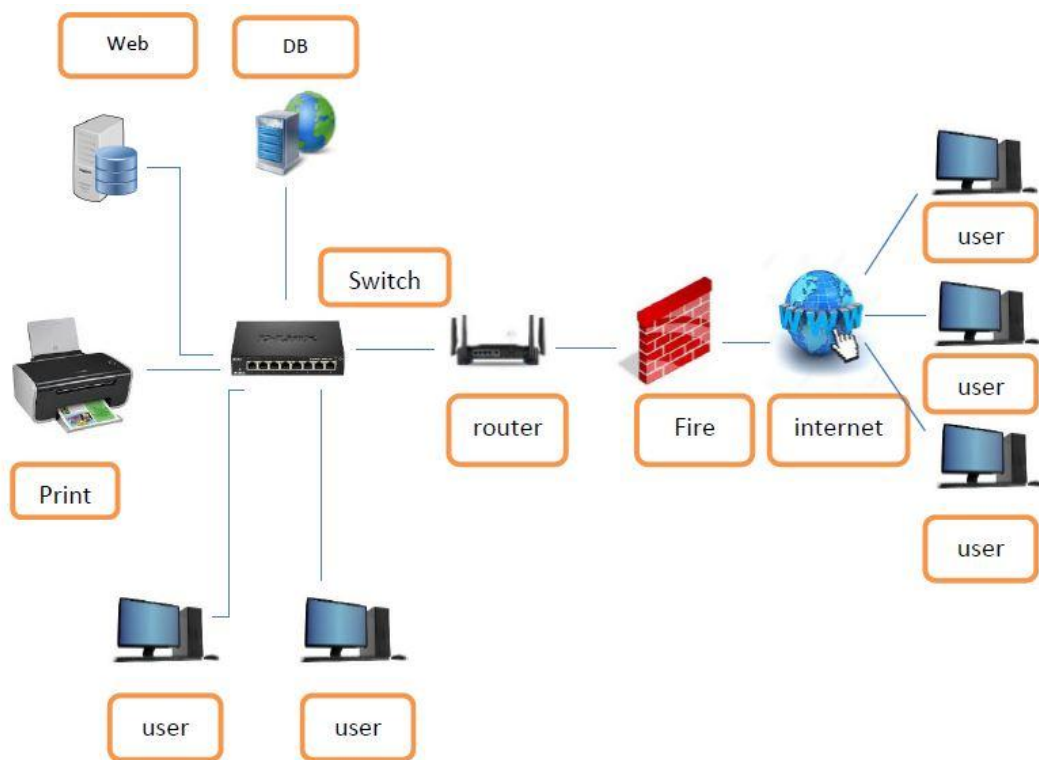


Figure 16: Network Implementation

5.3.1. Network implementation for SMS sending module

The following figure 17 depicts Network implementation for SMS sending module using ozeki server connection with the GSM network.



Figure 17: an SMS sending module

5.3.1. Google Map Developer

Build customized, agile experiences that bring the real world to your users with static and dynamic maps, Street View imagery, and 360° views. Features included: Maps, Street View

The following figure 17 illustrates the image real map loading which creates and point out the reallocation of the given dengue patient details.

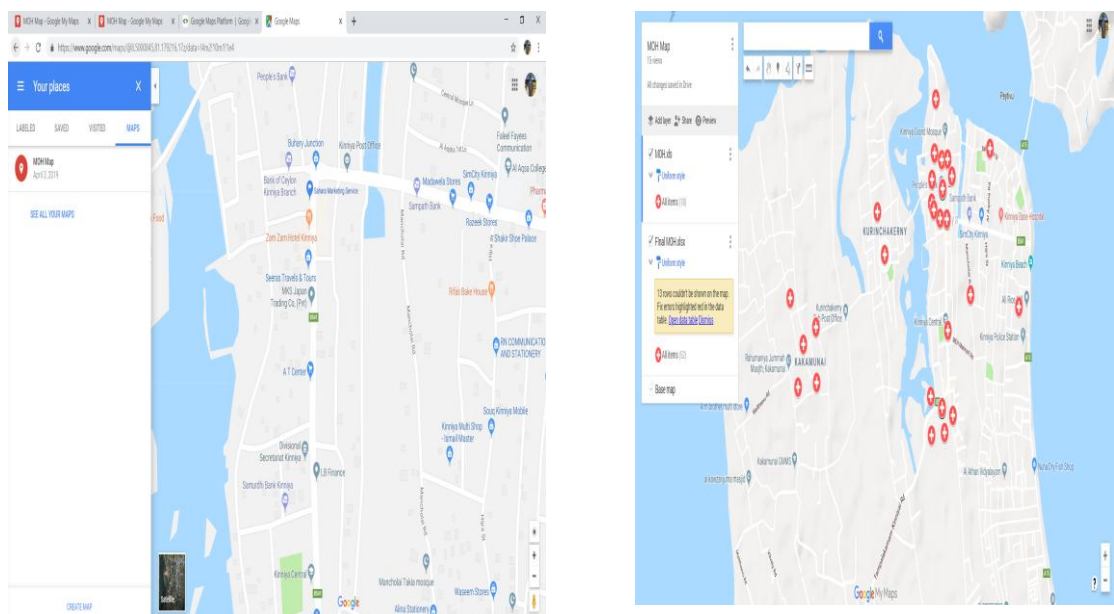


Figure 18: Map Generator

Evaluation

Now many governments, non-government organizations and well-wishers are working together to eradicate dengue from projected area these works are happening fully manual and human activities because of that but they don't have proper IT related system to implement. MOH and others are randomly choosing places to identify dengue-affected areas. There are some web sites related to dengue problem but they can give only awareness regarding dengue, yet they don't have a proper system to reduce dengue. we visit MOH's office to observe their current activity then discusses with MOH officer and his staff through this we identify that they don't have a proper system to identify affected areas exactly their activities are lower than dengue fever spreading because of their manual system

6.1 Evaluation and Testing

System evaluation is a process of determining if the requirements of the users are met and the system is running as expected. It is a process of verifying and validating the system against the users' requirements and real-world data scenarios. Verifying the system includes checking whether all requirements of the users are fulfilled while the validation process checks if all functional requirements of the users are met. Testing is vital as it determines the stability of the system as it will be used by users to take important decisions and they need a valid working system.

The WBDPS was tested in parallel with the development since software testing and implementation are most of the time-interleaved. Moreover, with the adaptation of RUP development lifecycle, it was made possible to test iterative increments of the software. Test strategies define the number of ways of evaluating the system functionalities to ensure that it performs as expected. Different testing techniques were adopted to aid in the testing process.

Both structural and functional testing techniques were used to ensure adequate testing. Structural testing also known as white-box testing is concerned with the internals of a program and aims to exercise the code thoroughly as possible whereas functional

testing is also known as black-box testing addresses the overall behavior of the program and no knowledge of the internal logic of the system is needed to develop test cases.

Unit testing is a form of testing that is done at or near the code level to ensure that the components operate correctly. Once the individual code fragments are unit tested, they ensure that everything has been linked together correctly and to uncover errors associated with interfacing. Moreover, system testing was also performed in a similar environment where the system will be practically used to check the integrated system as a whole. With the adaptation of RUP, system testing is concerned with testing an increment to be delivered to the customer.

Apart from that, another thing that should be paid emphasis is a change or a fix in one part of the system should not cause another portion to break that was previously working. This is achieved through regression testing.

6.1.1 Test plan and test case

Test plan covers all type of testing phases. It was used as a master document throughout the system development to aid in the testing areas. The test plan was developed before the implementation phase. The test case was designed when the test plan was created. A test case consists of data, procedures and expected the result of the system. The cases were designed separately for each function to improve the quality of the test process

Test data was input to the system for the testing the system in the implementation and later real data records were imported to the system.

6.1.2 User evaluation

After the system was implemented a demonstration was done to the client and given the system for a test run. Users who were involved with the test run were given a questionnaire to conduct a survey. According to the results of the survey and the suggestions from the management, some small modifications were needed to be done. Finally, the client was satisfied with the implemented system and agreed that the system had met with the original requirements. All the users' feedback rating were

taken and converted into a graphical evaluation chart. The following figure shows user feedback results.

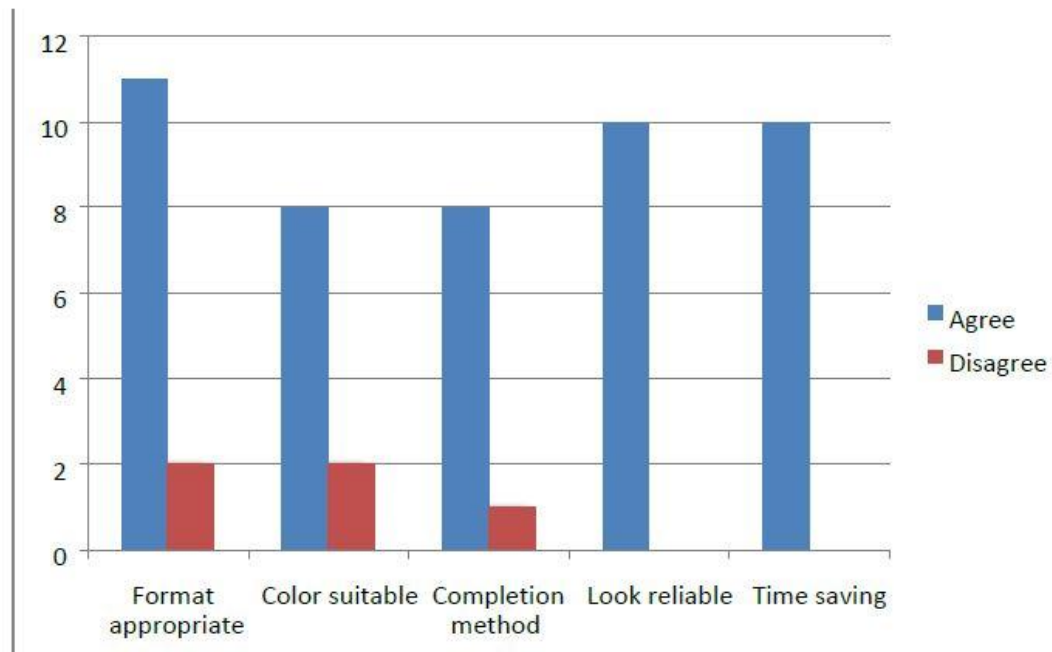


Figure 19: User evaluation

Below in Appendix A-C describes some functions of WBDPS and gives methods of testing and their expected results.

