



Field Planning and Monitoring System for World Mosquito Program, Sri Lankan Site

**A dissertation submitted for the Degree of Master
of Information Technology**

**H.G.G. Kith
University of Colombo School of Computing
2021**



Declaration

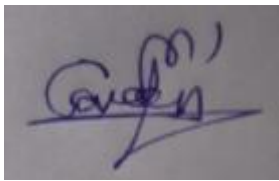
The thesis is my original work and has not been submitted previously for a degree at this or any other university/institute.

To the best of my knowledge it does not contain any material published or written by another person, except as acknowledged in the text.

Student Name: H.G.G. Kith

Registration Number: 2018/MIT/043

Index No: 18550433



.....

Signature

Date: 26/11/2021

This is to certify that this thesis is based on the work of
Mr. H.G.G. Kith

Under my supervision. The thesis has been prepared according to the format stipulated and is of acceptable standard.

Certified By:

Supervisor Name: MGNAS Fernando



.....

Signature

Date: 29/11/2021

Abstract

The World Mosquito Program (WMP) is a not-for-profit initiative that exists to protect the global community from mosquito-borne diseases. WMP uses safe and natural bacteria called “Wolbachia” to prevent the transmission of mosquito-borne diseases such as Dengue, Zika, Chikungunya and Yellow Fever. Currently, WMP operates one of its sites in Sri Lanka by covering two pilot Medical Officer of Health (MOH) areas such as Colombo and Nugegoda. The present field planning and field monitoring operations are handled by the use of partially computerized system which has given rise to many problems and is very time consuming and proven to be very ineffective. As a solution, WMP has decided to converge on an information technology solution in order to develop a comprehensive computerized system which facilitates various needs and expectations of the diverse stakeholders of the system. The proposed field planning and monitoring system is expected to cover almost all the operations and processes related to field planning and monitoring of the organization including spatial data management, field planning management, field activities management, field incidents management, mosquito diagnostic management, screening results management, bio-banking exports management and reports generation.

The proposed field planning and monitoring system consists of mobile application and a web based application. The proposed system is developed in accordance to object oriented techniques employing various UML diagrams from modelling the system and adhering to the MVC architecture. Meanwhile, the system is developed according to RUP framework. Android Studio is used as the integrated development environment to develop the proposed mobile application by targeting Android platform. Also, Java is used as the programming language inside Android Studio for the development of proposed mobile application. XAMPP web server solution stack is used to develop the proposed system utilizing Apache as the web server. Codeigniter is used as the server-side framework and JavaScript is used as the client-side programming language to build the web application while MySQL is used for handling the centralized database. Data synchronization between mobile application and remote MySQL database is done using web based application program interfaces. Geographic Information Systems technologies have been applied in the system using software development kits and application program interfaces. Google Maps SDK for Android has been used in mobile application and LeafletJs API has been used in web application to apply Geographic Information Systems Technologies in the system.

Ultimately, the developed system is thoroughly tested using a comprehensive testing procedure and evaluated to determine whether the system and user requirements are accomplished. Therefore, it can be understood that the developed system confirms to its specification whilst providing the intended functionality and anticipated outcome.

Acknowledgement

First, I would like to thank my parents, sister and wife for giving me great helps to make this project success. Also, I would like to thank Dr. MGNAS Fernando who is the supervisor of this project, for his excellent guidance in every phase of this project. Also, I would like to thank managerial staff of World Mosquito Program for providing me the domain knowledge about their existing system, weaknesses of the existing system, and expectations from a new system. Also, I would like to thank my friends for sharing their ideas and knowledge with me.

Table of Contents

Declaration.....	ii
Abstract.....	iii
Acknowledgement.....	v
Table of Contents.....	vi
List of Figures.....	ix
List of Tables.....	xiii
List of Acronyms.....	xv
1.0 Introduction.....	01
1.1 Project Overview.....	01
1.2 Motivation	02
1.3 Objectives.....	03
1.4 Scope of the Study.....	03
1.5 Structure of the Dissertation.....	04
2.0 Background.....	05
2.1 Introduction.....	05
2.2 Requirement Analysis	05
2.2.1 Analysis of Existing System.....	05
2.2.2 Functional Requirements	06
2.2.3 Non-Functional Requirements.....	07
2.3 Review of Similar Systems	07
2.4 Comparison Between Similar Systems and Required Features.....	13
2.5 Selected Process Model.....	15
2.6 Summary.....	16
3.0 Design	17
3.1 Design Solution	17
3.2 System Architecture.....	18

3.3 Design Approach.....	21
3.4 High-Level Use-Case Diagram of Proposed System.....	22
3.5 Use-case Diagrams and Use-Case Narratives for Module Design	22
3.6 Class Diagram.....	32
3.7 Sequence Diagrams.....	33
3.8 Database Design.....	38
3.8.1 Entity-Relationship Diagram.....	38
3.8.2 Mapping Entity-Relationship Diagram to Relational Schemas.....	38
3.9 User Interface Designs.....	40
3.10 MIS Reports.....	46
3.11 Summary.....	48
4.0 Implementation	49
4.1 Introduction.....	49
4.2 Implementation Environment.....	49
4.3 Development Tools.....	49
4.4 Technologies.....	50
4.5 Reused Components.....	51
4.6 Major Code Segments.....	51
5.0 Testing and Evaluation.....	59
5.1 Introduction.....	59
5.2 Testing Overview.....	59
5.3 Test Cases and Test Results.....	59
5.4 User Evaluation.....	81
5.5 Summary.....	83
6.0 Conclusion	84
6.1 Introduction.....	84

6.2 Objectives Achieved.....	84
6.3 Problems Encountered.....	85
6.4 Lessons Learnt.....	86
6.5 Future Improvements.....	86
References	87
Appendix A – System Manual.....	89
Appendix B – Design Documentation.....	91
Appendix C – User Manual.....	98
Appendix D – MIS Reports.....	101
Appendix E – Test Cases and Test Results.....	104

List of Figures

Figure 2.1 A user interface of Google MyMaps.....	8
Figure 2.2 User interface of map view of Google MyMaps mobile application.....	9
Figure 2.3 User interface of list view of Google MyMaps mobile application.....	10
Figure 2.4 A user interface of KoBo Toolbox web application.....	12
Figure 2.5 A user interface of KoBo Collect mobile application.....	12
Figure 3.1 MVC architectural design pattern.....	18
Figure 3.2 System architecture of proposed mobile application.....	19
Figure 3.3 System architecture of proposed web application	20
Figure 3.4 System module structure of the proposed web application.....	20
Figure 3.5 High level use case diagram of proposed system.....	22
Figure 3.6 Use case diagram of spatial data management module.....	23
Figure 3.7 Use-case diagram of field planning management module.....	25
Figure 3.8 Use-case diagram of field activities management module.....	27
Figure 3.9 Use-case diagram of field incident management module	31
Figure 3.10 Class diagram.....	32
Figure 3.11 Sequence diagram of add new map layer.....	33
Figure 3.12 Sequence diagram of add new field run	34
Figure 3.13 Sequence diagram of add new field incident.....	35
Figure 3.14 Sequence diagram of add identification result.....	36
Figure 3.15 Sequence diagram of add new screening result.....	37
Figure 3.16 ER diagram of proposed system.....	38
Figure 3.17 Main menu user interface of proposed mobile application.....	42
Figure 3.18 OVI field activities interface of proposed mobile application.....	43
Figure 3.19 Map view of proposed mobile application.....	44
Figure 3.20 User interface of add new mosquito diagnostic of web application.....	45
Figure 3.21 User interface of add new map layer screen of web application	46
Figure 3.22 OVI performance report.....	47
Figure 3.23 Mosquito screening report.....	48
Figure 4.1 Code segment for database connection establishment in mobile application...	52

Figure 4.2 Code segment for table creation in mobile application.....	52
Figure 4.3 Code segment for data insertion to tables in mobile application.....	53
Figure 4.4 Code segment for querying data from tables in mobile application.....	53
Figure 4.5 Code segment for updating a table in mobile application	53
Figure 4.6 Attributes of model class of mobile application.....	54
Figure 4.7 Constructors of model class of mobile application.....	54
Figure 4.8 Getter and setter methods of model class in mobile application.....	55
Figure 4.9 Controller class of mobile application.....	56
Figure 4.10 Maps_model model class in web application.....	57
Figure 4.11 goOviPerformanceReportView() function of ReportController class.....	57
Figure 5.1 Error message for empty required fields.....	60
Figure 5.2 Error message for already existing MRC identifier.....	61
Figure 5.3 Success message after adding new MRC location.....	61
Figure 5.4 Error message for invalid email address.....	62
Figure 5.5 Error message for invalid phone no.....	62
Figure 5.6 Success message after adding new field incident	62
Figure 5.7 Error message for existing service id.....	63
Figure 5.8 Error message for empty service status.....	63
Figure 5.9 Error message after adding MRC service	64
Figure 5.10 Error message for existing release id.....	65
Figure 5.11 Error message for empty release status.....	65
Figure 5.12 Success message after adding MRC release.....	65
Figure 5.13 Error message for existing BG collection id.....	66
Figure 5.14 Error message for empty collection status.....	66
Figure 5.15 Success message after adding BG collection.....	66
Figure 5.16 Map view of MRC locations.....	68
Figure 5.17 Error message for empty required fields	69
Figure 5.18 Error message for already existing map name.....	70
Figure 5.19 Success message after adding new map layer to system	70
Figure 5.20 Error message for uploading a file of invalid file type.....	71

Figure 5.21 Error message for existing run name.....	71
Figure 5.22 Error message for invalid trap ids which are not existing in system.....	72
Figure 5.23 Error message for trap ids which have invalid status.....	72
Figure 5.24 Error message for trap ids which are already in a pending field run.....	72
Figure 5.25 Success message after adding OVI service field run.....	72
Figure 5.26 Error message for already existing diagnostic id.....	73
Figure 5.27 Error message for invalid identified date.....	73
Figure 5.28 Success message after adding identification details.....	74
Figure 5.29 Error message for existing screening id.	75
Figure 5.30 Error message for invalid screened date	75
Figure 5.31 Success message after adding screening result.....	75
Figure 5.32 Error message for already existing export id.....	76
Figure 5.33 Error message for invalid export date.....	76
Figure 5.34 Success message after adding bio-bank export.....	76
Figure 5.35 Error message for invalid date range.....	77
Figure 5.36 Error message for not having enough data for selected date range.....	77
Figure 5.37 Generated OVI performance report.....	78
Figure 5.38 Location visualization in GIS dashboard.....	79
Figure B.1 Use-case diagram of mosquito diagnostic management module.....	91
Figure B.2 Use-case diagram of screening results management module.....	93
Figure B.3 Use-case diagram of bio-banking export management module.....	95
Figure B.4 Use-case diagram of report generation module.....	96
Figure C.1 Login screen of mobile application.....	98
Figure C.2 Menu screen of mobile application.....	99
Figure C.3 Main menu screen of web application.....	100
Figure D.1 Mosquito diagnostic report.....	101
Figure D.2 BG performance report.....	102
Figure D.3 MRC performance report	103
Figure E.1 Error message for empty required fields.....	105
Figure E.2 Error message for existing BG trap id.....	105

Figure E.3 Success message after adding BG trap location.....	105
Figure E.4 Error message for empty required fields.....	106
Figure E.5 Error message for already existing OVI trap id.....	107
Figure E.6 Success message after adding OVI trap location.....	107
Figure E.7 Error message for existing service id.....	108
Figure E.8 Error message for empty service status.....	108
Figure E.9 Success message after adding BG service.....	108
Figure E.10 Error message for existing service id.....	109
Figure E.11 Error message for empty service status.....	109
Figure E.12 Success message after adding OVI service.....	109
Figure E.13 Error message for existing OVI collection id.....	110
Figure E.14 Error message for empty collection status.....	110
Figure E.15 Error message after adding OVI collection.....	111
Figure E.16 Map visualization of the current location.....	112
Figure E.17 Confirm box for deleting map layer.....	114
Figure E.18 Success message box after deleting map layer.....	115

List of Tables

Table 2.1 Comparison between similar systems and proposed solution.....	14
Table 3.1 Use-case narrative of add new map layer.....	23
Table 3.2 Use-case narrative of add new field run.....	25
Table 3.3 Use-case narrative of add new field location.....	27
Table 3.4 Use-case narrative of add new collection/release.....	28
Table 3.5 Use-case narrative of do data synchronization.....	29
Table 3.6 Use-case narrative of add new field incident.....	31
Table 4.1 Implementation environment.....	49
Table 4.2 Development tools	49
Table 4.3 Technologies.....	50
Table 4.4 Reused components.....	51
Table 5.1 Test case for add a new MRC location in mobile application.....	59
Table 5.2 Test case for add a new field incident in mobile application.....	61
Table 5.3 Test case for add a new MRC service in mobile application.....	62
Table 5.4 Test case for add new MRC release in mobile application.....	64
Table 5.5 Test case for add a new BG collection in mobile application.....	65
Table 5.6 Test case for view MRC field locations in map view.....	67
Table 5.7 Test case for add new map layer in web application.....	68
Table 5.8 Test case for add new OVI service run in web application.....	70
Table 5.9 Test case for add new mosquito diagnostic in web application.....	73
Table 5.10 Test case for add new mosquito screening result in web application.....	74
Table 5.11 Test case for add new bio-banking export in web application.....	75
Table 5.12 Test case for generate OVI performance report in web application.....	76
Table 5.13 Test case for GIS dashboard in web application.....	78
Table 5.14 Test case for save encrypted passwords.....	80
Table 5.15 Test cases for usability testing.....	80
Table A.1 Hardware requirements for mobile application.....	89
Table A.2 Software requirements for mobile application.....	89
Table A.3 Hardware requirements for web application.....	90

Table A.4 Software requirements for web application.....	90
Table B.1 Use-case narrative of add new identification result.....	91
Table B.2 Use-case narrative of delete identification result.....	92
Table B.3 Use-case narrative of add new screening result.....	94
Table B.4 Use-case narrative of add new bio banking export.....	95
Table B.5 Use-case narrative of generate performance report of BG traps.....	96
Table E.1 Test case for add a new BG location in mobile application.....	104
Table E.2 Test case for add new OVI location in mobile application.....	105
Table E.3 Test case for add a new BG service in mobile application.....	107
Table E.4 Test case for add a new OVI service in mobile application.....	108
Table E.5 Test case for add a new OVI collection in mobile application.....	109
Table E.6 Test case for view current location of mobile device in map view.....	111
Table E.7 Test case for update new OVI location in mobile application.....	112
Table E.8 Test case for update BG collection in mobile application	113
Table E.9 Test case for view map layers in map view in web application.....	113
Table E.10 Test case for delete a map layer in web application.....	114
Table E.11 Test case for update a map layer in web application.....	115

List of Acronyms

API	– Application Program Interface
BG	– Biogents Sentinel
ER	– Entity Relationship
GIS	– Geographic Information Systems
GPS	– Geographic Positioning System
KML	– Keyhole Markup Language
MIS	- Management Information Systems
MOH	– Medical Officer of Health
MRC	– Mosquito Release Container
MVC	– Model – View – Controller
OVI	– Ovum
PHP	– Hypertext Pre Processor
RUP	– Rational Unified Process
SDK	– Software Development Kit
URL	– Uniform Resource Locator
WMP	– World Mosquito Program

Chapter 01: Introduction

This chapter provides an introduction of the project including project overview, motivation, objectives, scope of the study, and structure of the dissertation.

1.1 Project Overview

“The World Mosquito Program (WMP) is a not-for-profit initiative that exists to protect the global community from mosquito-borne diseases. WMP uses safe and natural bacteria called Wolbachia to prevent the transmission of mosquito-borne diseases such as Dengue, Zika, Chikungunya and Yellow Fever. Currently, WMP operates one of its sites in Sri Lanka by covering two pilot Medical Officer of Health (MOH) areas such as Colombo and Nugegoda” (World Mosquito Program, 2020).