6.1.3 Matching and analysis

The System can be comparing dengue patients with past year dengue record because of that the public and the administration can identify the speeding methods of the dengue between the past year compared with the present year

The system can help the users to take the decision to prevent dengue fever in the heavily affected location which is pointed by the real Google map location

Also through this system MOH, Hospital, Users are can take a decision to prevent dengue fever in areas by using the report, chart and Google map pointed location (Count)

The main supportive of this system is getting patients of relevant target groups who are directly related to the dengue fever because of that the information provided by the system are efficiency and accuracy.

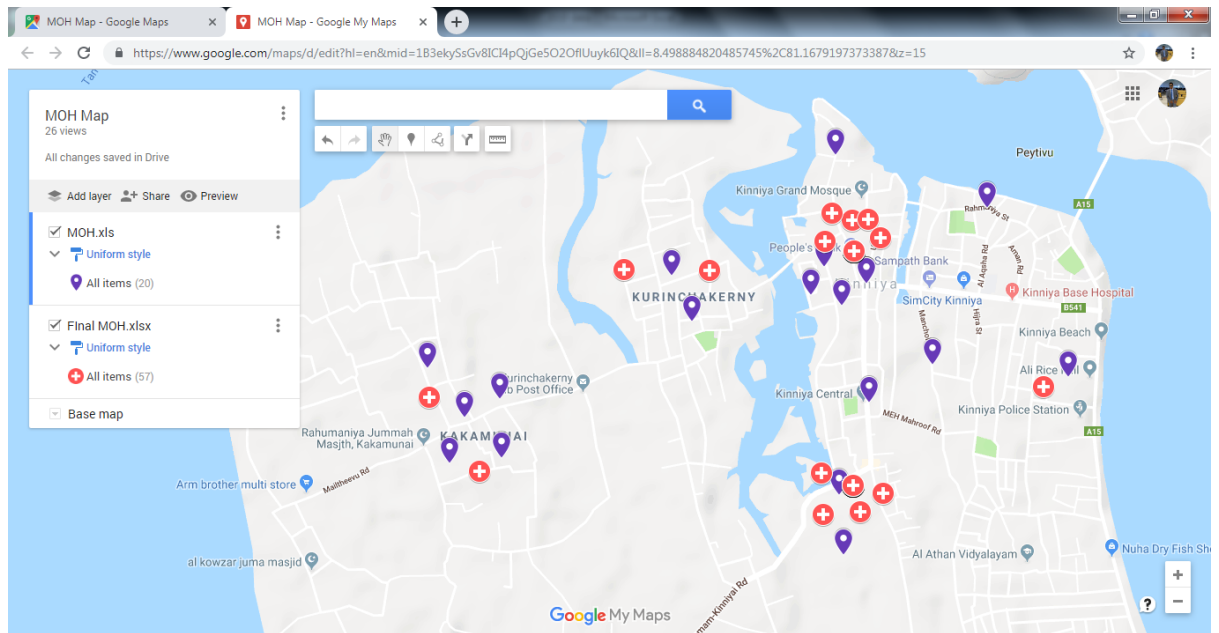


Figure 20: Matching dengue patient year wise

The above Figure 20 matching the data of 2017 and 2018 dengue affected patient locations through DPOS and this figure indicate different symbols for each year data

Conclusion & Further Work

7.1 Critical Evaluation

When compared with manual file systems and developed a web-based system, it was found that there were many unique functions in the new system. Along the various stages of the project, the client requirements were checked and made sure that the requirements had been addressed during implementation phases. This constant checking with the requirements makes sure that the developed system met the goals and objectives that were devised at the beginning of the project. By reviewing the functional and non-functional requirements which were discovered during the analysis phase and checked with the functionalities implemented in the developed system.

The system allows the user to enter, calculate and view relevant data with complete ease. According to the clients' feedback, the lack of paperwork also has motivated the employees on working extended hours. The simple and intuitive user interface was designed and developed to run the office success without difficulty and increase productivity.

7.2 Lessons learned

The knowledge gained throughout the project was really valuable. In addition to this which gave me an outstanding experience of being involved in a complete, Software Development Lifecycle Cycle, starting from the feasibility study to the conclusion of the project. Furthermore, this project gave me a chance to develop the system in MVC architecture. And also it helped me to learn some technologies called AJAX, jQuery, PHP, MySQL and Object Oriented theories. Most importantly this helped me to apply my technical knowledge in a real working environment.

7.3 Future improvements

The system has fulfilled the entire functional and non-functional requirements requested by the client along with additional features and managed to overcome most of the problems in the manual system, some future enhancements are essential to make this as a fully qualified system.

The followings are the future enhancements which are considered to be implemented

The system has fulfilled the entire functional and non-functional requirements requested by the client along with additional features and managed to overcome most of the problems in the manual system, some future enhancements are essential to make this as a fully qualified system.

The followings are the future enhancements which are considered to implement

- Track the environmental polluted area using GPS Technology.
- Add a module and allow management to generate a national wide report of dengue patient.
- Use a digital certificate to improve security.

References

- [1] Dengue.health.gov.lk, "National Dengue Control Unit", [Online]. Available: <http://www.dengue.health.gov.lk>. [Accessed: 05- Nov- 2017].
- [2] Dengue.lk, "Dengue Sri Lanka | ඩි ටිංගු | டெங்கு |Latest Dengue News|වසංගත ඩි ටිංගු | Epidemiology Sri Lanka", [Online]. Available: <http://www.dengue.lk/>. [Accessed: 20- Oct- 2017]
- [3] who.int.int, "World Health Organization", [Online]. Available: <http://www.who.int/csr/don/19-july-2017-dengue-sri-lanka/en/>. [Accessed: 03- Oct- 2017].
- [4] F. Shull, R. Tesoriero, E. Barrenechea and S. Pfleeger, *Software engineering, fourth edition, Shari Lawrence Pfleeger, Joanne M. Atlee*. Upper Saddle River, N.J.: Prentice Hall, 2010.
- [5] P. Kruchten, *The rational unified process*. Boston: Addison-Wesley, 2004
- [6] T. Marston, "The Model-View-Controller (MVC) Design Pattern for PHP", *Tonymarston.net*. [Online]. Available: <http://www.tonymarston.net/php-mysql/model-view-controller.html>. [Accessed: 05- Oct- 2017].
- [7] iamat.org, [Online]. Available: <http://www.iamat.org/country/sri-lanka/risk/dengue> [Accessed: 05- Nov- 2017].
- [8] reliefweb.lk "reliefweb Srilanka: Dengue Outbreak", [Online]. Available: <http://www.relifeweb.int/desater/ep-2017-00086-lka> [Accessed: 05- Nov- 2017].

Appendix A

Common functions

Test No	Test Description	Steps to Test	Expected Results
1	User Interface	If the user enters the invalid domain name	Page won't be displayed
2	User Interface	If the user enters the valid domain name	A page will be displayed
3	Log in to the System	If user enter invalid username and password.	An error message will appear indicating the error
4	Log in to the System	If user enter valid username and password.	The user will be logged in to the system and redirect to the admin dashboard
5	Unauthorized access to a page with login.	A user trying to access the system through URL with login in	The user will be redirected to the login page

Table 1: Common functions

Manage patient details module

Test No	Test Description	Steps to Test	Expected Results
1	Add Patient	Click the "Add " link on the Menu list and click the Save button with a fill of data	Details Added
2	View all patients details	If the user Click "PATIENTS' DETAILS"	Data Table will be displayed with all details
3	View Reports	If the user clicks "View Reports"	The page will be displayed with report type options
4	View monthly report	If the user clicks "MONTHLY REPORT"	The page will be displayed patients details with current Month report
5	View Weather wise report	If the user clicks "WEATHER VICE REPORT"	The page will be displayed patients details with current Month report
6	View Gender wise report	If the user clicks "GENDER VICE REPORT"	The page will be displayed patients details with GENDER report
7	View Ares wise report	If the user clicks "AREA VICE REPORT"	The page will be displayed patients details with AREA VICE report

Table 2: Manage patient details module

Complaint to MOH

Test No	Test Description	Steps to Test	Expected Results
1	Send Mail to MOH	Fill the complaint form	Receive mail to MOH account
2	Send reply mail to the complainer	Click on the reply button and Fill the reply form	Complainer receive a reply to his mail

Table 3: Complaint to MOH

Point affected place

Test No	Test Description	Steps to Test	Expected Results
1	Public view Map of the affected place	Click on affected place button and map will be loaded automatically to the system	Publish map to a system and able to view

Table 4: Point affected the place

News update regarding dengue

Test No	Test Description	Steps to Test	Expected Results
1	Public view News of Dengue fever using the system	Upload News to system	Publish News to a system and able to view

Table 5: News update regarding dengue

FAQ and Suggestions regarding dengue fever protection

Test No	Test Description	Steps to Test	Expected Results
1	Public view FAQ of Dengue fever via system	Upload FAQ to system Publish	A FAQ to the system and able to view

Table 6: FAQ and Suggestions

Test Result

User Login Page

WBDPS Users can log into the System through the following interface figure 17 shown below.

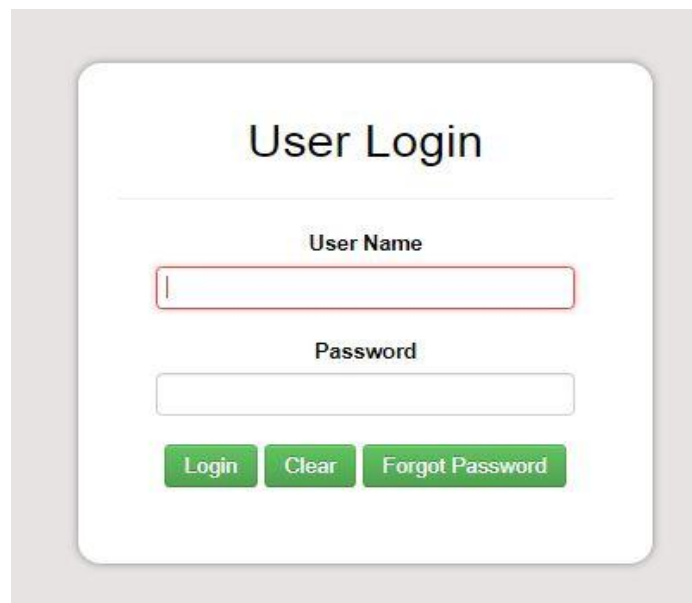
The image shows a 'User Login' form. It has a title 'User Login' at the top. Below the title, there are two input fields: 'User Name' and 'Password'. The 'User Name' field is highlighted with a red border. Below the input fields, there are three buttons: 'Login', 'Clear', and 'Forgot Password'. The buttons are green with white text.

Figure 21: User Login Page

MOH Home Page (Main Page)

The following figure 18 shows the main components of the Project

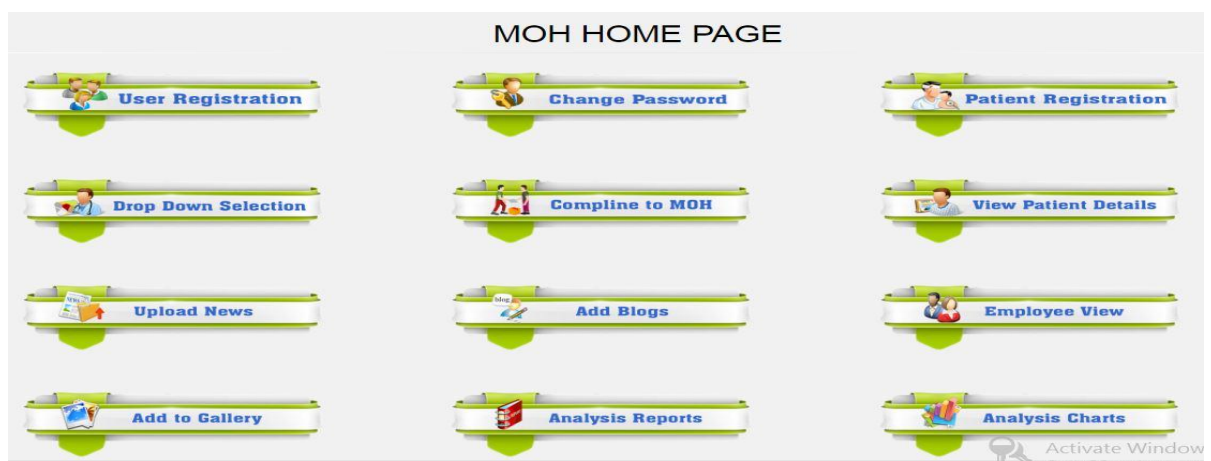


Figure 22: MOH Home Page

User Registration

The following figure 19 describes the way of registering the new user like PHI / MOH / Hospital Admin.

The screenshot shows a web application interface for user registration. A modal form is open with the following fields: NIC, Full Name, Telephone No, Email, and User Type (set to 'Clerk'). There is a 'Choose File' button for the 'Picture' field, which shows 'No file chosen'. A red error message is displayed: 'Notice: Undefined variable: pUserRegPic in C:\wamp\www\MOHapps\viewUserMgt\registeradd.php on line 48'. Below the error message is a 'Call Stack' table:

#	Time	Memory	Function	Location
1	0.0004	252752	main()	..add.php:0

At the bottom of the modal are 'Register' and 'Remove' buttons. The background shows a table of users with columns for ID, NIC, and Name, and buttons for 'ADD USER' and 'Export to Excel'.

Figure 23: User Registration

Patients Registration (New Patient)

The following figure 20 is the place for registering new dengue patient

The screenshot shows a 'Patient Registration' form with the following fields:

- NIC: Eg. 881234567V
- Date of Admission: dd/mm/yyyy
- First name: Ganith
- Last name: Niantha
- Address: Ex Mancholai Kinniya
- Telephone Number: Ex 0776622882
- Date of Birth: dd/mm/yyyy
- Gender: Male
- GS Division: Kaakamunai
- Marital Status: Married
- Name of Next of kin: Ex.ALM. Mafahir
- Relationship to next of kin: Ex Mother
- Consultant name: 45355555554
- Description of symptoms:
- Case Admission Status: ECU

At the bottom are 'Register', 'Clear', and 'VIEW PATIENT DETAILS' buttons. An 'Activate Windows' watermark is visible in the bottom right corner.

Figure 24: Patients Registration

View Registered Patients

The following interface is the place to see all the patients registered into the system



USER REGISTRATION

ADD USER

Export to Excel

Show10entries

Search:

ID	NIC	Full Name	TP	Email	Type	Pic	Action
17	853013811V	Anif Lebbe Mohamed Mafahir	776622882	mafahir@gmail.com	MOH		EDIT
18	863013811V	Mohamed Razeen	776622882	razeen@gmail.com	Hospital		EDIT

Showing 1 to 2 of 2 entries

FirstPrevious1NextLast

Figure 25: View Registered Patients

Add New Doctors and GS Division

The following figure gives permission to Add new doctors and GS division.

[BACK](#)
Enter New Doctor
Name of The Doctor

[Enter](#) [Update](#) [Remove](#)

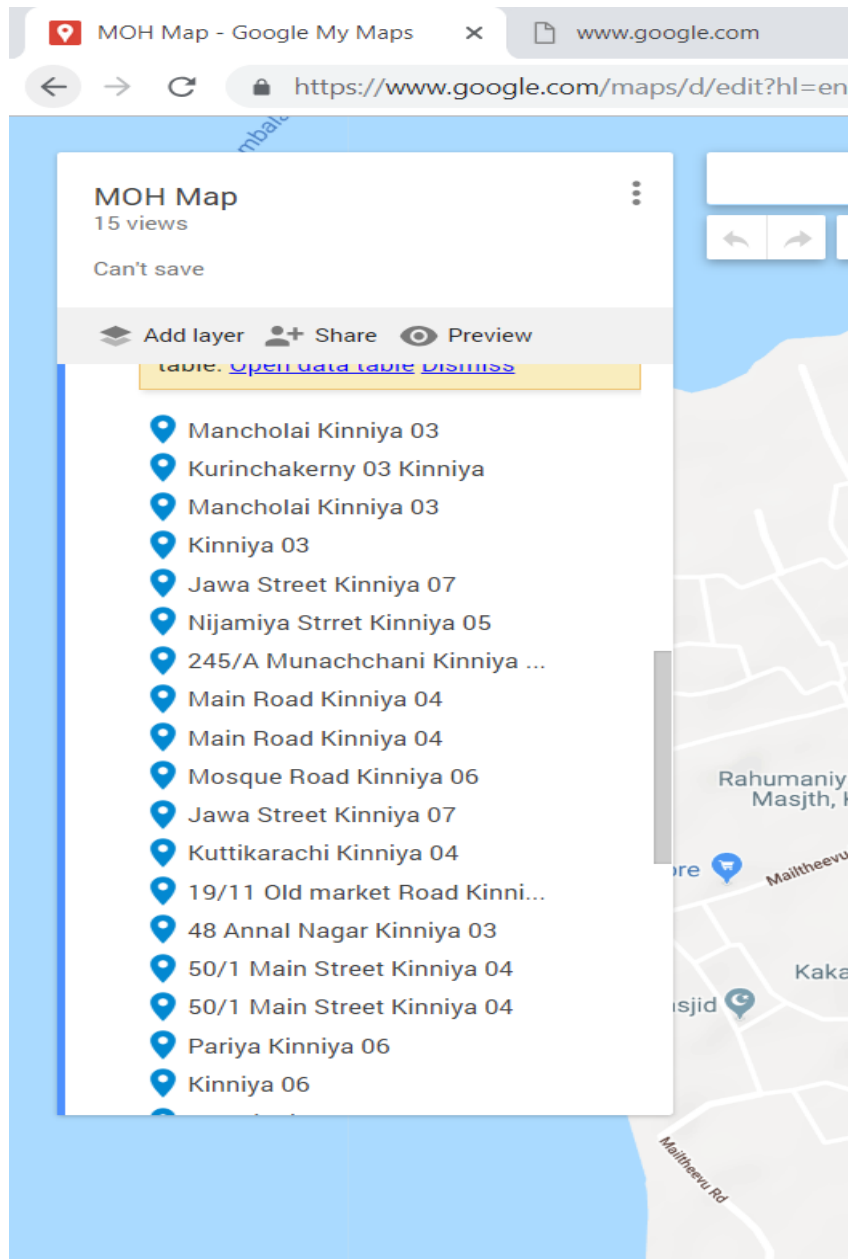
[BACK](#)
Enter New GS DIVISION
Name of GS Division

[Enter](#) [Update](#) [Remove](#)

Figure 26: Add New Doctors and GS Division

System Import data to google map

The dengue prevention online system will export excel format patient details and it will load to real google map developer tool.



Load google real Map to find affected places

By using google map developer and identify the coordination of the GS division and load into the project. Then the import tool will show an asterisk for relevant patients location also it shows the details of the patient.

figure 26 illustrates that the red hospital symbols of the real google map location of a patient affected by dengue.