Sri Lankan site has two main field activities such as field planning and field monitoring. The main field planning activities are divided as local mosquito eggs related field planning (OVI field planning), local mosquitoes related field planning (BG field planning) and imported mosquito eggs related field planning (MRC field planning). The local mosquito eggs related field planning is divided as location identification and eggs collection. The local mosquitoes related field planning is divided as location identification and mosquito collection. The imported mosquito eggs related field planning is divided as location identification and eggs releasing. The main field monitoring activities are divided as mosquito identification (diagnostic), screening (lab test results of Wolbachia level of eggs and mosquitoes) and biobanking export (World Mosquito Program, 2020).

The present field planning and field monitoring operations are handled by the use of partially computerized system which has given rise to many problems and is very time consuming and proven to be very ineffective. As a solution, WMP has decided to converge on an information technology solution in order to develop a comprehensive computerized system which facilitates various needs and expectations of the diverse stakeholders of the system.

The proposed field planning and monitoring system is expected to cover almost all the operations and processes related to field planning and monitoring of the organization including spatial data management, field planning management, field activities management, field incidents management, mosquito diagnostic management, screening results management, biobanking exports management and reports generation.

The proposed field planning and monitoring system consists of mobile application and a web based application. The proposed system is developed in accordance to object oriented techniques employing various UML diagrams from modelling the system and adhering to the MVC architecture. Meanwhile, the system is developed according to RUP framework. Android Studio is used as the integrated development environment to develop the proposed mobile application by targeting Android platform. Also, Java is used as the programming language inside Android Studio for the development of proposed mobile application. XAMPP web server solution stack is used to develop the proposed system utilizing Apache as the web server. Codeigniter is used as the server-side framework and JavaScript is used as client-side programming language to build the web application while MySQL is used for handling the centralized database. Also, this proposed system uses GIS (Geographic Information Systems) technologies for spatial data management. Ultimately, the developed system is thoroughly tested using a comprehensive testing procedure and evaluated to determine whether the system and user requirements are accomplished. Therefore, it can be understood that the developed system confirms to its specification whilst providing the intended functionality and anticipated outcome.

1.2 Motivation

The existing field planning and monitoring system of the organization has several weaknesses and drawbacks since it handles substantial part of its operations using a partially computerized system. Following is the list of weaknesses and drawbacks which occurs in the existing field planning and monitoring system of the organization:

- In the manual process, large set of data are recorded in paper-based files. Therefore, the data flow of the manual process is very inefficient. Also, the results of the manual process may be inaccurate and unreliable.
- Data redundancy and data anomalies occur in both paper-based file system and flat file system (spreadsheet-based file system).
- The absence of accurate, meaningful statistical reports and graphical charts which can be generated at any time.
- Poor communication between field coordinators, GIS (Geographic Information Systems) team and field assistants.
- Difficult to perform data validation and verifications correctly.
- The free software which are currently being used by the organization, cannot be customized satisfying entire requirements of the organization.

- Data are not stored in a centralized database. Therefore, data manipulation and analysis are difficult to be performed.
- Difficult to track field incidents on real time.

At present, the management board of WMP expects a complete software system to overcome above mentioned operational drawbacks and weaknesses.

1.3 Objectives

The main goal of this project is to develop a mobile application and a web application to eliminate the drawbacks and weaknesses of the existing process. The objectives of the project are as follows:

- To handle data collections related to local mosquito eggs related field activities, local mosquitos related field activities and imported mosquitos related field activities through the mobile application.
- To record field incident details through the mobile application and manage those details through the web application.
- To administrate field planning activities through the web application.
- To handle report generation through the web application.
- To record and maintain bio-banking exports through the web application.
- To record and maintain mosquito identifications (diagnostics) through the web application.
- To record and maintain screening results through the web application.
- To manage spatial data related to field planning through the web application.

1.4 Scope of The Study

- The mobile application should be able to be used by the field assistants as a data collection tool for all types of field activities such as imported mosquitos related field activities, local mosquitos related field activities and local mosquito eggs related field activities.
- The field assistants should be able to capture field incidents using the mobile application and the field coordinators should be able to manage those field incidents using the web application.

- GIS officers should be able to manage the administration of field planning activities using the web application.
- GIS officers should be able to manage all required spatial data using the web application.
- The web application should be able to generate MIS reports for different stakeholders of the system such as field activities performance reports, diagnostic report, screening report and etc.
- The entomology officers should be able to manage mosquito diagnostic (identification) results using the web application.
- The molecular biologists should be able to manage screening results using the web application.
- The molecular biologists should be able to manage bio banking exports using the web application.
- The web application should provide a dashboard which shows the current situation of field planning and monitoring to all stakeholders of the system.

1.5 Structure of The Dissertation

The background chapter covers requirements analysis, review of similar systems, quality of the proposed solution and selected process model. The design architecture chapter covers design solution, system architecture, design approach, high-level use-case diagram of proposed system, use-case diagrams and use-case narratives for module design, class diagram, sequence diagrams, database design, user interface design and MIS reports. The implementation chapter covers implementation environment (hardware / software), development tools, technologies used for development, reused components and major code segments. The evaluation chapter covers testing overview, test cases and test results and user evaluation. The conclusion chapter covers objectives achieved, problems encountered, lessons learnt and future improvements.

Chapter 2: Background

2.1 Introduction

This chapter includes requirement analysis, review of similar systems and selected process model.

2.2 Requirement Analysis

2.2.1 Analysis of Existing System

The main stakeholders of the existing system are GIS (Geographic Information Systems) officers, field assistants, entomology officers, molecular biologists and field coordinators. GIS officers involve for the spatial data management, field planning management and field activities management. In spatial data management, GIS officers create and manipulate several map layers using some desktop GIS software platforms such as ArcGIS and QGIS. Also, GIS officers publish those map layers to a web based mapping application called Google My Maps when map layers are required to be shared with other stakeholders of the system. In field planning, GIS officers plan several field activities such as MRC (Mosquito Release Container) runs, BG (Biogents Sentinel) runs and OVI (Ovum) runs using Google My Maps for setting up spatial data related to field runs and Kobo Toolbox for setting up attribute data related to field runs. Field assistants visit the field runs assigned to them to complete their field activities. In field activities, field assistants use Google My Maps mobile application for visualizing the field locations and navigation purposes. Also, field assistants use Kobo Collect mobile application to collect relevant field data in field activities. At the end of each day, field assistants synchronize collected field data from their Kobo Collect mobile application to Kobo data repository. Then, GIS officers log into the Kobo data repository and download the field data and insert them into the spreadsheets. Meanwhile in the field activities, field assistants record details of different types of field incidents to paper-based files and handover those paper-based files to the field coordinator at the end of each day. Then, field coordinator insert those field incident details to spreadsheets. When the field coordinator take actions for those field incidents, relevant spreadsheets are updated by the field coordinator. In mosquito identification, entomology officers diagnose the type of collected local mosquitos and store those diagnostic results in the paper-based files. At the end of the week, entomology officers insert those diagnostic results stored in paper-based files to the spreadsheets. In the screening process, molecular biologists identify the level of the Wolbachia of the collected local

mosquitos and store those screening results in the paper-based files. At the end of the week, molecular biologists insert those screening results stored in paper-based files to the spreadsheets. Also, according to the instructions of the high-level management, molecular biologists export some of the collected local mosquito eggs to the biobank. Molecular biologists record those biobank export details in paper-based files. GIS officers, field coordinator, entomology officers and molecular biologists prepare weekly or monthly reports to submit them to high level management of WMP for making strategic decisions. According to the analysis of existing system, some weaknesses and drawbacks are occurred in the existing system which are mentioned in section 1.2 motivation. Therefore, the management team of WMP expects a complete software system to satisfy all the field planning and monitoring requirements by eliminating weaknesses and drawbacks of existing system.

2.2.2 Functional Requirements of Proposed System

“The functional requirements are the functions and services which should be contained in a software system to satisfy the business requirements of the stakeholders of the system” (GeeksForGeeks, 2020). The functional requirements of the proposed field planning and monitoring system for WMP Sri Lankan Site are as follows:

- Field assistants should be able to use the system as a data collection tool, map viewer and navigation tool inside the field through their mobile devices.
- Field assistants should be able to capture the field incidents and submit those incident details to the field coordinator using system through mobile device.
- GIS officers should be able to manage field planning using the system.
- GIS officers should be able to manage spatial data using the system.
- System should provide different kinds of reports to different stakeholders.
- Entomology officers should be able to manage their diagnostic results using the system.
- Molecular biologists should be able to manage their screening results, biobanking exports results using the system.
- System should provide a dashboard to stakeholders which displays current progress of field planning and monitoring activities.

2.2.3 Non - Functional Requirements of Proposed System

“Non-functional requirements are the constraints on functions and services offered by the system. Those non-functional requirements are not directly concerned with the specific functions delivered by the system” (GeeksForGeeks, 2020). Following are the non-functional requirements that system is expected to achieve.

Security: Access is limited only to the authorized parties for the system.

Accuracy: System should ensure that it provides accurate data for decision making.

User friendly: System should not be complicated to be understood for the users to learn and use. It should be easy for the users to learn and use the system with minimum training.

Timeliness: The time taken for the processing and response should be relatively less.

Reliability: The system should display a high level of reliability as the users of the system will be reluctant to use the system if they found out that the system is unreliable and potential to threats.

Reusability and Maintainability: Time to time, the needs and requirements of a business are changed. At each time, it is not possible to develop the system from the scratch. Therefore, at each time, system should be able to modify the existing requirements and add new requirements.

2.3 Review of Similar Systems

Due to the nature of the industry, it was very difficult to find existing field planning and monitoring systems for mosquito programs. Following systems were studied to get an understanding of some features which are similar to some field planning and monitoring requirements of WMP.

Google MyMaps

“Google MyMaps is a feature in Google Maps that enables users to create custom maps for personal use or sharing. Users can add points, lines, shapes, notes and images on the top of Google maps. This product uses Google map as the base map. The individual persons or organizations who have Google accounts can use this service” (Google, 2020). The web application of this service can only be used for field planning requirements. In this system, the users can create custom web maps by importing data through following types of files.

- CSV
- XLSX

- KML
- GPX

Other than importing data, Google (2020) says that Google MyMaps provides following features to the users.

- Drawing: adding points or drawing shapes anywhere.
- Searching: finding places and saving them to the maps.
- Personalization: users can select different styles for maps with icons and colors. Also, users are allowed to add photos and videos to any place of the maps.
- Sharing: users can share existing maps with other users who have Google accounts.

Following figure 2.1 is a user interface of Google MyMaps:

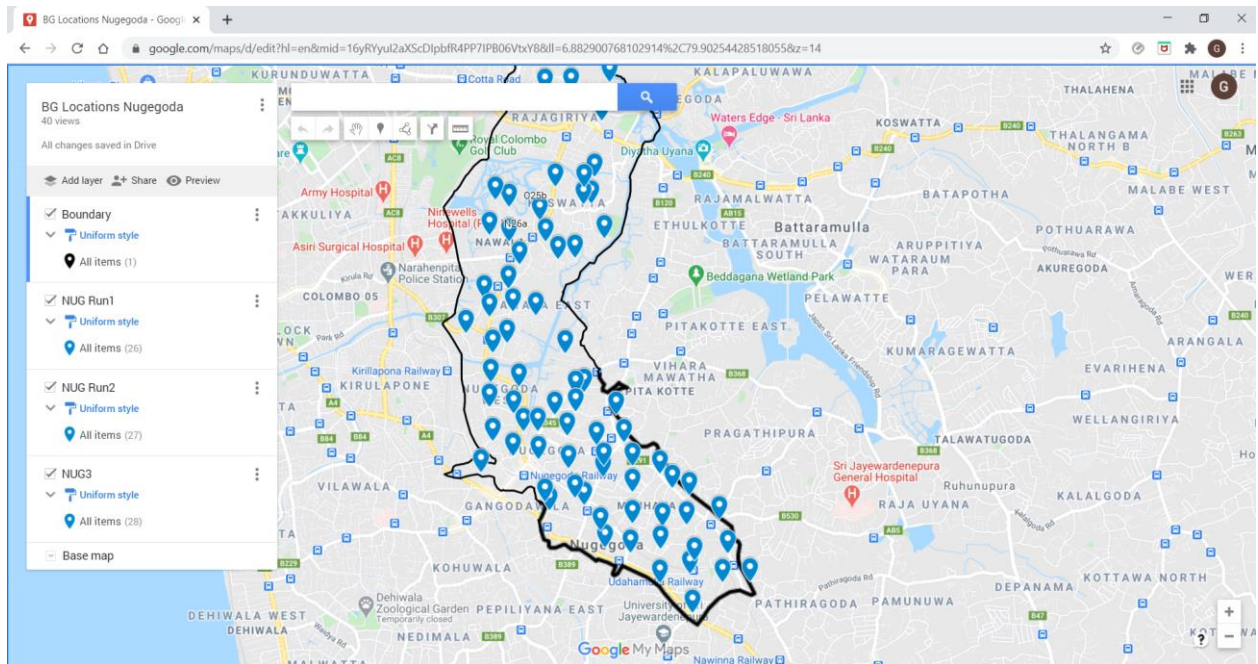


Figure 2.1 A user interface of Google MyMaps

Also, Google MyMaps provides a mobile app for the users to track places, visualize maps and navigation (Google, 2020).

Following figure 2.2, figure 2.3 are some of the screenshots of the user interfaces of that mobile application.