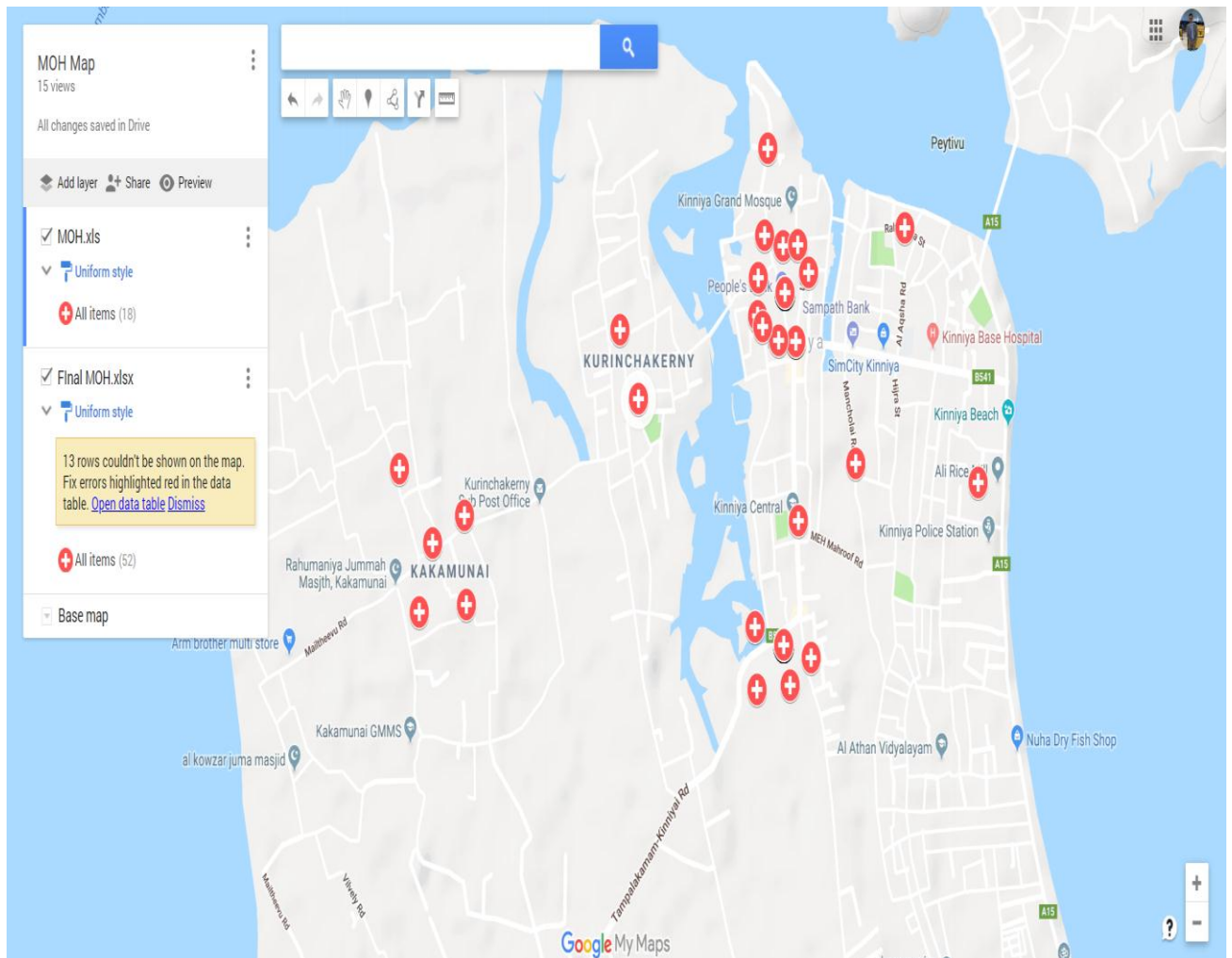


Figure 27: affected places

Reports and Charts

Through the following Link, the system User (MOH) can load the reports and charts for the decision making.

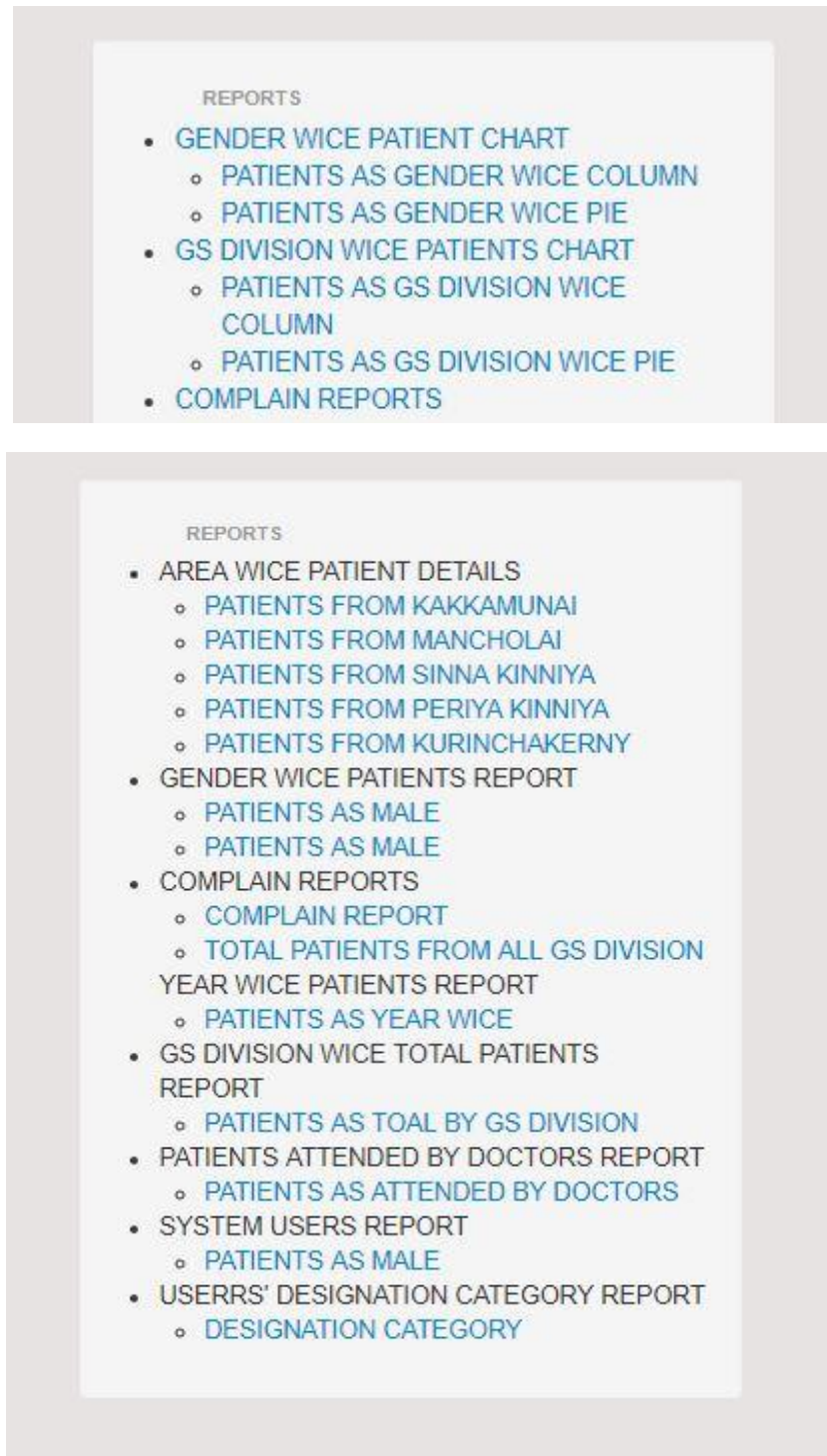


Figure 28: Reports and Charts

Complain to MOH

HOME MOH OUR STAFF GALLERY BLOG NEWS AFFECTED PLACE COMPLAINT LOGOUT mafahir@gmail.com

Complain Registration

Full Name

Address

GS Division

Telephone

E-mail

Comment

Figure 29: Complaint to MOH

Test Case: Logout

Expected: Output Complain form will be displayed

Status: Pass

Administrator

HOME MOH OUR STAFF GALLERY BLOG NEWS AFFECTED PLACE COMPLAINT LOGOUT mafahir@gmail.com

Patient

Show 10 entries Search:

ID	NIC	First Name	Last Name	Address	TP	DOB	Gender	GSdiv	Status	Next of Kin	Next of Kin Rel	Cons	Symptoms	Status of Comment	Action
19	853013811	Anif Lebbe	Mohamed Mafahir	Hijra Streek Kinniya 03	776622882	0000-00-00	Male	Sinna_Kinniya	Married	Faseer	Machan	Dr.Izza Hayam	Sever	ICU	<input type="button" value="EDIT"/>
20	893013811	Mohamed Ibrahim	Farook	Kinniya	771858074	1985-06-21	FeMale	Sinna_Kinniya	Married	Smaed	Father	Dr.Sawfan Sithara	Find	OPD	<input type="button" value="EDIT"/>
21	873013811	Saliheen	Sifath	Kinniya 04	778533665	2006-08-27	Male	Kurichakemy	Single	Ifam	Father	Dr. Ajith Perera	Fine	ECU	<input type="button" value="EDIT"/>

Showing 1 to 3 of 3 entries First Previous 1 Next Last

Figure 30: Administrator

Test Case: Take backups.

Expected: Output Backup must be taken and a success message must be displayed

Status: Pass

User privileges

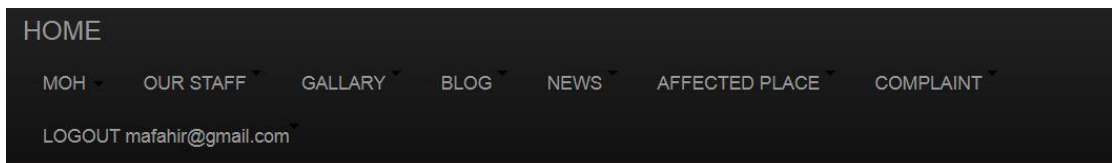


Figure 31: User privileges

Test Case : Logout.