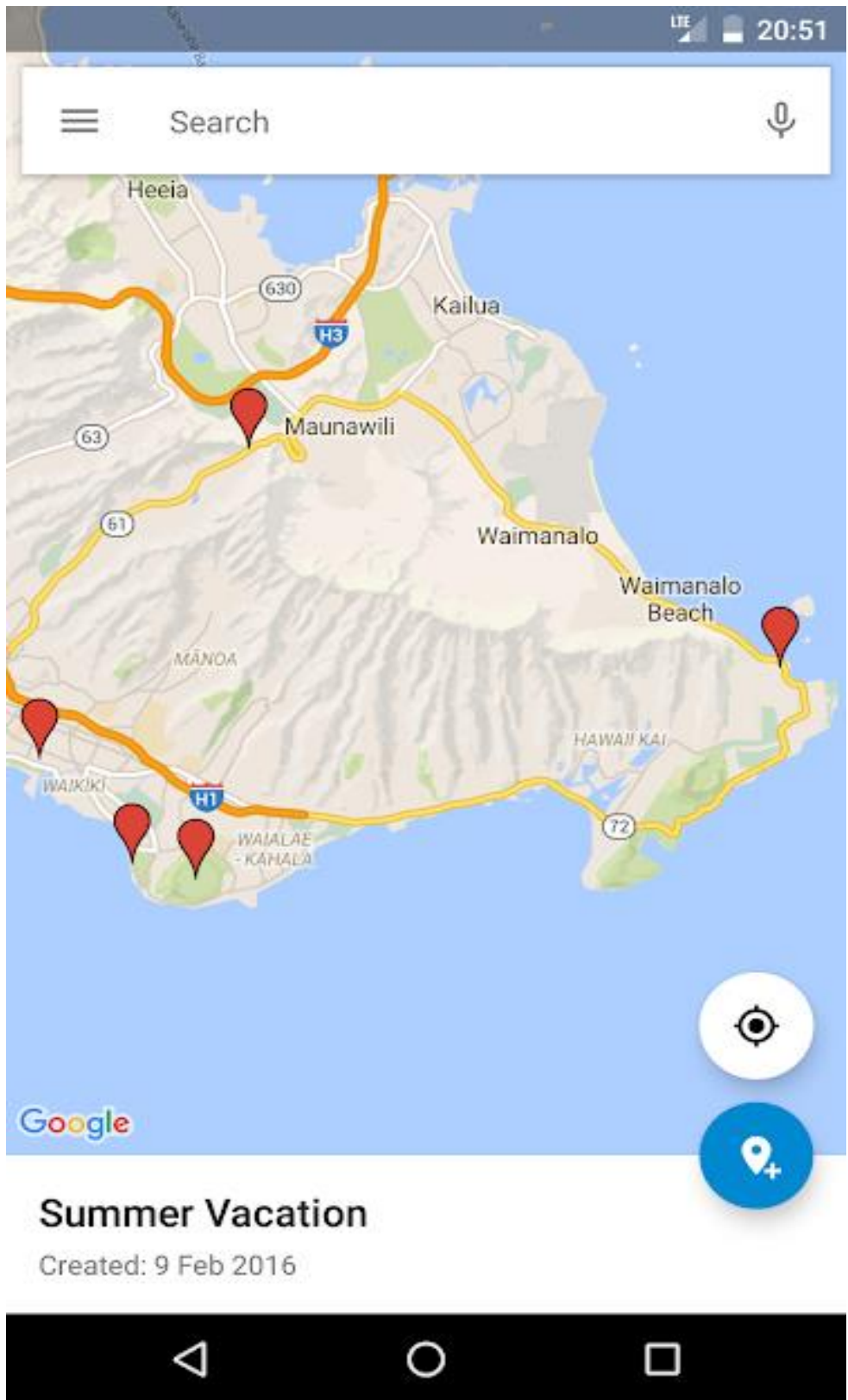


Figure 2.2 User interface of map view of Google MyMaps mobile application

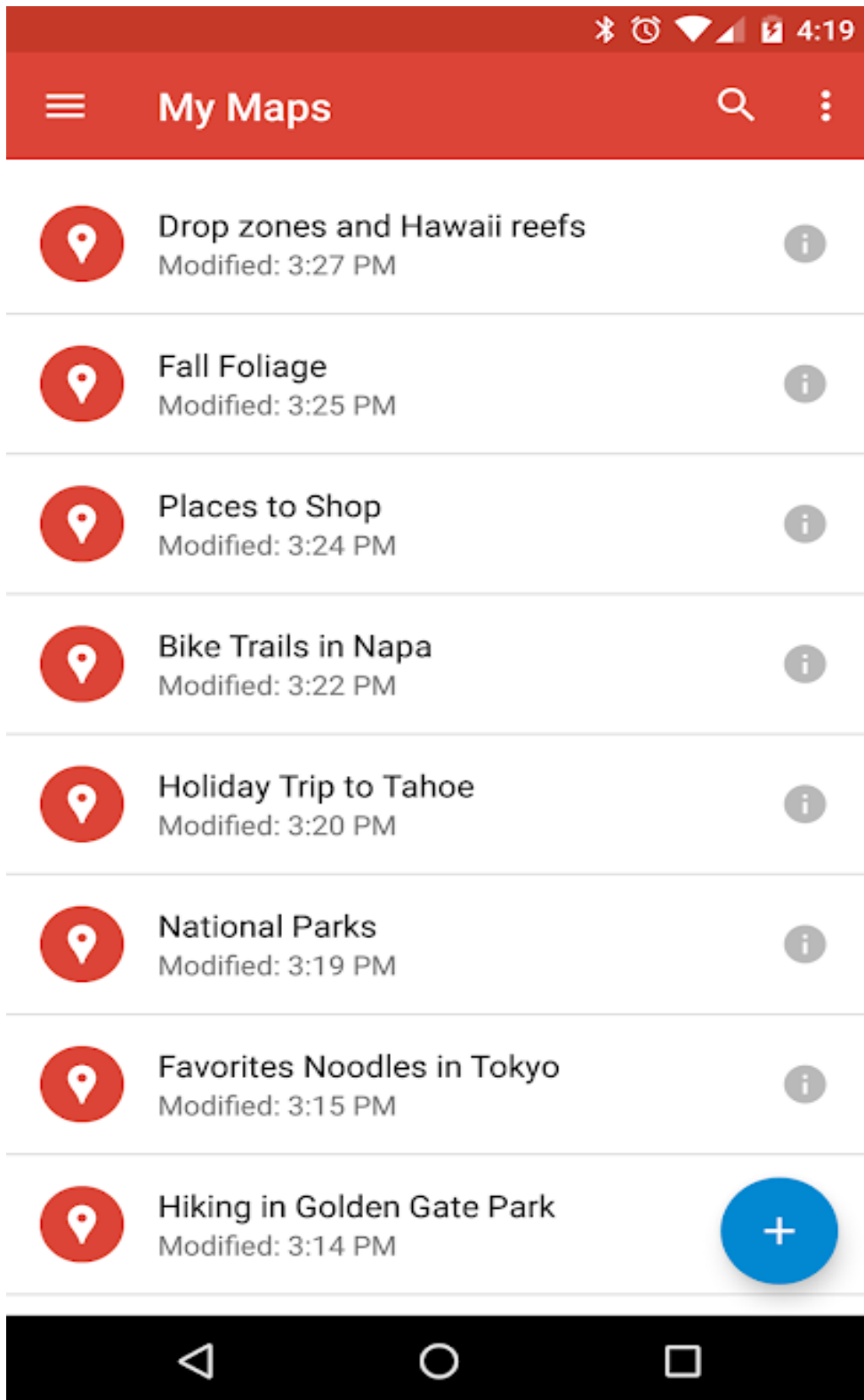


Figure 2.3 User interface of list view of Google MyMaps mobile application

There are some drawbacks in Google MyMaps service as follows:

- There is not an API (Application Program Interface) for Google MyMaps. Therefore, other applications cannot access the features of Google MyMaps programmatically.
- Google MyMaps does not provide data insertion facility for the users. Therefore, Google MyMaps cannot be used as a data collection tool for field planning activities of WMP (World Mosquito Program).

Compared to the current requirements of field planning management and field activities management, Google MyMaps service can satisfy following requirements up to a certain extent.

- Administration of field planning activities.
- Visualizing given map and navigation using mobile app.

Also, Google MyMaps cannot satisfy following requirements of field planning management and field activities management of WMP.

- Data collection in the field.
- Capturing field incident information and synchronize them with the centralized database on real time.

Kobo

“KoBo is a free open-source tool for mobile data collection, available to all. It allows you to collect data in the field using mobile devices such as mobile phones or tablets using the mobile application” (Kobo, 2021). Also, the users can create an admin account in the KoBo Toolbox web application and perform administrative tasks related to field works. In this system, admin user can design forms using a form builder and user of the mobile app can synchronize those forms to the mobile app called Kobo Collect. Then, the user of the mobile app can do data collection and send the collected data to remote database. Then, admin user can view the collected data in the web application and download them as .xls files. Following figure 2.4 and figure 2.5 are some screenshots of user interface designs “KoBo Toolbox” web application and “KoBo Collect “mobile application.

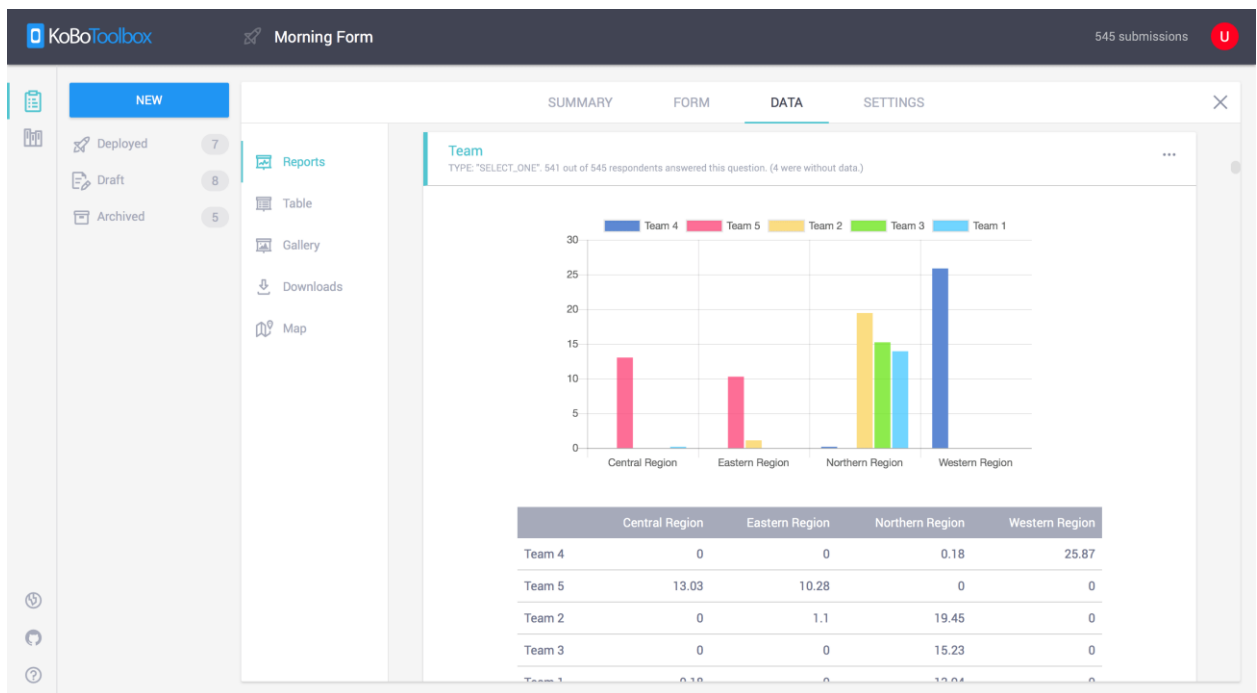


Figure 2.4 A user interface of KoBo Toolbox web application



Figure 2.5 A user interface of KoBo Collect mobile application

Compared to current requirements of field activities management of WMP, KoBo system can satisfy only data collection, data retrieving and report generation requirement. The drawbacks of KoBo system compared to the requirements of proposed system are as follows:

- KoboCollect mobile application does not provide a facility to users to transfer the collected data directly to the user's own server. Users have to access the web application of Kobo Toolbox and download the collected data. Therefore, the collected data of field activities of WMP have become insecure and the workflow become inefficient.
- This system cannot be used for map visualization, mosquito identification management, mosquito screening management.

CommCare

CommCare is a world class data collection system which includes both mobile application and web application. In this system, admin user can design forms using a form builder and user of the mobile app can synchronize those forms to the mobile app. Then, the user of the mobile app can do data collection and send the collected data to remote database. Then, admin user can view the collected data in the web application. This system has some features to improve the efficiency of the system such as field staff performance indicators, auto alerts and collaborative data collections. Also, this system allows the user to upload external data to manage data. Also, this system provides dashboard and report generation facilities to the users (Dimagi, 2020).

Compared to current requirements of field activities management of WMP, CommCare system can satisfy only data collection, data retrieving and report generation requirement. The drawbacks of CommCare system compared to the requirements of proposed system are as follows:

- CommCare mobile application does not provide a facility to users to transfer the collected data directly to the user's own server. Users have to access the web application of Kobo Toolbox and download the collected data. Therefore, the collected data of field activities of WMP have become insecure and the workflow become inefficient.
- This system cannot be used for map visualization, mosquito identification management, mosquito screening management.

2.4 Comparison between Similar Systems and Required Features of Proposed System

Following table 2.1 is a comparison between each similar system and the required features of the proposed solution.

Table 2.1 Comparison between similar systems and required features of proposed solution

Required features of proposed solution	Google MyMaps	KoBo	CommCare
Visualizing given locations using mobile app	√	X	X
Collecting data of field activities in the field runs	X	√	√
Capturing field incident information and updating those information on remote centralized database on real time	X	X	X
Administration of field planning activities	√	X	X
Report generation	X	√	√
Managing biobanking exports	X	X	X
Managing mosquito identifications	X	X	X
Managing screening results	X	X	X
Providing a dashboard which represents current situation of field activities and performance analysis of field activities	X	√	√

Required features of proposed solution column of above table represents all the features expected by the high-level management of WMP from the proposed solution. Visualizing given locations using mobile app, collecting data in the field, capturing field incident information and updating those information on remote centralized database on real time and administration of field planning activities are the required features related to field planning. But, according to above comparison, there are no any similar systems which can satisfy all four required field planning features. Also, capturing field incident information and updating those information on remote centralized database on real time is a required field planning feature which cannot be satisfied by any similar systems. The importance of the proposed solution is that is developed to satisfy all four required field planning features. Managing biobanking exports, managing mosquito identifications and managing screening results are the required features related to field monitoring. According to above comparison, all of the similar systems are unable to satisfy at least one required field monitoring feature. The importance of the proposed solution is that is developed to satisfy all three required field monitoring features. Kobo and

CommCare are the similar systems which provide report generation facility. But those similar systems provide limited types of reports and do not provide different types of reports according to the user's own requirements. The importance of the proposed solution is that provides different types of reports based on the specific requirements of WMP. Also, Kobo and CommCare are the similar systems which provide dashboard facility. But those dashboards provide limited information which do not satisfy the requirements of WMP. The importance of the proposed solution is its dashboard provides all the information required by WMP. Finally, it is proved that proposed solution is the only solution which can satisfy all the requirements of WMP.

2.5 Selected Process Model

Rational Unified Process (RUP) was chosen as the process model of this project. Following are the reasons for choosing RUP as the process model for the proposed solution.

- Accurate document can be provided.
- All the phases of software development life cycle can be achieved.
- Availability of more online resources for training.
- Less development time due to the reuse of components.

Following are the main phases of RUP and how those phases are utilized in this project.

Inception phase: After discussing with the high-level management team of WMP and stakeholders of existing system, following tasks have been done in this phase.

- Requirements gathering and understanding
- Defining the scope of the system
- Defining the objectives of the system
- Identifying the stakeholders of the proposed system
- Defining the project timeline

Elaboration phase: Design specifications were prepared in this phase. System modelling was done using UML (Unified Modelling Language) to design use-case diagrams, class diagrams and sequence diagrams. Also, database design and user interface design were done in this phase.

Construction phase: Design specifications were converted into an executable system which consists of a mobile application and a web application using several technology stacks.

Transition phase: Proposed mobile application and web application will be tested for finding defects. Then, system modifications will be done. Acceptance test will be done by the stakeholders of the system and high-level management team of WMP. Then, proposed mobile application will be deployed to mobile devices of field assistants and proposed web application will be deployed to a server space purchased from dedicated hosting service.

2.6 Summary

At the beginning of this chapter, existing system of the organization has been analyzed by specifying main stakeholders of the existing system, how each stakeholder interacts with existing system including their functions, workflow of existing system, weaknesses of existing system and purpose of expecting a new system. Then, functional and non-functional requirements which are expected by stakeholders from proposed solution, have been specified. It was very difficult to find a system which covers all the requirements of the proposed system. Therefore, some system systems which cover some of the requirements of proposed solution, have been reviewed and compared with the requirements of proposed solution. Then, importance of the development of proposed solution has been described. Then, selected process model has been described with its phases and features.

Chapter 03: Design

This chapter provides detailed information about the design of the project including design solution, system architecture, design approach, high-level use case diagram of proposed system, use-case diagrams for module design, use-case narratives for module design, class diagram, sequence diagrams, database design and, interface designs and MIS reports.

3.1 Design Solution

After considering all of above requirements, it was decided to develop a mobile application and web application to satisfy client's requirements. It was decided to use the mobile application as the data collection tool for the field activities. Also, it was decided to use the web application as the data management and administration tool for field planning and monitoring activities. Following are the reasons for selecting a web application as a solution over desktop application:

- Accessible from everywhere at anytime
- Not depend on client's operating system
- Cost-effectiveness
- In this selected design solution, web services are needed to synchronize the data between mobile application and remote database. Therefore, those web services can be reused to develop the web application and it will be very effective and efficient.

Also, following are the reasons to select a native mobile application to be developed to be used by the field assistants on their mobile devices over a mobile responsive web application:

- Native mobile applications can perform tasks by spending less time than mobile web sites.
- In native mobile applications, data are stored in its local database while web applications store their data in servers. Therefore, data retrieval is easy in native mobile applications.
- Native mobile applications can personalize their contents based on the preferences of the customers.
- Mobile responsive web sites used JavaScript codes to perform most of their functions. But mobile application frameworks are faster than JavaScript code executions.

3.2 System Architecture

System architecture of the proposed system is based on Model-View-Controller (MVC) architectural design pattern. Following figure 3.1 represents the MVC architectural design pattern.

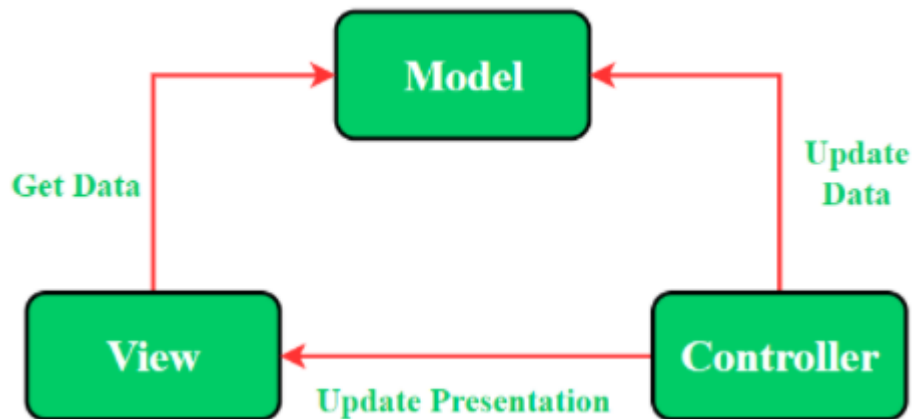


Figure 3.1 MVC architectural design pattern

Following strategy was used to design the system architecture of the proposed mobile application in Android Studio.

- Activities are used as controllers to control the views. Activities act as observers in mobile application. When an event is triggered, activity directs the model to handle further operations. When activity receives data from model, the view is updated by the activity.
- Model is a separate class which contains the data which are sent to the SQLite database or retrieved from SQLite database.

Following figure 3.2 represents the arrangement of models and controllers in system module structure of proposed mobile application in Android Studio. Following figure 3.3 represents the system module structure of the proposed web application. Also, figure 3.4 represents the system module architecture of proposed web application.

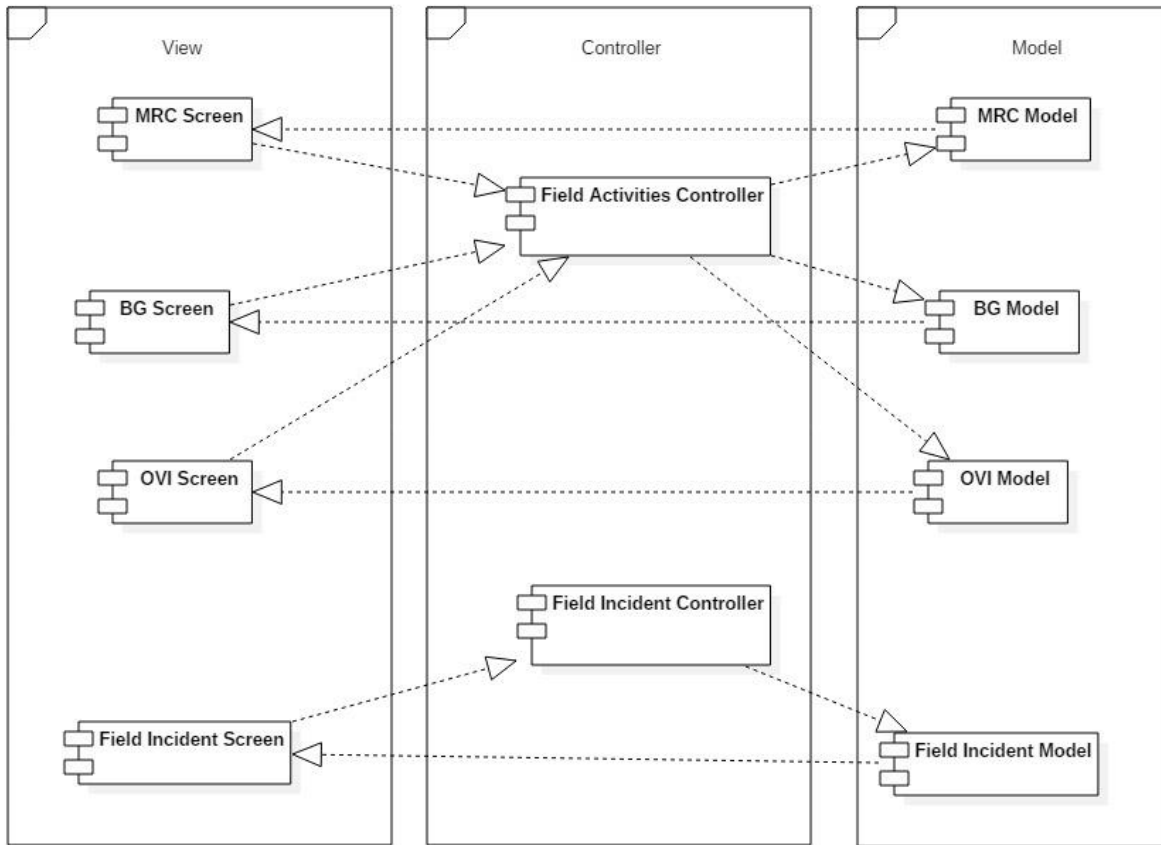


Figure 3.2 System architecture of proposed mobile application

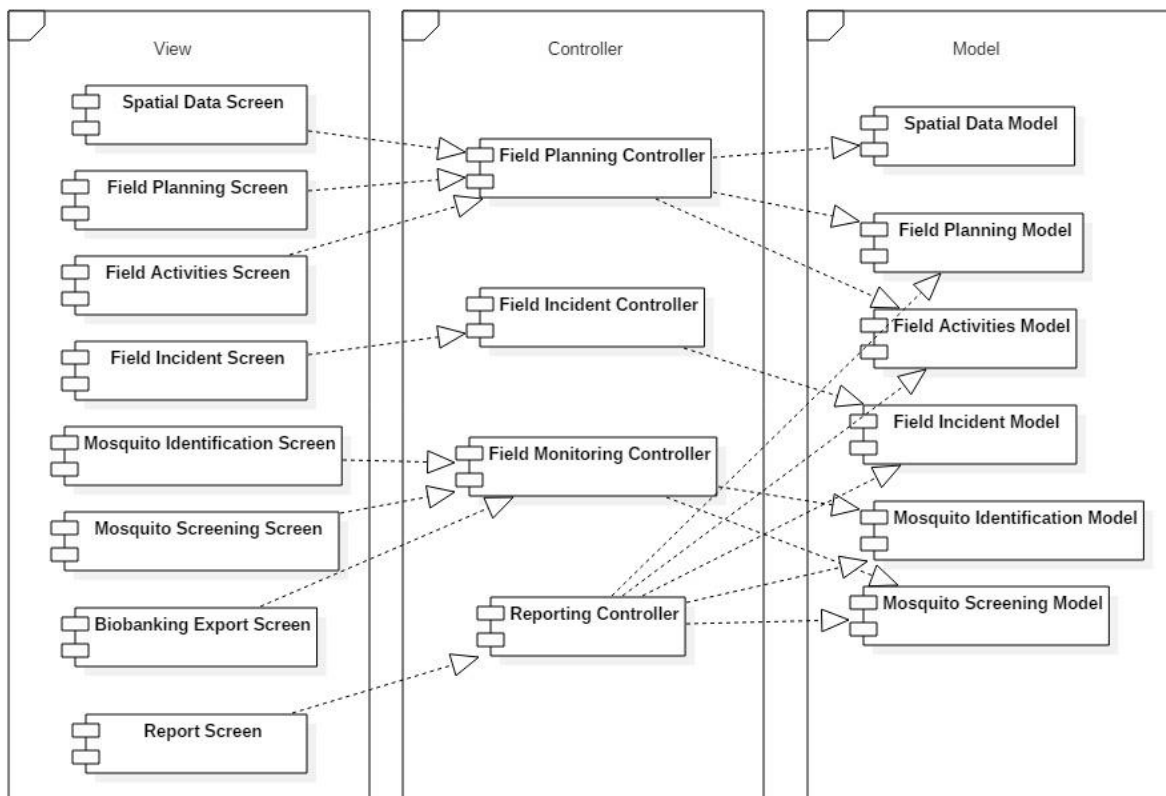


Figure 3.3 System architecture of proposed web application

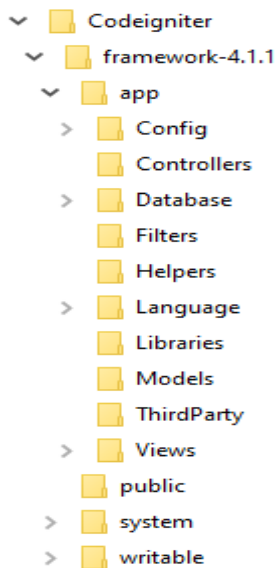


Figure 3.4 System module structure of the proposed web application

There are separate directories for model, view and controller in above system module structure of web application. View directory contains all files related to user interfaces and logics reside. Model directory contains all files of models where crud operations are carried out with

database. Controllers directory contains all the files of controllers which make interaction between models and views. Requests and responses from models are passed to the controller and the controller updates the views based on the results accordingly.

3.3 Design Approach

Design approaches provide methods to model proposed software solutions. There are two main design approaches such as structured design approach and object oriented design approach. Object oriented design approach has been selected as the design approach for the proposed system after considering the various advantages offered by object oriented design approach.

“In the object-oriented approach, the focus is on capturing the structure and behavior of information systems into small modules that combines both data and process. The main aim of Object Oriented Design (OOD) is to improve the quality and productivity of system analysis and design by making it more usable” (Tutorialspoint, 2021).

UML (Unified Modelling Language) is a most popular method used to design the systems with object oriented approach. There are several object models defined by UML which can be used in system design process (Object Management Group, 2021). Following UML modelling diagrams have been used to design the proposed system.

Use case diagram: describes in which ways the system is used by its stakeholders.

Class diagram: used to understand the overall structure of the system.

Sequence diagram: shows the interaction between objects overtime. It represents all messages which are placed in a sequential order between objects involved in a single use case over a timeline.

3.4 High-Level Use Case Diagram of Proposed System

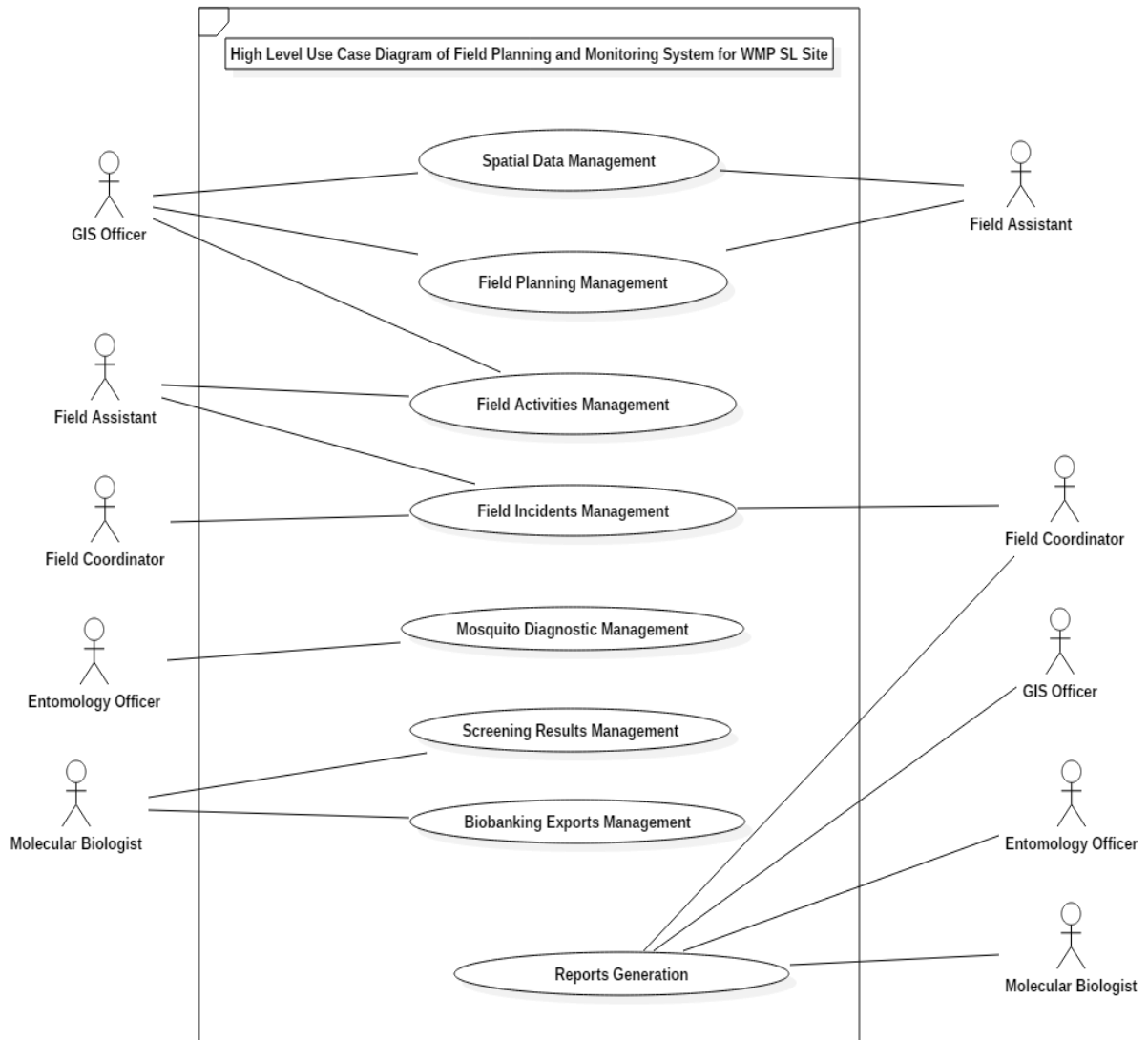


Figure 3.5 High level use case diagram of proposed system

3.5 Use Case Diagrams and Use Case Narratives for Module Design

Spatial Data Management Module

In this module, GIS (Geographic Information Systems) officers should be able to add new map layers to the system through the web application. Also, GIS officers should be able to delete, update existing map layers in the system through web application. Also, field assistants should be able to view existing map layers through the mobile application. Following figure 3.6 is the use case diagram for spatial data management module.