Expected: Output User should be back to the front page

Status: Pass

Appendix C – Sample Code

```
<?php
require_once '../controller/config/config.php';
$template = new template();
if (isset($_SESSION['username'])) { // chk user logged status & user type
    } else {
        header('Location:' . BASE_URL . 'apps/view/userMgt/login.php'); // direct login
    }
?>
<!DOCTYPE html>
<!--[if IE 8]> <html class="no-js lt-ie9" lang="en"> <![endif]-->
<!--[if gt IE 8]><!--> <html class="no-js" lang="en"> <!--<![endif]-->
<?php $template->getHead(); ?>
<body>
    <?php $template->getMainMenu(); ?>
    <div class="vspace40">
        <div class="container marketing">
            <h2 align="center">Patient Registration</h2><hr/>
            <form class="form login-form" action="_patient.php" method="POST">
                PATIENT PERSONAL INFORMATION
                <div>
                    <div class="span8">
                        <div class="control-group">
                            <label class="control-label">Patient Number</label>
                            <div class="controls">
                                <input type="text" class="span6" name="patientRegNIC"
required="" autofocus="" maxlength="10" placeholder="Eg: 881234567V">
                            </div>
                        </div>
                        <div class="control-group">
                            <label class="control-label">Patient Registration Date</label>
                            <div class="controls">
                                <input type="text" class="span6" name="patientRegNIC"
required="" autofocus="" maxlength="10" placeholder="Eg: 881234567V">
                            </div>
                        </div>
                        <div class="control-group">
                            <label class="control-label">NIC</label>
                            <div class="controls">
                                <input type="text" class="span6" name="patientRegNIC"
required="" autofocus="" maxlength="10" placeholder="Eg: 881234567V">
                            </div>
                        </div>
                    </div>
                </div>
            </form>
        </div>
    </div>
</body>
</html>
```

```

        </div>
    </div>
    <div class="control-group">
        <label class="control-label">Title </label>
        <div class="controls">
            <select name="patientRegGender" >
                <option>Male</option>
                <option>FeMale</option>
            </select>
        </div>
    </div>
    <div class="control-group">
        <label class="control-label">First name</label>
        <div class="controls">
            <input type="text" class="span6"
name="patientRegFirstName" required="" placeholder="Perera">
        </div>
    </div>
    <div class="control-group">
        <label class="control-label">Last name</label>
        <div class="controls">
            <input type="text" class="span6"
name="patientRegLastName" required="" placeholder="Janith">
        </div>
    </div>
</div>
<div class="span4">

    <div class="control-group">
        <label class="control-label">Date of birth </label>
        <div class="controls">
            <input type="date" class="span4" name="patientRegDOB">
        </div>
    </div>
    <div class="control-group">
        <label class="control-label">Gender </label>
        <div class="controls">
            <select name="patientRegGender" class="span4">
                <option>Male</option>
                <option>FeMale</option>
            </select>
        </div>
    </div>
    <div class="control-group">

```

```

<label class="control-label">GS Division </label>
<div class="controls">
    <?php
        // This loads values to drop downbox from Database
        $ops = "";
        $stmt = $conn->query("select gsDivName from tbl_gsdiv order
by gsDivName");
        while ($row = $stmt->fetch(PDO::FETCH_ASSOC)) {
            $ops.= "<option value=\"" . $row['gsDivName'] . "\">" .
$row['gsDivName'] . "</option>";
        }
    ?>

    <select name="patientRegGSdiv" class="span4">
        <?php echo $ops; ?>
    </select>
</div>
</div>
<div class="control-group">
    <label class="control-label">Marital Status </label>
    <div class="controls">
        <select name="patientRegStatus" class="span4">
            <option>Married</option>
            <option>Single</option>
        </select>
    </div>
</div>
<div class="control-group">
    <label class="control-label">Occupation </label>
    <div class="controls">
        <select name="patientRegStatus" >
            <option>Married</option>
            <option>Single</option>
        </select>
    </div>
</div>
</div>
<div>
    <div class="span8">
        <?php $template->getMessage(); // display sucess and err
messages?>
    </div>
    <div class="control-group">
        <label class="control-label"> Address 1 </label>

```

```

        <div class="controls">
            <input type="text" class="span6" name="patientRegAddress"
placeholder="Ex Mancholai Kinniya">
        </div>
    </div>
    <div class="control-group">
        <label class="control-label">address 2 </label>
        <div class="controls">
            <input type="text" class="span6" name="patientRegAddress"
placeholder="Ex Mancholai Kinniya">
        </div>
    </div>
    <div class="control-group">
        <label class="control-label">City</label>
        <div class="controls">
            <input type="text" class="span6" name="patientRegAddress"
placeholder="Ex Mancholai Kinniya">
        </div>
    </div>
    <div class="control-group">
        <label class="control-label">District </label>
        <div class="controls">
            <input type="text" class="span6" name="patientRegAddress"
placeholder="Ex Mancholai Kinniya">
        </div>
    </div>
    </div>
    <div class="span4">
        <div class="control-group">
            <label class="control-label"> Name of Next of kin </label>
            <div class="controls">
                <input type="text" class="span4" name="patientRegKin"
placeholder="Ex Junaideen Faseer">
            </div>
        </div>
        <div class="control-group">
            <label class="control-label">Relationship to next of kin </label>
            <div class="controls">
                <input type="text" class="span4" name="patientRegKinRel"
placeholder="Ex Mother">
            </div>
        </div>
    </div>

```

```

        </div>
    </div>
    <div>
        <div class="span8">
            <div class="control-group">
                <label class="control-label">Consultant name </label>
                <div class="controls">
                    <?php
                        // This loads values to drop downbox from Database
                        $ops = "";
                        $stmt = $conn->query("select doctorName from tbl_doctor
order by doctorName");
                        while ($row = $stmt->fetch(PDO::FETCH_ASSOC)) {
                            $ops.= "<option value=\"" . $row['doctorName'] . "\">" .
$row['doctorName'] . "</option>";
                        }
                    ?>

                    <select name="patientRegCons" >
<?php echo $ops; ?>
                    </select>
                </div>
            </div>
            <div class="control-group">
                <label class="control-label">Description of symptoms </label>
                <div class="controls">
                    <input type="text" class="span6" name="patientRegSymp">
                </div>
            </div>
            <div class="control-group">
                <label class="control-label">Case Admission Status</label>
                <div class="controls">
                    <?php
                        // This loads values to drop downbox from Database
                        $ops = "";
                        $stmt = $conn->query("select caseAdStatus from
tbl_caseadmstatus order by caseAdStatus");
                        while ($row = $stmt->fetch(PDO::FETCH_ASSOC)) {
                            $ops.= "<option value=\"" . $row['caseAdStatus'] . "\">" .
$row['caseAdStatus'] . "</option>";
                        }
                    ?>
                </div>
            </div>
        </div>
    </div>

```

```

?>

<select name="patientStatus" >
<?php echo $ops; ?>
</select>
</div>
</div>
</div>
<div class="span4">
<div class="control-group">
<label class="control-label">Consultant name </label>
<div class="controls">
<?php
// This loads values to drop downbox from Database
$ops = "";
$stmt = $conn->query("select doctorName from tbl_doctor
order by doctorName");
while ($row = $stmt->fetch(PDO::FETCH_ASSOC)) {
    $ops.= "<option value=\"" . $row['doctorName'] . "\">" .
$row['doctorName'] . "</option>";
}
?>

<select name="patientRegCons" >
<?php echo $ops; ?>
</select>
</div>
</div>
<div class="control-group">
<label class="control-label">Description of symptoms </label>
<div class="controls">
<input type="text" class="span6" name="patientRegSymp">
</div>
</div>
<div class="control-group">
<label class="control-label">Case Admission Status</label>
<div class="controls">
<?php
// This loads values to drop downbox from Database
$ops = "";
$stmt = $conn->query("select caseAdStatus from
tbl_caseadmstatus order by caseAdStatus");
while ($row = $stmt->fetch(PDO::FETCH_ASSOC)) {

```

```

                $ops.= "<option value=\"" . $row['caseAdStatus'] . "\">" .
$row['caseAdStatus'] . "</option>";
            }
        ?>

        <select name="patientStatus" >
<?php echo $ops; ?>
        </select>
    </div>
</div>
    <div class="control-group">
        <div class="controls action">
            <button type="submit" class="btn btn-success"
name="patient_Register">Register</button>
            <button type="submit" class="btn btn-success"
name="patient_Update">Update</button>
            <button type="submit" class="btn btn-success"
name="patient_Remove">Remove</button>
            <button type="submit" class="btn btn-success"
hrer="patient_View">View</button>
            <button type="Reset" class="btn btn-success"
name="user_Clear">Clear</button>
        </div>
    </div>
    <a href="patient.php" class="fancybox fancybox.ajax btn btn-
success">VIEW PATIENT DETAILS</a>
    </form>
</div>
</div>
</div>
<?php $template->getFooter(); ?>
<?php $template->getUp(); ?>
    <script src="lib/template/js/holder.js"></script>
    <script src="lib/template/js/script.js"></script>
</body>
</html>

```


Map Lording

```
<?php

require_once '../controller/config/config.php';

$template = new template();

$sqlA = "SELECT patientRegGSdiv,count(patientRegID) patientRegID FROM
db_moh.tbl_patientregistration WHERE patientRegGSdiv='Kurichakerny'";

$stmtA = $conn->prepare($sqlA);

$stmtA->execute(array());

while ($row = $stmtA->fetch(PDO::FETCH_NUM)) {

    $patientRegGSdivA = $row[0];

    $patientRegIDA = $row[1];

}

$sqlB = "SELECT patientRegGSdiv,count(patientRegID) patientRegID FROM
db_moh.tbl_patientregistration WHERE patientRegGSdiv='Kaakamunai'";

$stmtB = $conn->prepare($sqlB);

$stmtB->execute(array());

while ($row = $stmtB->fetch(PDO::FETCH_NUM)) {

    $patientRegGSdivB = $row[0];

    $patientRegIDB = $row[1];

}

$sqlC = "SELECT patientRegGSdiv,count(patientRegID) patientRegID FROM
db_moh.tbl_patientregistration WHERE patientRegGSdiv='Sinna_Kinniya'";

$stmtC = $conn->prepare($sqlC);
```

```

$stmtC->execute(array());

while ($row = $stmtC->fetch(PDO::FETCH_NUM)) {

    $patientRegGSdivC = $row[0];

    $patientRegIDC = $row[1];

}

$sqlID = "SELECT patientRegGSdiv,count(patientRegID) patientRegID FROM
db_moh.tbl_patientregistration WHERE patientRegGSdiv='Periya_Kinniya'";

$stmtD = $conn->prepare($sqlID);

$stmtD->execute(array());

while ($row = $stmtD->fetch(PDO::FETCH_NUM)) {

    $patientRegGSdivD = $row[0];

    $patientRegIDD = $row[1];

}

$sqlE = "SELECT patientRegGSdiv,count(patientRegID) patientRegID FROM
db_moh.tbl_patientregistration WHERE patientRegGSdiv='Kuttikarachi'";

$stmtE = $conn->prepare($sqlE);

$stmtE->execute(array());

while ($row = $stmtE->fetch(PDO::FETCH_NUM)) {

    $patientRegGSdivE = $row[0];

    $patientRegIDE = $row[1];

}

?>