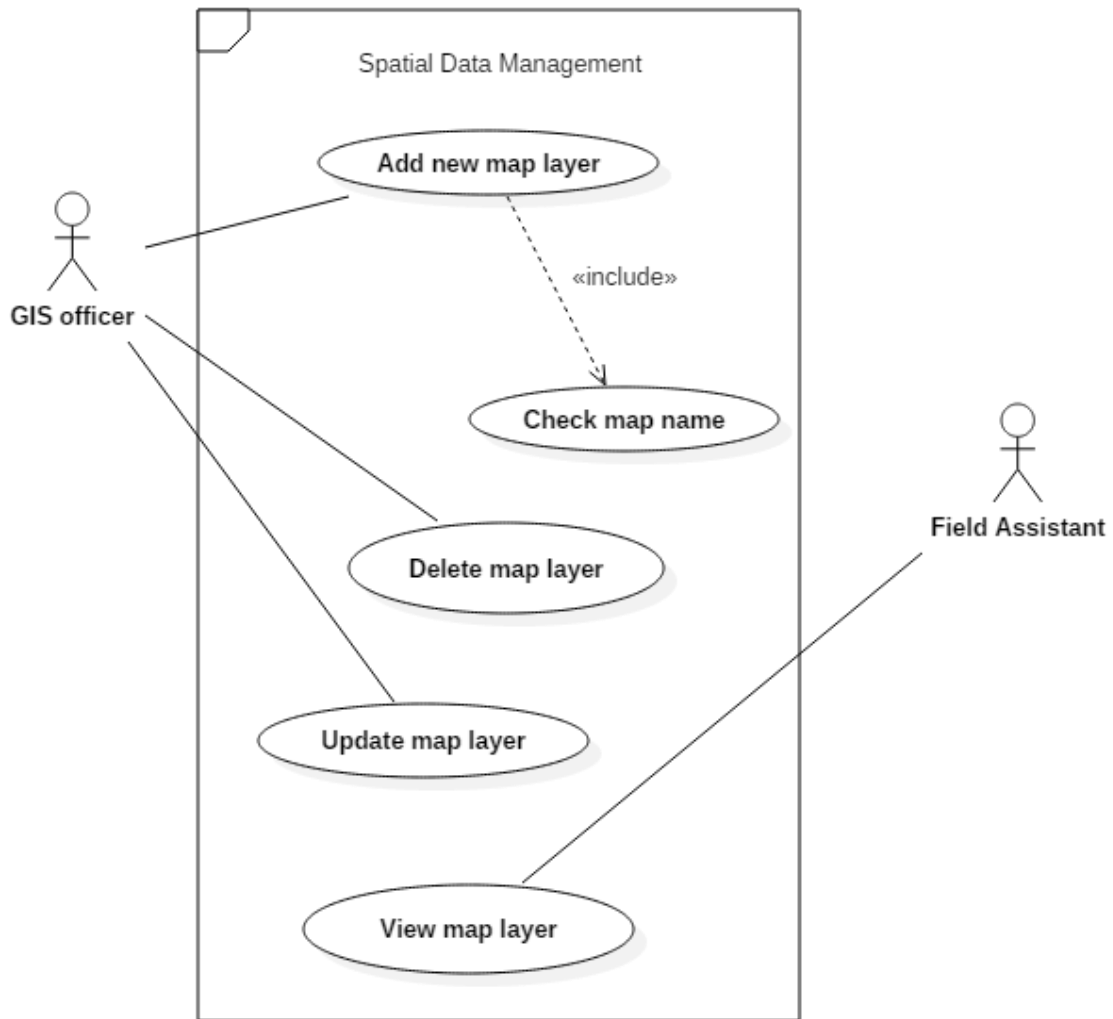


Figure 3.6 Use case diagram of spatial data management module

Following table 3.1 is the use-case narrative for the “Add new map layer” use case of spatial data management module.

Table 3.1 Use-case narrative of add new map layer

Use case name	Add new map layer
Actors	GIS officer
Description	Describes how GIS officer add a new map layer to the system through web application.
Normal sequence of events	1) GIS officer opens the form of new map layer in web application.

	<ol style="list-style-type: none"> 2) GIS officer fills the form and submits it. 3) System checks the given name of new map layer. 4) New map layer is added to the system.
Alternative sequence of events	<ol style="list-style-type: none"> 4) If the given name of map layer is already existing in the system then user is notified to change the name of the new map layer.
Pre-conditions	<ul style="list-style-type: none"> • User should give a name which is not existing in the system to the map layer.
Post-conditions	<ul style="list-style-type: none"> • Relevant tables of the database should be updated.
Assumptions	<ul style="list-style-type: none"> • User may give a name which is existing in the system to the map layer.

Field Planning Management Module

In this module, GIS officer should be able to add different types of field runs such as MRC (Mosquito Release Container) runs, BG (Biogents Sentinel) runs and OVI (Ovum) runs through the web application. Also, GIS officers should be able to delete existing active field runs from the system through web application. Also, GIS officers should be able to update existing field run details in the system through web application. Field assistants should be able to view existing active field runs through mobile application. Following figure 3.7 is the use-case diagram of field planning management module.

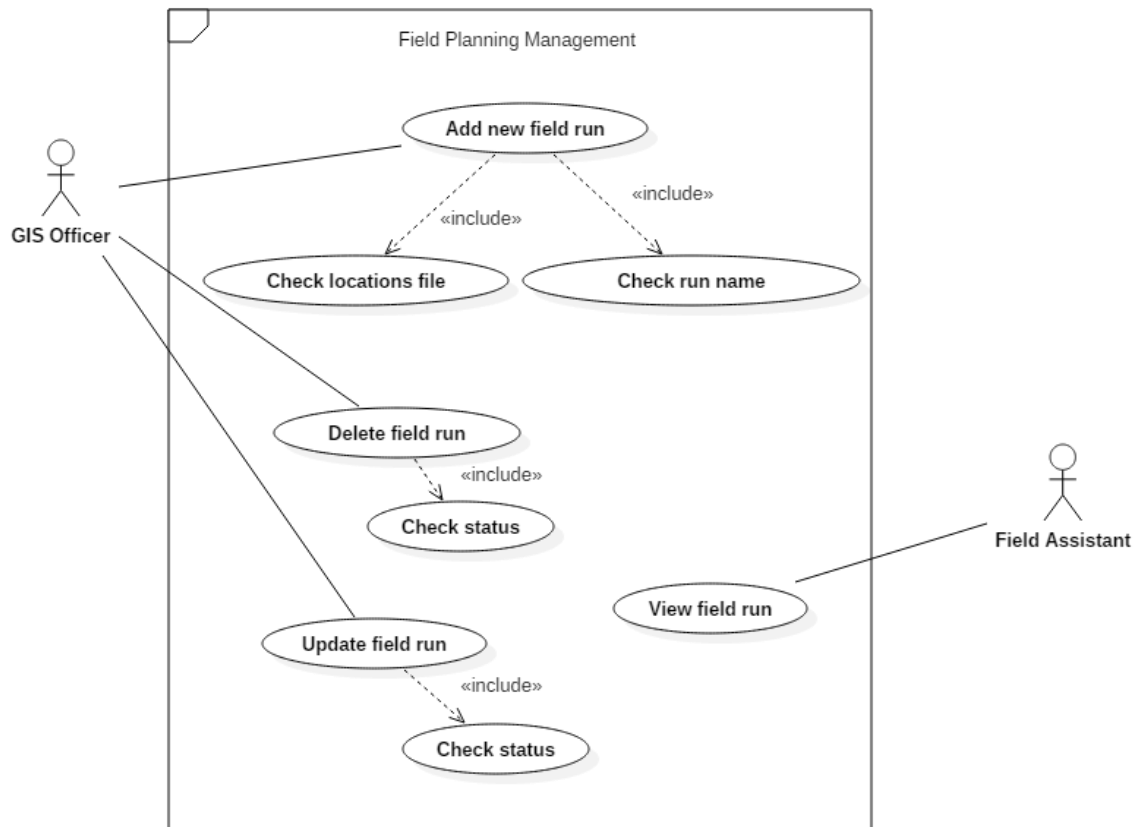


Figure 3.7 Use-case diagram of field planning management module

Following table 3.2 is the use case narrative of “Add new field run” use case of field planning management module.

Table 3.2 Use-case narrative of add new field run

Use case name	Add new field run
Actors	GIS officer
Description	Describes how GIS officer add new field run to the system.
Normal sequence of events	<ol style="list-style-type: none"> 1) GIS officer opens the form of new field run in web application. 2) GIS officer submits the form to the system after entering basic details and uploading location details file. 3) System checks run name. 4) System checks location file. 5) New field run is added to the system.

Alternative sequence of events	<p>4) If run name is already existing in system then user will be notified an error message.</p> <p>5) If there are any invalid locations, then user will be notified an error message.</p>
Pre-conditions	<ul style="list-style-type: none"> • There should not be any invalid locations in location details file. • Run name should not be existing in the system.
Post-conditions	<ul style="list-style-type: none"> • Relevant tables in the database should be updated.
Assumptions	<ul style="list-style-type: none"> • User may include invalid locations to the location details file. • User may include run name which is already existing in system.

Field Activities Management Module

In this module, field assistants should be able to add new field location, add new collection/release, and do data synchronization through mobile application. Also, GIS officers should be able to delete/update a field location and delete/update a collection/ release through mobile application. Following figure 3.8 is the use case diagram of field activities management module.

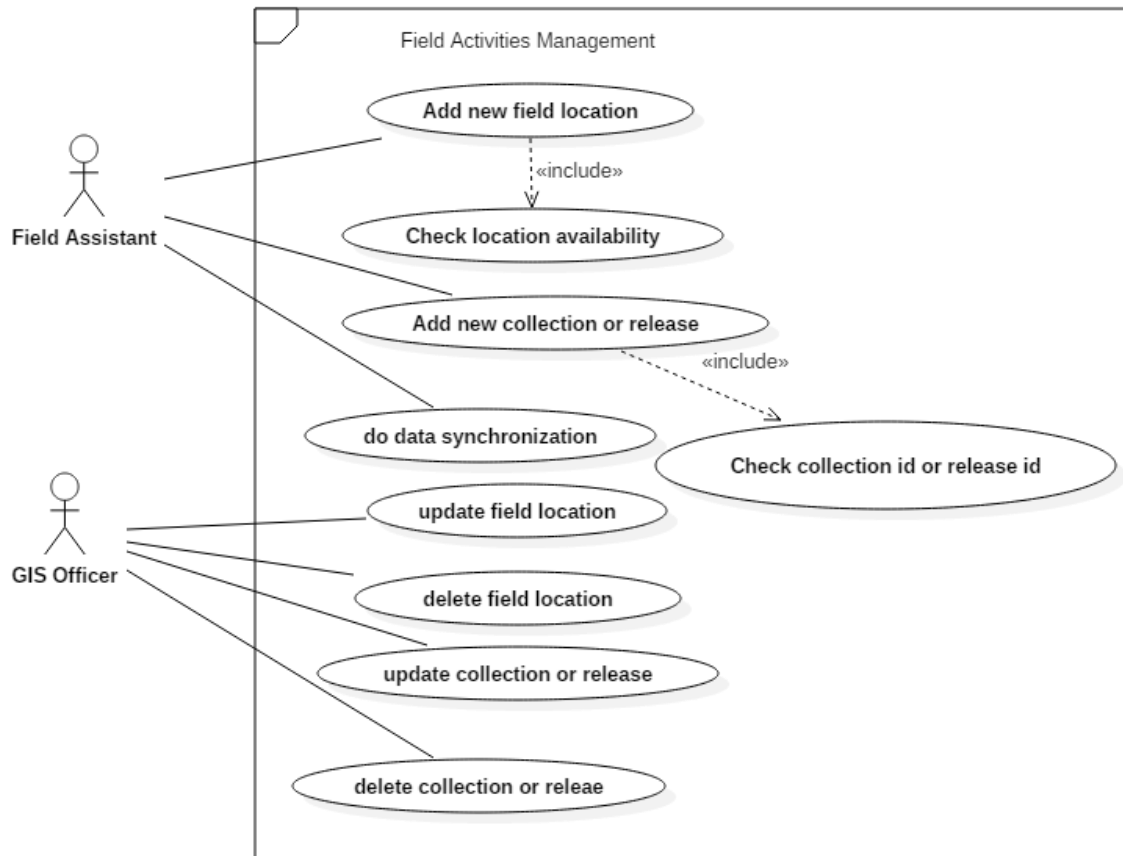


Figure 3.8 Use-case diagram of field activities management module

Followings table 3.3, table 3.4 and table 3.5 are the use-case narratives of add new field location use-case, add new collection/release use-case and data synchronization use-case.

Table 3.3 Use-case narrative of add new field location

Use case name	Add new field location
Actors	Field assistant
Description	Describes how field assistant add new field location to the system through mobile application.
Normal sequence of events	<ol style="list-style-type: none"> 1) Field assistant opens new location form of specific field activity in mobile application. 2) Field assistant fills the form and submits it.

	<p>3) System checks the availability of new field location.</p> <p>4) New field location is added to the internal database of mobile application.</p>
Alternative sequence of events	<p>4) If the identifier of new field location is already existing in the system then user is notified an error message.</p>
Pre-conditions	<ul style="list-style-type: none"> • Identifier of new field location should not be existing in the system.
Post-conditions	<ul style="list-style-type: none"> • Relevant tables of internal database of mobile application should be updated. • User should be directed to the map view of mobile application and new field location should be zoomed- in on the map view.
Assumptions	<ul style="list-style-type: none"> • User may assign an identifier which is already existing in system to new field location.

Table 3.4 Use-case narrative of add new collection/release

Use case name	Add new collection/ release
Actors	Field Assistant
Description	Describes how field assistant add a new collection/ release to the system through mobile application.
Normal sequence of events	<p>1) Field assistant selects a specific field activity and selects an available field run in mobile application.</p> <p>2) Field assistant selects a field location of selected field run from list view or map view and opens the new collection/ release form.</p>

	<p>3) Field assistant fills the form and submits it.</p> <p>4) System checks collection/ release id.</p> <p>5) New collection/ release is added to the internal database of mobile application.</p>
Alternative sequence of events	<p>5) If collection/ release id is already existing then, user will be notified an error message.</p>
Pre-conditions	<ul style="list-style-type: none"> • All field locations available for collections/ releases should be related to any available field run. • Collection/ release id should not be existing in system.
Post-conditions	<ul style="list-style-type: none"> • Relevant tables of internal database of mobile application should be updated. • User should be directed to the map view of mobile application and updated field location should be zoomed- in on the map view in relevant color based on new release/ collection status of field location.

Table 3.5 Use-case narrative of do data synchronization

Use case name	Do data synchronization
Actors	Field assistant
Description	Describes how field assistant do data synchronization between mobile application and centralized remote database.
Normal sequence of events	<p>1) Field assistant opens the mobile application and select specific field activity type.</p>

	<p>2) Field assistant selects data synchronization option.</p> <p>3) System checks compatibility between data of mobile application and data of remote database.</p> <p>4) Internal database of mobile application and remote database are updated based on new changes done in both databases.</p>
Alternative sequence of events	<p>4) If there are any conflicts between the internal database of mobile application and remote database then the user is notified to accept the changes done in remote database. If user does not agree to accept the changes of remote database then data synchronization will be terminated.</p>
Pre-conditions	<ul style="list-style-type: none"> • There should not be any conflicts between the data of internal database of mobile application and data of remote database.
Post-conditions	<ul style="list-style-type: none"> • Both remote database and internal database of mobile application should be updated.
Assumptions	<ul style="list-style-type: none"> • There may be conflicts between data of internal database of mobile application and data of remote database.

Field Incident Management Module

In this module, field assistant should be able to add new field incidents through the mobile application. Field coordinator should be able to delete, update and display field incidents through web application. Following figure 3.9 is the use-case diagram of field incident management module.