```

<!DOCTYPE html>

<!--[if IE 8]> <html class="no-js lt-ie9" lang="en"> <![endif]-->

```
<!--[if gt IE 8]><!--> <html class="no-js" lang="en"> <!--<![endif]-->
```

```
<?php $template->getHead(); ?>
```

```
<body>
```

```
<?php $template->getMainMenu(); ?>
```

```
<div class="vspace40">
```

```
<div class="container marketing">
```

```
<div class="row">
```

```
<h2 align="center">Map of Affected Places</h2><hr/>
```

```

```

```
<map name="Map" id="Map">
```

```
<area alt="Kurichakerny" title=<?php echo 'Division:' .
$patientRegGSdivA . ':Patients:' . $patientRegIDA; ?> href="#" shape="poly"
coords="318,13,335,6,349,6,362,10,378,15,387,24,388,29,385,34,374,39,362,40,356,
50,357,60,362,69,371,83,375,97,369,110,369,116,372,123,372,136,370,152,370,163,
377,177,379,185,372,192,366,184,359,169,351,160,342,144,335,135,328,125,319,11
1,312,100,305,87,305,65" />
```

```
<area alt="Kaakamunai" title=<?php echo 'Division:' .
$patientRegGSdivB . ':Count:' . $patientRegIDB; ?> href="#" shape="poly"
coords="192,186,197,213,200,238,204,254,254,308,294,302,338,285,365,258,387,23
6,409,216,418,196,403,178,403,171,376,161,238,146,202,148,188,148,187,149" />
```

```

<area alt="Sinna_Kinniya" title=<?php echo 'Division:' .
$patientRegGSdivC . ':Count:' . $patientRegIDC; ?> href="#" shape="poly"
coords="371,199,375,211,399,264,412,267,432,263,458,256,472,245,490,237,504,23
3,514,220,518,210,515,201,508,183,506,176,506,162,502,156,480,159,462,159,444,1
60,431,160,424,156,418,156,411,159,402,162" />

```

```

<area alt="Periya_Kinniya" title=<?php echo 'Division:' .
$patientRegGSdivD . ':Count:' . $patientRegIDD; ?> href="#" shape="poly"
coords="397,112,390,71,387,55,390,45,396,35,397,22,396,16,407,20,418,36,425,44,
436,43,446,45,452,46,467,51,477,51,486,55,495,60,500,66,502,77,502,87,502,106,50
2,120,501,129,502,137,502,147,502,154,487,156,451,160,436,157,426,155,417,154,4
09,154,402,157,399,151,398,144" />

```

```

<area alt="Kuttikarachi" title=<?php echo 'Division:' .
$patientRegGSdivE . ':Count:' . $patientRegIDE; ?> href="#" shape="poly"
coords="366,201,355,214,344,227,327,246,292,265,265,277,245,281,222,280,208,27
7,200,276,188,277,175,290,162,301,146,314,136,317,129,330,125,342,116,355,109,3
69,108,380,106,390,105,394,119,396,137,395,151,395,180,396,206,396,227,395,249,
394,272,391,300,380,312,372,331,361,338,351,350,339,357,329,362,315,374,301,38
0,293,384,281,386,270,390,260,390,254,384,243,377,231,375,219,371,210,368,204"
/>

```

```

</map>

```

```

</div>

```

```

<div class="span4">

```

```

<?php $template->getMessage(); // display sucess and err mesages ?>

```

```

</div>

```

```

</div>

```

```

</div>

```

```

</div>

```

```

<?php $template->getFooter(); ?>

```

```
<?php $template->getUp(); ?>
```

```
</body>
```

```
</html>
```

Report Area wise

```
<?php
ob_start();

require_once '../controller/config/config.php';

$template = new template();

?>

<page style="font-size: 10pt; background: url(../images/contback3.jpg) no-repeat
center; display: block; -moz-background-size: cover; background-size: cover; line-
height: 16px; background-color: #101010;">

<!---->

<!-- HEADER -->

<br/>

<div id="report_header" align="center"><span style="font-weight: bold; font-size:
20pt; color: #545355;">MOH Reports</span><br /></div><br />

<div id="report_title" align="center"><span style="font-weight: bold; font-size:
14pt; color: #D35D5D;">Patients from Kaakamunai</span><br /></div>

<hr/>

<!-- HEADER -->

<?php

$sql = "SELECT
patientRegNIC,patientRegAdd,patientRegFirstName,patientRegLastName,patientRe
```

```
gAddress,patientRegTP,patientRegDOB,patientRegGender,patientRegGSdiv FROM  
db_moh.tbl_patientregistration WHERE patientRegGSdiv='Kaakamunai';
```

```
$stmt = $conn->prepare($sql);
```

```
$stmt->execute(array());
```

```
echo '<table cellpadding="17" cellspacing="10" style=padding-left:25px ;">';
```

```
echo '<thead style="font-weight: bold; font-size: 12pt; color: grey;">';
```

```
echo '<tr>';
```

```
    echo '<th>NIC</th>';
```

```
    echo '<th>FirstName</th>';
```

```
    echo '<th>LAST NAME</th>';
```

```
    echo '<th>ADDRESS</th>';
```

```
    echo '<th>TP</th>';
```

```
    echo '<th>DOB</th>';
```

```
    echo '<th>GENDER</th>';
```

```
echo '</tr>';
```

```
echo '</thead>';
```

```
echo '<tbody>';
```

```
    while ($row = $stmt->fetch(PDO::FETCH_NUM))
```

```
    {
```

```
        echo '<tr>';
```

```
            echo '<td>'.$row[0].</td>';
```

```

        echo '<td>'.$row[2].</td>';

        echo '<td>'.$row[3].</td>';

        echo '<td>'.$row[4].</td>';

        echo '<td>'.$row[5].</td>';

        echo '<td>'.$row[6].</td>';

        echo '<td>'.$row[7].</td>';


        echo '</tr>';

    }

    echo '</tbody>';

    echo '</table>';

?>

<!-- FOOTER -->

<hr/>

<p>Created By Admin || Created On:: <?php echo date('m/d/Y h:i:s a', time());
?></p>

</page>

<?php

$content = ob_get_clean();

```



```
// convert in PDF

require_once('html2pdf/html2pdf.class.php');

try {

    $html2pdf = new HTML2PDF('P', 'A4', 'en', true, 'UTF-8', 3);

    $html2pdf->pdf->SetDisplayMode('fullpage');

    $html2pdf->writeHTML($content, isset($_GET['vuehtml']));

    $html2pdf->Output('user_report.pdf');

} catch (HTML2PDF_exception $e) {

    echo $e;

    exit;

}

?>
```

```
<?php

ob_start();

require_once '../controller/config/config.php';

$template = new template();

?>
```

```
<page style="font-size: 10pt; background: url(../images/contback3.jpg) no-repeat
center; display: block; -moz-background-size: cover; background-size: cover; line-
height: 16px; background-color: #101010;">

<!---->

<!-- HEADER -->

<br/>
```

```
<div id="report_header" align="center"><span style="font-weight: bold; font-size: 20pt; color: #545355;">MOH Reports</span><br /></div><br />
```

```
<div id="report_title" align="center"><span style="font-weight: bold; font-size: 14pt; color: #D35D5D;">Patients from Kurinchakerny</span><br /></div>
```

```
<hr/>
```

```
<!-- HEADER -->
```

```
<?php
```

```
$sql = "SELECT
patientRegNIC,patientRegAdd,patientRegFirstName,patientRegLastName,patientRegAddress,patientRegTP,patientRegDOB,patientRegGender,patientRegGSdiv FROM
db_moh.tbl_patientregistration WHERE patientRegGSdiv='kurichakerny'";
```

```
$stmt = $conn->prepare($sql);
```

```
$stmt->execute(array());
```

```
echo '<table cellpadding="17" cellspacing="10" style=padding-left:25px ;">';
```

```
echo '<thead style="font-weight: bold; font-size: 12pt; color: grey;">';
```

```
echo '<tr>';
```

```
    echo '<th>NIC</th>';
```

```
    echo '<th>FirstName</th>';
```

```
    echo '<th>LAST NAME</th>';
```

```
    echo '<th>ADDRESS</th>';
```

```
    echo '<th>TP</th>';
```

```
    echo '<th>DOB</th>';
```

```
echo '<th>GENDER</th>';
```

```
echo '</tr>';
```

```
echo '</thead>';
```

```
echo '<tbody>';
```

```
while ($row = $stmt->fetch(PDO::FETCH_NUM))
```

```
{
```

```
echo '<tr>';
```

```
    echo '<td>'.$row[0].</td>';
```

```
    echo '<td>'.$row[2].</td>';
```

```
    echo '<td>'.$row[3].</td>';
```

```
    echo '<td>'.$row[4].</td>';
```

```
    echo '<td>'.$row[5].</td>';
```

```
    echo '<td>'.$row[6].</td>';
```

```
    echo '<td>'.$row[7].</td>';
```

```
        echo '</tr>';
```

```
    }
```

```
echo '</tbody>';
```

```
echo '</table>';
```

```
?>
```

```
<!-- FOOTER -->
```

<hr/>

<p>Created By Admin || Created On:: <?php echo date('m/d/Y h:i:s a', time());
?></p>

</page>

<?php

\$content = ob_get_clean();

// convert in PDF

require_once('html2pdf/html2pdf.class.php');

try {

 \$html2pdf = new HTML2PDF('P', 'A4', 'en', true, 'UTF-8', 3);

 \$html2pdf->pdf->SetDisplayMode('fullpage');

 \$html2pdf->writeHTML(\$content, isset(\$_GET['vuehtml']));

 \$html2pdf->Output('user_report.pdf');

} catch (HTML2PDF_exception \$e) {

 echo \$e;

 exit;