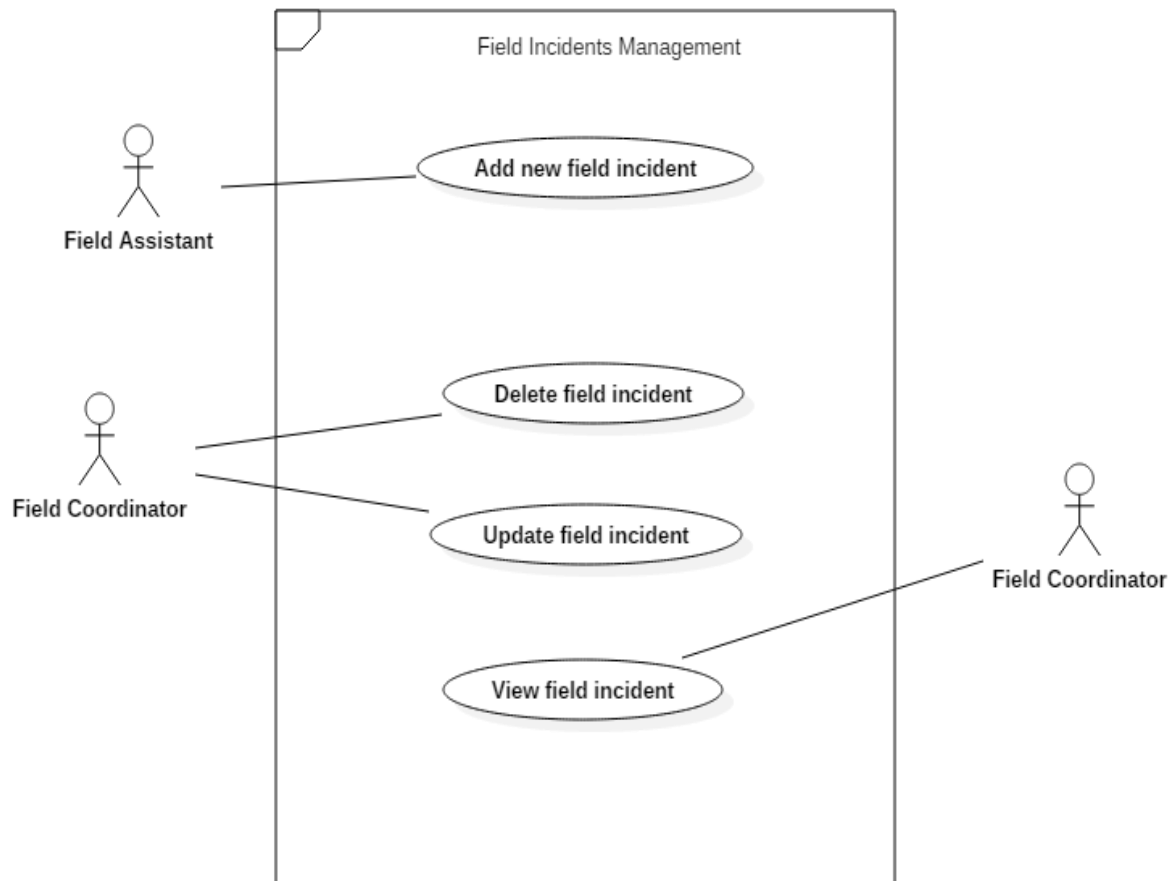


Figure 3.9 Use-case diagram of field incident management module

Following table 3.6 is the use-case narrative for add new field incident use case of field incident management module.

Table 3.6 Use-case narrative of add new field incident

Use case name	Add new field incident
Actors	Field assistant
Description	Describes how field assistant add new field incidents to the system through mobile application.
Normal sequence of events	<ol style="list-style-type: none"> 1) Field assistant opens add new field incident form in mobile application. 2) Field assistant fills the form and submits it. 3) New field incident is added to the system.

Post-conditions	<ul style="list-style-type: none"> Relevant tables of remote database should be updated.
------------------------	---

Please refer Appendix B – Design Documentation for the rest of Use-case Diagrams and Use-Case Narratives for Module Design.

3.6 Class Diagram

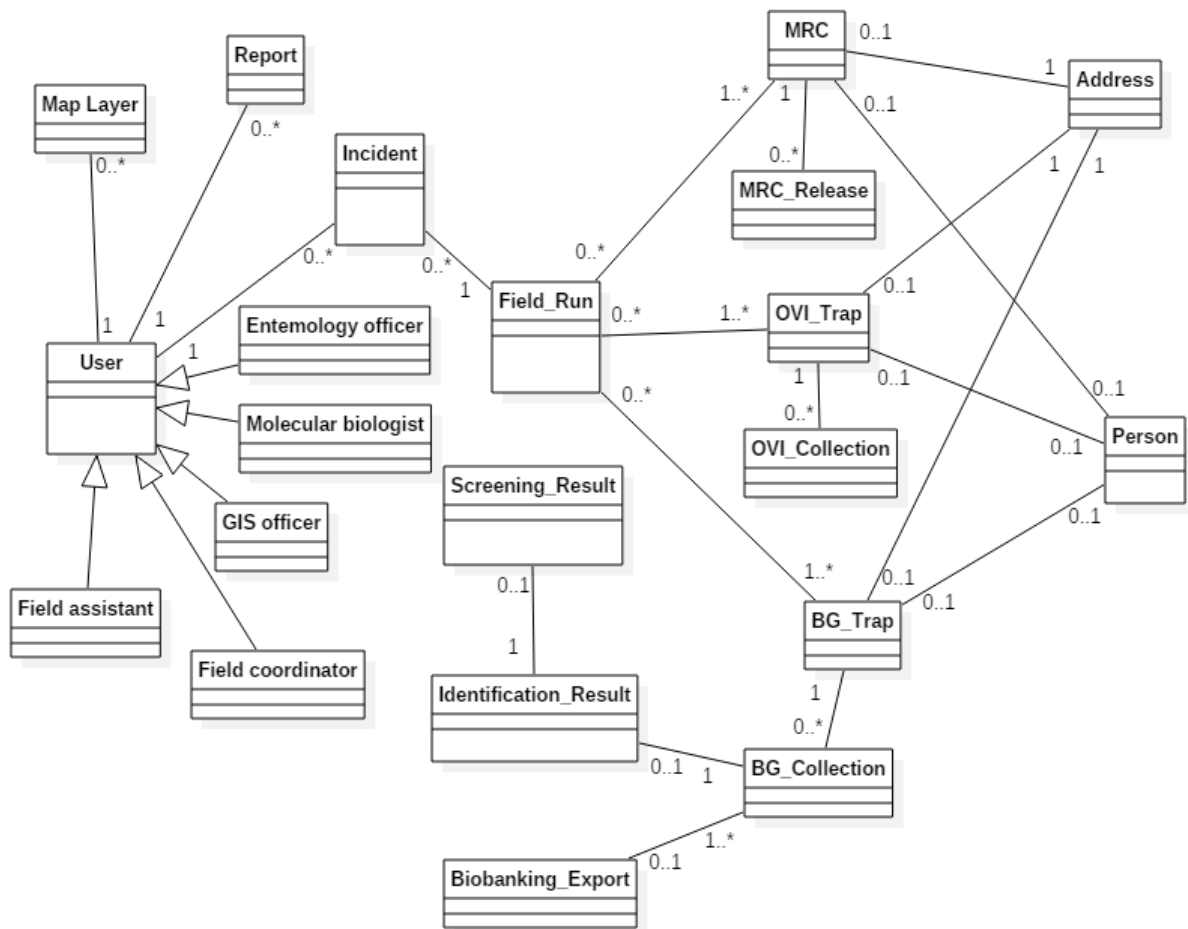


Figure 3.10 Class diagram

3.7 Sequence Diagrams

Use case name: Add new map layer

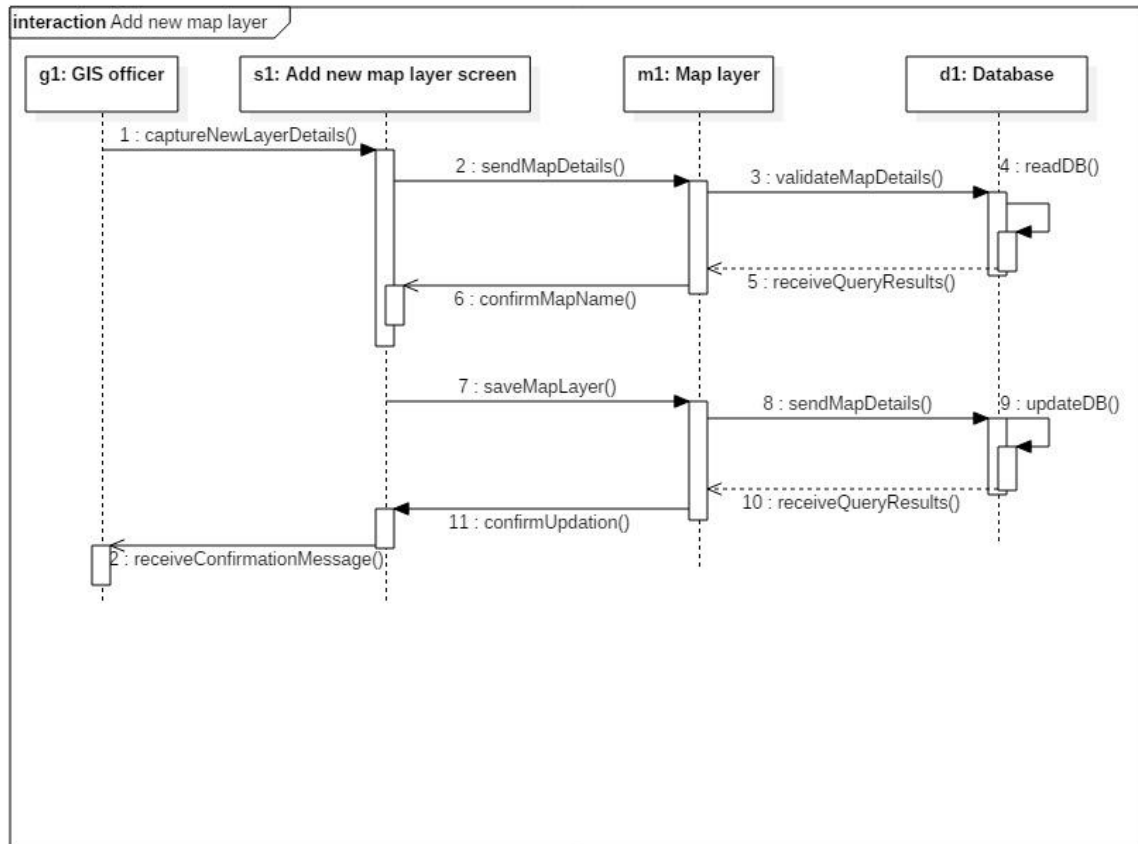


Figure 3.11 Sequence diagram of add new map layer

Use case name: Add new field run

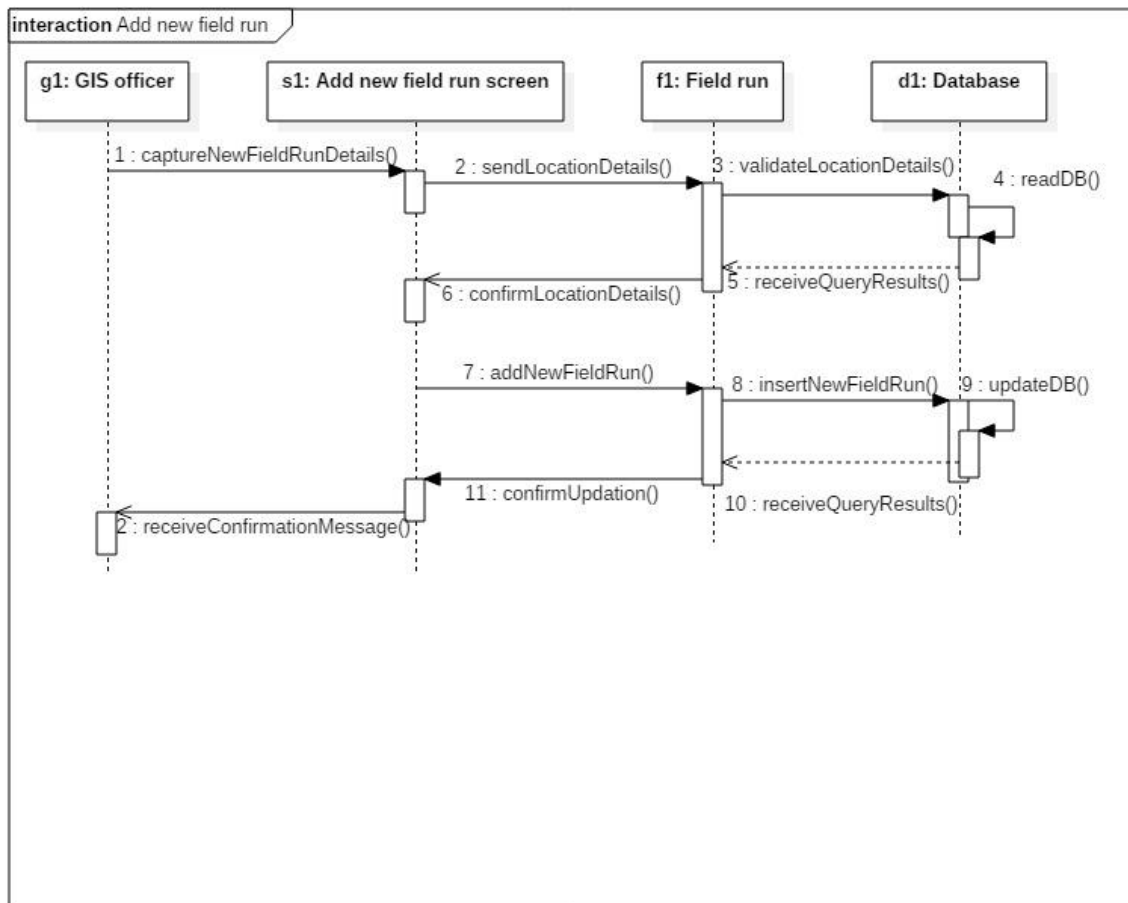


Figure 3.12 Sequence diagram of add new field run

Use case name: Add new field incident

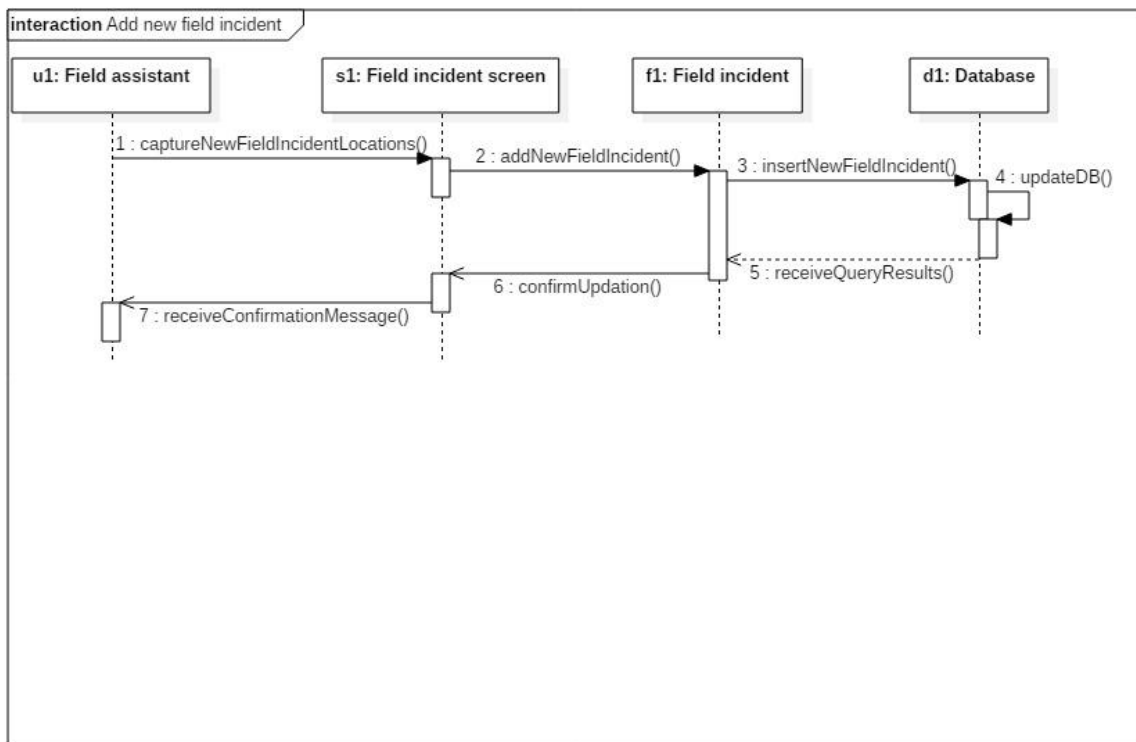


Figure 3.13 Sequence diagram of add new field incident

Use case name: Add new identification result

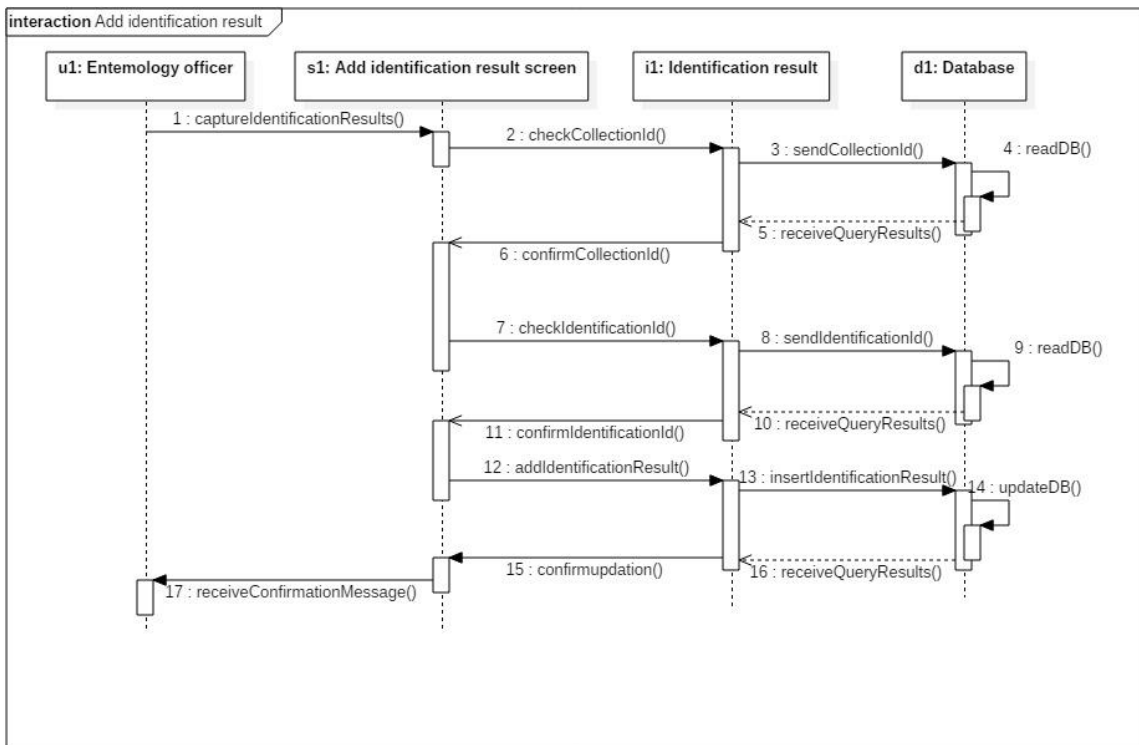


Figure 3.14 Sequence diagram of add identification result

Use case name: Add new screening result

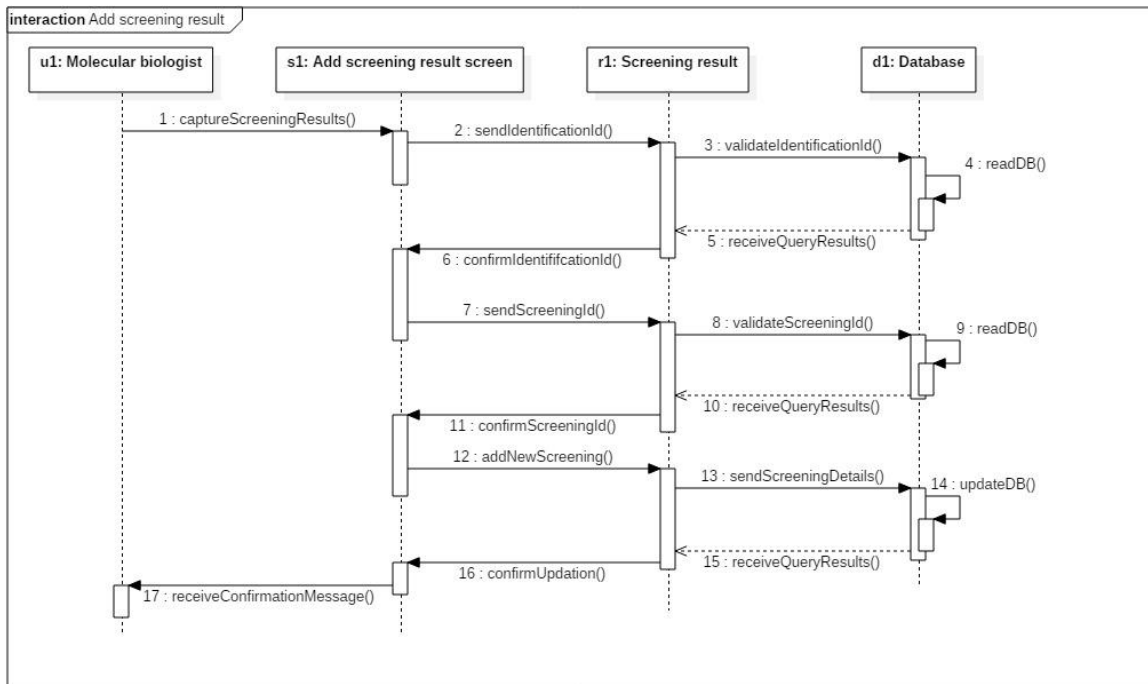


Figure 3.15 Sequence diagram of add new screening result

3.8 Database Design

3.8.1 Entity-Relationship Diagram

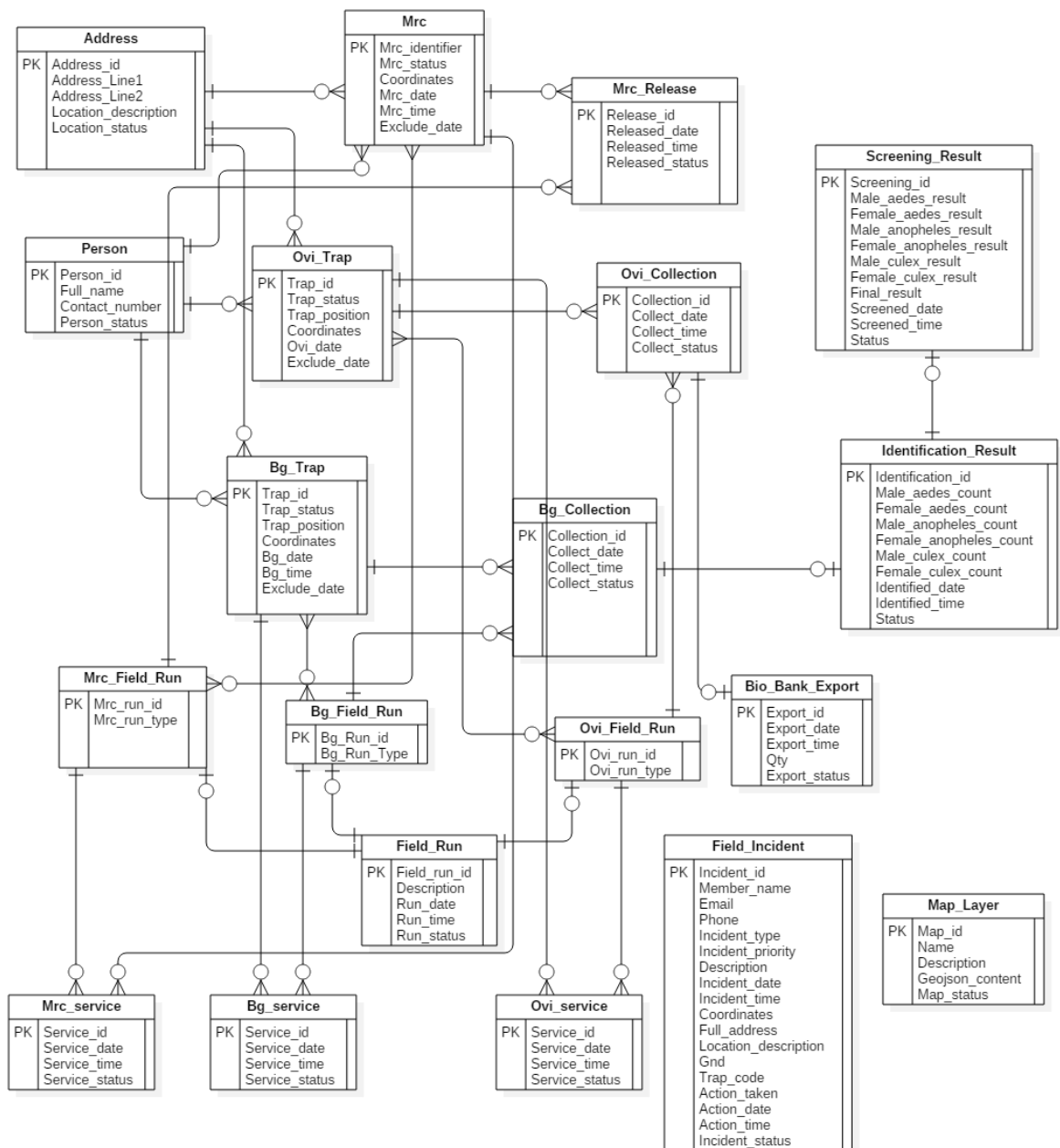


Figure 3.16 ER diagram of proposed system

3.8.2 Mapping Entity-Relationship Diagram to Relational Schemas

Address (address_id, add_line1, add_line2, location_description, location_status)

Bg_collection (collection_id, trap_id*, collect_date, collect_time, collect_status, run_id*)

Bg_field_run (bg_run_id*, bg_run_type)

Bg_run_traps (bg_run_id*, bg_trap_id*)

Bg_service (service_id, trap_id*, service_date, service_time, service_status, run_id*)

Bg_trap (trap_id, trap_status, trap_position, coordinates, bg_date, bg_time, person_id*, address_id*, exclude_date)

Biobank_export (export_id, export_date, export_time, export_status, qty, ovi_collection_id*)

Field_run (field_run_id, description, run_date, run_time, run_status)

Identification_result (identification_id, male_aedes_count, female_aedes_count, male_anopheles_count, female_anopheles_count, male_culex_count, female_culex_count, identified_date, identified_time, collection_id*, status)

Incident (incident_id, member_name, email, phone, incident_type, incident_priority, description, incident_date, incident_time, coordinates, full_address, location_description, gnd, trap_code, action_taken, action_date, action_time, incident_status)

Map_layer (map_id, name, description, geojson_content, map_status)

Mrc (mrc_identifier, mrc_status, coordinates, person_id*, address_id*, mrc_date, mrc_time, exclude_date)

Mrc_field_run (mrc_run_id, mrc_run_type)

Mrc_release (release_id, identifier*, released_date, released_time, release_status, run_id*)

Mrc_run_traps (mrc_run_id*, mrc_trap_id*)

Mrc_service (service_id, trap_id*, service_date, service_time, service_status, run_id*)

Ovi_collection (collection_id, trap_id*, collect_date, collect_time, collect_status, run_id*)

Ovi_field_run (ovi_run_id*, ovi_run_type)

Ovi_run_traps (ovi_run_id*, ovi_trap_id*)

Ovi_service (service_id, trap_id*, service_date, service_time, service_status, run_id*)

Ovi_trap (trap_id, trap_status, trap_position, coordinates, ovi_date, ovi_time, person_id*, address_id*, exclude_date)

Person (person_id, full_name, contact_number, person_status)

Screening_result (screening_id, male_aedes_result, female_aedes_result, male_anopheles_result, female_anopheles_result, male_culex_result, female_culex_result, final_result, identification_id*, screened_date, screened_time, status)

In those relational schemas, there are no any schema which contains multi-valued attributes. Therefore, all of above relational schemas are in first normal form (1 NF). Also, there are no any schema which contains partial functional dependencies. Therefore, all of above relational schemas are in second normal form (2 NF). Also, there are no any schema which contains transitive dependencies. Therefore, all of above relational schemas are in third normal form (3 NF).

3.9 User Interface Designs

Following figure 3.17 represents the user interface of main menu of proposed mobile application. There are four options in that user interface such as MRC field activities, BG field activities, OVI field activities and field incidents. Users can click on “MRC Field Activities” option to navigate to “MRC Field Activities” screen to use field runs related to MRC releases or MRC services and add new MRC locations to the system. Users can click on “BG Field Activities” option to navigate to “BG Field Activities” screen to use field runs related to BG collections or BG services and add new BG locations to the system. Users can click on “OVI Field Activities” option to navigate to “OVI Field Activities” screen to use field runs related to OVI collections or OVI services and add new OVI locations to system. Users can click on “Field Incidents” option to navigate to “Field Incidents” screen to add new field incidents to the system.



07:35

WMP Field

Menu

MRC Field Activities

BG Field Activities

OVI Field Activities

Field Incidents

Figure 3.17 Main menu user interface of proposed mobile application

Following figure 3.18 represents the “OVI Field Activities” user interface of proposed mobile application. There is a drop down which contains all available field runs related to OVI field activities. There are two icons in the top right corner of the screen such as map view icon and list view icon. When user clicks on the map view icon then, the OVI locations related to the selected field run are visualized on a map. When user clicks on the list view icon then, the OVI locations related to the selected field run are visualized as a list. There is a link as “+ Create a new place”. Users can click on that link to navigate to “Add New OVI Location” screen to add new OVI locations to the system. OVI locations related to the selected field run are displayed as a list below that link. There are two buttons at the bottom of the screen such as “Back” button and “Sync Data” button. When user clicks on “Back” button then, user is navigated to the previous step. When user clicks on “Sync Data” button then, a data synchronization is lunched between OVI data of local database of mobile application and OVI data of remote centralized database.

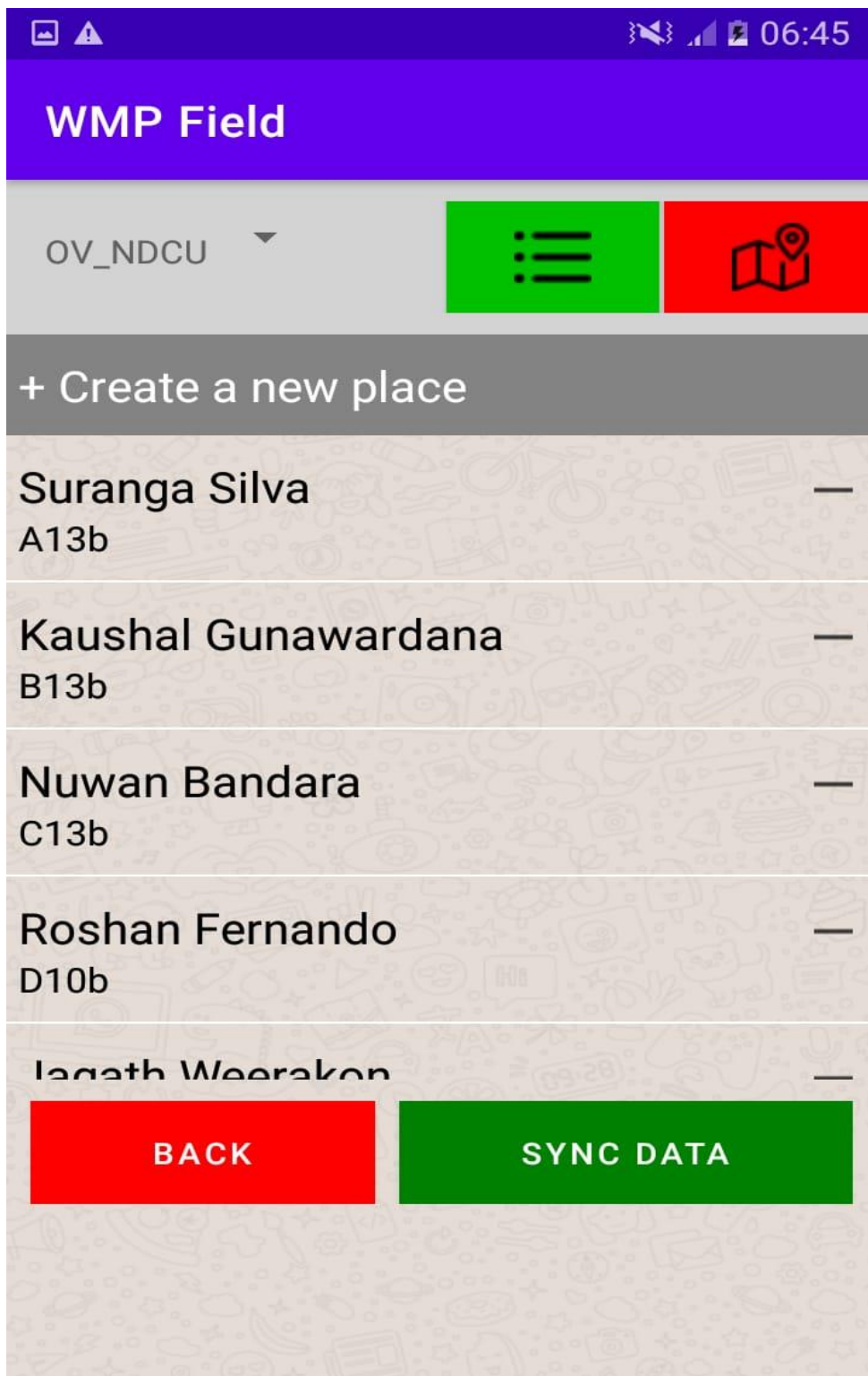


Figure 3.18 OVI field activities interface of proposed mobile application

Following figure 3.19 represents the user interface of map view of mobile application. Locations related to different types of field activities such as services, collections are displayed as points in map view. Also, new field locations can be displayed as points in that map view.