}

?>

Report gender wise Male / Female

```
<?php
```

```
ob_start();
```

```
require_once '../controller/config/config.php';
```

```
$template = new template();
```

```
?>
```

```
<page style="font-size: 10pt; background: url(../images/contback3.jpg) no-repeat  
center; display: block; -moz-background-size: cover; background-size: cover; line-  
height: 16px; background-color: #101010;">
```

```
<!---->
```

```
<!-- HEADER -->
```

```
<br/>
```

```
<div id="report_header" align="center"><span style="font-weight: bold; font-size:  
20pt; color: #545355;">MOH Reports</span><br /></div><br />
```

```
<div id="report_title" align="center"><span style="font-weight: bold; font-size:  
14pt; color: #D35D5D;">Patients as FeMale</span><br /></div>
```

```
<hr/>
```

```
<!-- HEADER -->
```

```
<?php
```

```
$sql = "SELECT
patientRegNIC,patientRegAdd,patientRegFirstName,patientRegLastName,patientRe
gAddress,patientRegTP,patientRegDOB,patientRegGender,patientRegGSdiv FROM
db_moh.tbl_patientregistration WHERE patientRegGender='FeMale'";
```

```
$stmt = $conn->prepare($sql);
```

```
$stmt->execute(array());
```

```
echo '<table cellpadding="17" cellspacing="10" style=padding-left:25px ;">';
```

```
echo '<thead style="font-weight: bold; font-size: 12pt; color: grey;">';
```

```
echo '<tr>';
```

```
    echo '<th>NIC</th>';
```

```
    echo '<th>FirstName</th>';
```

```
    echo '<th>LAST NAME</th>';
```

```
    echo '<th>ADDRESS</th>';
```

```
    echo '<th>TP</th>';
```

```
    echo '<th>DOB</th>';
```

```
    echo '<th>GS DIVISION</th>';
```

```
echo '</tr>';
```

```
echo '</thead>';
```

```
echo '<tbody>';
```

```
    while ($row = $stmt->fetch(PDO::FETCH_NUM))
```

```
    {
```

```
        echo '<tr>';
```

```

        echo '<td>'.$row[0].</td>';

        echo '<td>'.$row[2].</td>';

        echo '<td>'.$row[3].</td>';

        echo '<td>'.$row[4].</td>';

        echo '<td>'.$row[5].</td>';

        echo '<td>'.$row[6].</td>';

        echo '<td>'.$row[8].</td>';


        echo '</tr>';

    }

    echo '</tbody>';

    echo '</table>';

?>

<!-- FOOTER -->

<hr/>

<p>Created By Admin || Created On:: <?php echo date('m/d/Y h:i:s a', time());
?></p>

</page>

<?php

$content = ob_get_clean();

```

```

// convert in PDF

require_once('html2pdf/html2pdf.class.php');

try {

    $html2pdf = new HTML2PDF('P', 'A3', 'en', true, 'UTF-8', 3);

    $html2pdf->pdf->SetDisplayMode('fullpage');

    $html2pdf->writeHTML($content, isset($_GET['vuehtml']));

    $html2pdf->Output('user_report.pdf');

} catch (HTML2PDF_exception $e) {

    echo $e;

    exit;

}

?>

<?php

ob_start();

require_once '../controller/config/config.php';

$template = new template();

?>

<page style="font-size: 10pt; background: url(../images/contback3.jpg) no-repeat
center; display: block; -moz-background-size: cover; background-size: cover; line-
height: 16px; background-color: #101010;">

<!---->

<!-- HEADER -->

<br/>

```



```
<div id="report_header" align="center"><span style="font-weight: bold; font-size: 20pt; color: #545355;">MOH Reports</span><br /></div><br />
```

```
<div id="report_title" align="center"><span style="font-weight: bold; font-size: 14pt; color: #D35D5D;">Patients as Male</span><br /></div>
```

```
<hr/>
```

```
<!-- HEADER -->
```

```
<?php
```

```
$sql = "SELECT
patientRegNIC,patientRegAdd,patientRegFirstName,patientRegLastName,patientReg
gAddress,patientRegTP,patientRegDOB,patientRegGender,patientRegGSdiv FROM
db_moh.tbl_patientregistration WHERE patientRegGender='Male'";
```

```
$stmt = $conn->prepare($sql);
```

```
$stmt->execute(array());
```

```
echo '<table cellpadding="17" cellspacing="10" style=padding-left:25px ;">';
```

```
echo '<thead style="font-weight: bold; font-size: 12pt; color: grey;">';
```

```
echo '<tr>';
```

```
    echo '<th>NIC</th>';
```

```
    echo '<th>FirstName</th>';
```

```
    echo '<th>LAST NAME</th>';
```

```
    echo '<th>ADDRESS</th>';
```

```
    echo '<th>TP</th>';
```

```
echo '<th>DOB</th>';  
echo '<th>GS DIVISION</th>';
```

```
echo '</tr>';  
echo '</thead>';
```

```
echo '<tbody>';  
while ($row = $stmt->fetch(PDO::FETCH_NUM))  
{  
    echo '<tr>';  
    echo '<td>'.$row[0].</td>';  
    echo '<td>'.$row[1].</td>';  
    echo '<td>'.$row[2].</td>';  
    echo '<td>'.$row[3].</td>';  
    echo '<td>'.$row[4].</td>';  
    echo '<td>'.$row[5].</td>';  
    echo '<td>'.$row[6].</td>';  
    echo '<td>'.$row[8].</td>';  
  
    echo '</tr>';  
}  
echo '</tbody>';  
echo '</table>';
```

```
?>
```

```
<!-- FOOTER -->
```

<hr/>

<p>Created By Admin || Created On:: <?php echo date('m/d/Y h:i:s a', time());
?></p>

</page>

<?php

\$content = ob_get_clean();

// convert in PDF

require_once('html2pdf/html2pdf.class.php');

try {

 \$html2pdf = new HTML2PDF('P', 'A3', 'en', true, 'UTF-8', 3);

 \$html2pdf->pdf->SetDisplayMode('fullpage');

 \$html2pdf->writeHTML(\$content, isset(\$_GET['vuehtml']));

 \$html2pdf->Output('user_report.pdf');

} catch (HTML2PDF_exception \$e) {

 echo \$e;

 exit;

}

?>

Report Year / Monthly Wise

```
<?php
```

```
ob_start();
```

```
require_once '../controller/config/config.php';
```

```
$template = new template();
```

```
?>
```

```
<page style="font-size: 10pt; background: url(../images/contback3.jpg) no-repeat  
center; display: block; -moz-background-size: cover; background-size: cover; line-  
height: 16px; background-color: #101010;">
```

```
<!---->
```

```
<!-- HEADER -->
```

```
<br/>
```

```
<div id="report_header" align="center"><span style="font-weight: bold; font-size:  
20pt; color: #545355;">MOH Reports</span><br /></div><br />
```

```
<div id="report_title" align="center"><span style="font-weight: bold; font-size:  
14pt; color: #D35D5D;">Patients Admitted by Year</span><br /></div>
```

```
<hr/>
```

```
<!-- HEADER -->
```

```
<?php
```

```
    $sql = "SELECT year(patientRegAdd) , count(*) FROM  
db_moh.tbl_patientregistration group by year(patientRegAdd)";
```

```
    $stmt = $conn->prepare($sql);
```

```
    $stmt->execute(array());
```

```
    echo '<table cellpadding="17" cellspacing="10" style=padding-left:25px ;">';
```

```
    echo '<thead style="font-weight: bold; font-size: 12pt; color: grey;">';
```

```
    echo '<tr>';
```

```
        echo '<th>patientRegAdd</th>';
```

```
        echo '<th>Total</th>';
```

```
    echo '</tr>';
```

```
    echo '</thead>';
```

```
    echo '<tbody>';
```

```
        while ($row = $stmt->fetch(PDO::FETCH_NUM))
```

```
        {
```

```
            echo '<tr>';
```

```
                echo '<td>'.$row[0].</td>';
```

```
                echo '<td>'.$row[1].</td>';
```

```

        echo '</tr>';

    }

    echo '</tbody>';

    echo '</table>';

?>

<!-- FOOTER -->

<hr/>

<p>Created By Admin || Created On:: <?php echo date('m/d/Y h:i:s a', time());
?></p>

</page>

<?php

$content = ob_get_clean();

// convert in PDF

require_once('html2pdf/html2pdf.class.php');

try {

    $html2pdf = new HTML2PDF('P', 'A4', 'en', true, 'UTF-8', 3);

    $html2pdf->pdf->SetDisplayMode('fullpage');

    $html2pdf->writeHTML($content, isset($_GET['vuehtml']));

    $html2pdf->Output('user_report.pdf');

```

```
} catch (HTML2PDF_exception $e) {  
    echo $e;  
    exit;  
}  
?>
```

Chart reports analysis gender , Area

```
<?php
require_once '../controller/config/config.php';
$template = new template();
?>
```

```
<!DOCTYPE html>
<!--[if IE 8]> <html class="no-js lt-ie9" lang="en"> <![endif]-->
<!--[if gt IE 8]><!--> <html class="no-js" lang="en"> <!--<![endif]-->
<?php $template->getHead(); ?>
<body>
```

```
<?php $template->getMainMenu(); ?>
```

```
<div class="vspace30">
  <div class="container">
    <div class="row">
      <div class="span4">

        <div class="well">
          <ul class="dropdown">
            <li class="nav-header">REPORTS</li>
            <li class="dropdown-toggle" data-toggle="dropdown">
              <a href="#">GENDER WICE PATIENT CHART</a>
```


- [PATIENTS AS GENDER
WICE COLUMN](../gender.php)
 - [PATIENTS AS GENDER
WICE PIE](../genderpie.php)

- [GS DIVISION WICE PATIENTS CHART](#)
 - [PATIENTS AS GS DIVISION
WICE COLUMN](../gsdiv.php)
 - [PATIENTS AS GS
DIVISION WICE PIE](../gsdivpie.php)

- [COMPLAIN
REPORTS](../complain/complain.php)
- [Link](#)

- [Link](#)

Link

<li class="nav-header">Sidebar

Link

Link

Link

</div>

</div>

</div>

```
</div>
```

```
</div>
```

```
<?php $template->getFooter(); ?>
```

```
</body>
```

```
</html>
```

```
<?php
```

```
require_once '../controller/config/config.php';
```

```
$template = new template();
```

```
?>
```

```
<!DOCTYPE html>
```

```
<!--[if IE 8]> <html class="no-js lt-ie9" lang="en"> <![endif]-->
```

```
<!--[if gt IE 8]><!--> <html class="no-js" lang="en"> <!--<![endif]-->
```

```
<?php $template->getHead(); ?>
```

```
<body>
```

```
<?php $template->getMainMenu(); ?>
```

```
<div class="vspace40">
```

```
<div class="container marketing">
```

```
<div class="row">
```

```
<div class="span8">
```

```
<h2 align="center">No of Patient on Gender</h2><hr/>
```

```
<?php
```

```
$sql = 'SELECT count(patientRegID) patientRegID, patientRegGender  
FROM tbl_patientregistration group by patientRegGender';
```

```
$stmt = $conn->prepare($sql);
```

```
$stmt->execute(array());
```

```
while ($row = $stmt->fetch(PDO::FETCH_NUM)) {
```

```
$data1[] = array('patientRegID' => $row[0], 'patientRegGender' =>  
$row[1]);
```

```
}
```

```
?>
```

```
<div id="chartdiv" style="width:500px; height:400px;"></div>
```

```
</div>
```

```
<div class="span4">
```

```
<?php $template->getMessage(); // display sucess and err mesages ?>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
<?php $template->getFooter(); ?>
```

```
<?php $template->getUp(); ?>
```

```
<!-- <script src="lib/template/js/holder.js"></script>
```

```
<script src="lib/template/js/script.js"></script>-->
```

```
</body>
```

```
<script type="text/javascript">
```

```
var chart;
```

```
var chartData = <?php echo json_encode($data1); ?>;
```

```
AmCharts.ready(function () {
```

```
    // SERIAL CHART
```

```
    chart = new AmCharts.AmSerialChart();
```

```
    chart.pathToImages = "../amcharts/images/";
```

```
    chart.dataProvider = chartData;
```

```
    chart.categoryField = "patientRegGender";
```

```
    chart.startDuration = 1;
```

```

chart.handDrawn = true;

chart.handDrawnScatter = 3;


// AXES

// category
var categoryAxis = chart.categoryAxis;
categoryAxis.gridPosition = "start";


// value
var valueAxis = new AmCharts.ValueAxis();
valueAxis.axisAlpha = 0;
chart.addValueAxis(valueAxis);


// GRAPHS

// column graph
var graph1 = new AmCharts.AmGraph();
graph1.type = "column";
graph1.title = "No of users";
graph1.lineColor = "#050505";
graph1.valueField = "patientRegID";
graph1.lineAlpha = 1;
graph1.fillAlphas = 1;
graph1.dashLengthField = "dashLengthColumn";
graph1.alphaField = "alpha";

graph1.balloonText = "<span style='font-size:13px;'>[[title]] in  
[[category]]:<b>[[value]]</b> [[additional]]</span>";

chart.addGraph(graph1);

```

```

// line

var graph2 = new AmCharts.AmGraph();

graph2.type = "line";

graph2.title = "User Types";

graph2.lineColor = "#fcd202";

graph2.valueField = "patientRegGender";

graph2.lineThickness = 3;

graph2.bullet = "round";

graph2.bulletBorderThickness = 3;

graph2.bulletBorderColor = "#fcd202";

graph2.bulletBorderAlpha = 1;

graph2.bulletColor = "#ffffff";

graph2.dashLengthField = "dashLengthLine";

graph2.balloonText = "<span style='font-size:13px;'>[[title]] in
[[category]]:<b>[[value]]</b> [[additional]]</span>";

chart.addGraph(graph2);


// LEGEND

var legend = new AmCharts.AmLegend();

legend.useGraphSettings = true;

chart.addLegend(legend);


// WRITE

chart.write("chartdiv");

});

</script>

```

</html>

<?php

require_once '../controller/config/config.php';

\$template = new template();

?>

<!DOCTYPE html>

<!--[if IE 8]> <html class="no-js lt-ie9" lang="en"> <![endif]-->

<!--[if gt IE 8]><!--> <html class="no-js" lang="en"> <!--<![endif]-->

<?php \$template->getHead(); ?>

<body>

<?php \$template->getMainMenu(); ?>

<div class="vspace40">

<div class="container marketing">

<div class="row">

<div class="span8">

<h2 align="center">No of Patient by GS Division</h2><hr/>


```

        <?php
            $sql = 'SELECT count(patientRegID) patientRegID, patientRegGSdiv
FROM tbl_patientregistration group by patientRegGSdiv';

            $stmt = $conn->prepare($sql);

            $stmt->execute(array());

            while ($row = $stmt->fetch(PDO::FETCH_NUM)) {

                $data1[] = array('patientRegID' => $row[0], 'patientRegGSdiv' =>
$row[1]);
            }
        ?>

        <div id="chartdiv" style="width:1000px; height:400px;"></div>

    </div>

    <div class="span4">

        <?php $template->getMessage(); // display sucess and err mesages ?>

    </div>

</div>

</div>

</div>

```

```
<?php $template->getFooter(); ?>
```

```
<?php $template->getUp(); ?>
```

```
<!--      <script src="lib/template/js/holder.js"></script>
<script src="lib/template/js/script.js"></script>-->
```

```
</body>
```

```
<script type="text/javascript">
```

```
var chart;
```

```
var chartData = <?php echo json_encode($data1); ?>;
```

```
AmCharts.ready(function () {
```

```
    // PIE CHART
```

```
    chart = new AmCharts.AmPieChart();
```

```
    // title of the chart
```

```
    chart.addTitle("Patient on GS Division", 16);
```

```
    chart.dataProvider = chartData;
```

```
    chart.titleField = "patientRegGSdiv";
```

```
    chart.valueField = "patientRegID";
```

```
    chart.sequencedAnimation = true;
```

```
    chart.startEffect = "elastic";
```

```

        chart.innerRadius = "30%";

        chart.startDuration = 2;

        chart.labelRadius = 15;

        chart.balloonText = "[[title]]<br><span style='font-size:14px'><b>[[value]]</b> ([[percents]]%)</span>";

        // the following two lines makes the chart 3D

        chart.depth3D = 10;

        chart.angle = 15;


        // WRITE

        chart.write("chartdiv");

    });
</script>


</html>


<?php
require_once '../controller/config/config.php';

$template = new template();

?>


<!DOCTYPE html>

<!--[if IE 8]> <html class="no-js lt-ie9" lang="en"> <![endif]-->

<!--[if gt IE 8]><!--> <html class="no-js" lang="en"> <!--<![endif]-->

<?php $template->getHead(); ?>

```

```

<body>

<?php $template->getMainMenu(); ?>

<div class="vspace40">

    <div class="container marketing">

        <div class="row">

            <div class="span8">

                <h2 align="center">View Bar Charts</h2><hr/>

                <?php

                    $sql = 'SELECT count(userLoginID) userLoginID, userLoginType
FROM tbl_userlogin group by userLoginType';

                    $stmt = $conn->prepare($sql);

                    $stmt->execute(array());

                    while ($row = $stmt->fetch(PDO::FETCH_NUM)) {

                        $data1[] = array('userLoginID' => $row[0], 'userLoginType' =>
$row[1]);

                    }

                ?>

```

```
<div id="chartdiv" style="width:1000px; height:400px;"></div>
```

```
</div>
```

```
<div class="span4">
```

```
<?php $template->getMessage(); // display sucess and err mesages ?>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
<?php $template->getFooter(); ?>
```

```
<?php $template->getUp(); ?>
```

```
<!-- <script src="lib/template/js/holder.js"></script>
```

```
<script src="lib/template/js/script.js"></script>-->
```

```
</body>
```

```
<script type="text/javascript">
```

```
var chart;
```

```
var chartData = <?php echo json_encode($data1); ?>;
```

```
AmCharts.ready(function () {  
    // SERIAL CHART  
  
    chart = new AmCharts.AmSerialChart();  
    chart.pathToImages = "../amcharts/images/";  
    chart.dataProvider = chartData;  
    chart.categoryField = "userLoginType";  
    chart.startDuration = 1;  
  
    chart.handDrawn = true;  
    chart.handDrawnScatter = 3;  
  
    // AXES  
  
    // category  
    var categoryAxis = chart.categoryAxis;  
    categoryAxis.gridPosition = "start";  
  
    // value  
    var valueAxis = new AmCharts.ValueAxis();  
    valueAxis.axisAlpha = 0;  
    chart.addValueAxis(valueAxis);  
  
    // GRAPHS  
  
    // column graph  
    var graph1 = new AmCharts.AmGraph();  
    graph1.type = "column";
```

```

graph1.title = "No of users";
graph1.lineColor = "#050505";
graph1.valueField = "userLoginID";
graph1.lineAlpha = 1;
graph1.fillAlphas = 1;
graph1.dashLengthField = "dashLengthColumn";
graph1.alphaField = "alpha";
graph1.balloonText = "<span style='font-size:13px;'>[[title]] in
[[category]]:<b>[[value]]</b> [[additional]]</span>";
chart.addGraph(graph1);

// line
var graph2 = new AmCharts.AmGraph();
graph2.type = "line";
graph2.title = "User Types";
graph2.lineColor = "#fcd202";
graph2.valueField = "userLoginType";
graph2.lineThickness = 3;
graph2.bullet = "round";
graph2.bulletBorderThickness = 3;
graph2.bulletBorderColor = "#fcd202";
graph2.bulletBorderAlpha = 1;
graph2.bulletColor = "#ffffff";
graph2.dashLengthField = "dashLengthLine";
graph2.balloonText = "<span style='font-size:13px;'>[[title]] in
[[category]]:<b>[[value]]</b> [[additional]]</span>";
chart.addGraph(graph2);

```

```
// LEGEND

var legend = new AmCharts.AmLegend();

legend.useGraphSettings = true;

chart.addLegend(legend);


// WRITE

chart.write("chartdiv");

});

</script>


</html>
```