Also, different types of locations can be visualized using different colors in the map view. When user click on a location point, information window with relevant information is loaded.

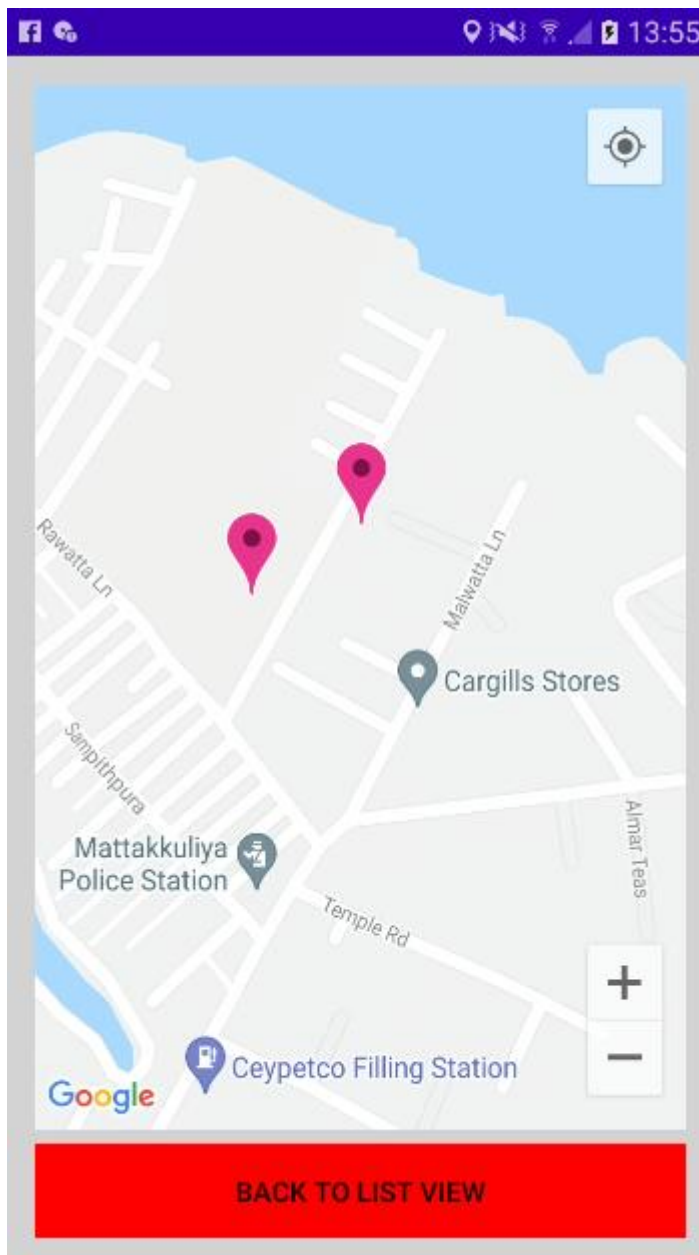


Figure 3.19 Map view of proposed mobile application

Following figure 3.20 represents the “add new identification result” user interface of proposed web application. When that screen is opened, relevant available collection ids are loaded into “Collection Id” drop down field. Then, the user should enter diagnostic id and select relevant collection id. Then, user should enter count values for male and female mosquitos of different mosquito categories such as Aedes Aegypti, Anopheles and Culex. The, user should enter identified date and identified time to relevant fields. When user clicks on “Save” button, entered

diagnostic details are inserted to the database and a success message is displayed to the user. Breadcrumbs have been used to navigate between screens of the web application.

The screenshot shows the 'Add New Mosquito Diagnostic' form in a web application. The header features the 'World Mosquito Program' logo on the left and the user name 'Gaveen Kith' with a 'Sign Out' link on the right. A blue breadcrumb trail reads 'Home / Mosquito Diagnostics / Add New Mosquito Diagnostic'. The form itself is titled 'Add New Mosquito Diagnostic' and contains the following fields:

- Diagnostic Id(*)**: A text input field with the placeholder 'Enter Diagnostic Id'.
- Collection Id**: A dropdown menu with the placeholder 'Select From Here'.
- Diagnostic Count**: A label for the following count fields.
- Male Aedes Aegypti Count(*)**: A text input field with the value '0'.
- Female Aedes Aegypti Count(*)**: A text input field with the value '0'.
- Male Anopheles Count(*)**: A text input field with the value '0'.
- Female Anopheles Count(*)**: A text input field with the value '0'.
- Male Culex Count(*)**: A text input field with the value '0'.
- Female Culex Count(*)**: A text input field with the value '0'.
- Identified Date(*)**: A date picker field showing 'dd/mm/yyyy'.
- Identified Time(*)**: A time picker field showing '--:--'.

At the bottom of the form are two buttons: a green 'Save' button and a red 'Cancel' button.

Figure 3.20 User interface of add new mosquito diagnostic of web application

Following figure 3.21 represents the “add new map layer” user interface of proposed web application. User should enter name and description of the new map layer to relevant fields. Then, user should enter GeoJson content of valid .GeoJson file by following valid format to the relevant field. When, user clicks on “Save” button entered details of new map layer are inserted into the database and a success message is displayed to the user. Breadcrumbs have been used to navigate between screens of the web application.

Add New Map Layer

Name:(*)

Description:(*)

Add GeoJSON Content Here:(*)

Figure 3.21 User interface of add new map layer screen of web application

3.10 MIS Reports

OVI Performance Report

Following figure 3.22 represents the template of OVI performance report. That report is used to get an overall summarized idea about field activities related to OVI traps for a given time period. Date range used to represent generate the report is displayed in top left corner and top right corner of the report template. Information which are provided by that report, is mainly divided for two parts such as new OVI trap locations and OVI exclusions and replacements. In new OVI trap locations part, count and percentage of count of both proposed traps and set traps are visualized statistically. Also, those information are visualized graphically using a pie chart. In OVI exclusions and replacements part, count and percentage of both replacements and exclusions are visualized statistically. Also, those information are visualized graphically using a pie chart.

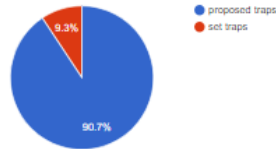
OVI Performance Report

Date From: 2020-01-01

Date To: 2021-09-30

New Ovi Trap Locations

	Count	Percentage
Proposed traps	223	90.650406500
Set traps	23	9.349593400



Ovi Exclusions and Replacements

	Count	Percentage
Replacements	1	50
Exclusions	1	50

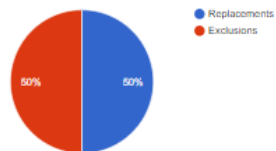


Figure 3.22 OVI performance report

Mosquito Screening Report

Following figure 3.23 represents the report template of mosquito screening report. That report provides an overall summarized idea about mosquito screening for a given period of time. That report is divided into two parts such as screening results values summary and screening results status summary. In screening results values summary part, average result of male mosquitos, average result of female mosquitos are visualized statistically for different mosquito categories. Also, those information are visualized graphically using a column chart. In screening results status summary part, count and percentage of different result status are displayed statistically. Also, those information are displayed graphically using a pie chart.

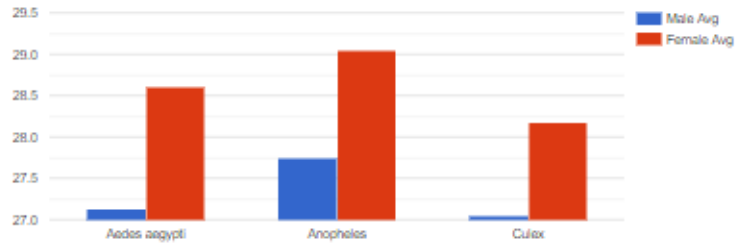
Mosquito Screening Report

Date From: 2020-01-01

Date To: 2021-10-01

Screening Results Values Summary

Mosquito Category	Male Results Average	Female Results Average
Aedes Aegypti	27.130434782608695	28.608695652173914
Anopheles	27.73913043478261	29.043478260869566
Culex	27.043478260869566	28.17391304347826



Screening Results Status Summary

Result Status	Count	Percentage
Pass	11	47.826086900
Fail	12	52.173913000

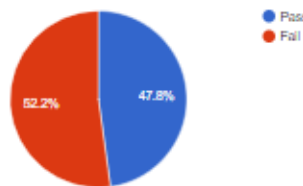


Figure 3.23 Mosquito screening report

Please refer Appendix D – MIS Reports for the rest of MIS reports.

3.11 Summary

At the beginning of the chapter, selected design solution has been described with reasons for selecting the solution and comparison with other alternative design solutions. Then, selected system architecture has been described with strategies used to apply selected system architecture to proposed system. Then, selected design approach has been described. Then, different UML diagrams such as use-case diagrams, use-case narratives, class diagram and sequence diagrams have been represented. Then, database design has been described including ER diagram and mapping ER diagram to relational schemas. Then, some selected user interface designs and some selected MIS reports have been described.

Chapter 04: Implementation

4.1 Introduction

“Implementation is the process of converting the system design into an executable program while fulfilling the requirements mentioned in the requirement specification. Thus, the ultimate goal is to develop a complete working system built upon the specification” (TutorialsPoint, 2021).

This chapter includes implementation environment, development tools, technologies, reused components and major code segments.

4.2 Implementation Environment

This describes the minimum set of hardware and software configurations for the implementation environment. The proposed system has been developed on a computer with a similar configuration. Following table 4.1 represents the implementation environment of the development process.

Table 4.1 Implementation environment

Hardware	Software
<ul style="list-style-type: none">• Intel Core i7 CPU with 2.40 GHz• 4GB RAM• 140 GB hard disc	<ul style="list-style-type: none">• Microsoft Windows 10 Pro operating system• XAMPP 3.1.4<ul style="list-style-type: none">▪ Apache HTTP Server 2.4.46▪ PHP 7.3.27▪ MySQL Server 15.1

4.3 Development Tools

Following table 4.2 represents the development tools used in the implementation.

Table 4.2 Development tools

Development Tool	Purpose
Sublime Text 2.0	For coding of web application
Android Studio 4.1.2	For coding of mobile application
Adobe Photoshop CS3	For image manipulation

ArcMap 10.4.1	For creating map layers and performing GIS tasks
Google Chrome	Used as primary web browser
XAMPP	Used as the web server solution stack for web application.

4.4 Technologies

Following table 4.3 represents set of technologies used in developing the system along with their descriptions.

Table 4.3 Technologies

Technologies	Description
CodeIgniter	CodeIgniter is a PHP framework and being used as the server-side scripting framework of the proposed web application.
MySQL	MySQL is used to implement the centralized database and handle all aspects related to the centralized database.
HTML	HTML is used to build the user interfaces of the proposed web application.
CSS	CSS is used for styling and formatting the user interfaces of proposed web application.
JavaScript	JavaScript is used as the client-side programming language of proposed web application to perform client-side validations and other client-side operations.
AJAX	AJAX is used in the client-side of proposed web application to send data to server and receive data from server without refreshing the web page.
Java	Java is used as the programming language in Android Studio to develop the proposed mobile application.

4.5 Reused Components

Following table 4.4 represents the reused components used in the implementation.

Table 4.4 Reused components

Component	Description
GitHub	GitHub is used as the code and resource repository of both development of web application and mobile application.
Git	Git is used as the version management tool in both development of mobile application and web application.
Google Charts	The web application uses Google Charts to visualize graphical charts.
LeafLet.js	Leaflet is used as the map visualization technology in the web application.
Google Maps SDK for Android	Google Maps SDK for Android is used as the map visualization technology in mobile application.

4.6 Major Code Segments

In this section, some code segments of both mobile application and web application are listed. Code segments of mobile application were written using Java in Android Studio. Code segments of web application were written in Codeigniter framework.

Code Segments of Mobile Application

Database Handling

Following figure 4.1, figure 4.2, figure 4.3, figure 4.4 and figure 4.5 are the code segments of DbHandler class which is used to handle SQLite database. In that class database connection is established, create tables, insert data to tables, query the data of tables and update tables.

```

public class DbHandler extends SQLiteOpenHelper {
    private static final int VERSION = 1;
    private static final String DB_NAME = "FieldDb";
    // table and column names for person table
    private static final String TABLE_PERSON = "person";
    private static final String PERSON_ID = "person_id";
    private static final String PERSON_NAME = "person_name";
    private static final String EXIST_IN_REMOTE_SERVER = "exist_in_remote_server";

```

Figure 4.1 Code segment for database connection establishment in mobile application

```

public DbHandler(@Nullable Context context) { super(context, DB_NAME, factory: null, VERSION); }

@Override
public void onCreate(SQLiteDatabase sqLiteDatabase) {
    String TABLE_CREATE_QUERY_PERSON = "CREATE TABLE " + TABLE_PERSON + " (" + PERSON_ID + " TEXT," + EXIST_IN_REMOTE_SERVER + " TEXT" + ")";
    sqLiteDatabase.execSQL(TABLE_CREATE_QUERY_PERSON);

    String TABLE_CREATE_QUERY_ADDRESS = "CREATE TABLE " + TABLE_ADDRESS + " (" + ADDRESS_ID + " TEXT," + ADDRESS_LINE2 + " TEXT," + LOCATION_DESCRIPTION + " TEXT," + EXIST_IN_REMOTE_SERVER + " TEXT" + ")";
    sqLiteDatabase.execSQL(TABLE_CREATE_QUERY_ADDRESS);

    String TABLE_CREATE_QUERY_MRC = "CREATE TABLE " + TABLE_MRC + " (" + IDENTIFIER + " TEXT," + RELEASE_TYPE + " TEXT," + RUN_NAME + " TEXT," + PERSON_ID + " INTEGER," + EXIST_IN_REMOTE_SERVER + " TEXT" + ")";
    sqLiteDatabase.execSQL(TABLE_CREATE_QUERY_MRC);

    String TABLE_CREATE_QUERY_MRC_RELEASE = "CREATE TABLE " + TABLE_MRC_RELEASE + " (" + IDENTIFIER + " TEXT," + RELEASE_TIME + " TEXT," + RELEASE_STATUS + " TEXT," + IDENTIFIER + " TEXT" + ")";
    sqLiteDatabase.execSQL(TABLE_CREATE_QUERY_MRC_RELEASE);

    String TABLE_CREATE_QUERY_BG_TRAP = "CREATE TABLE " + TABLE_BG_TRAP + " (" + BG_TRAP_ID + " TEXT," + BG_POSITION + " TEXT," + RUN_NAME + " TEXT," + PERSON_ID + " INTEGER," + EXIST_IN_REMOTE_SERVER + " TEXT" + ")";
    sqLiteDatabase.execSQL(TABLE_CREATE_QUERY_BG_TRAP);

```

Figure 4.2 Code segment for table creation in mobile application

```

public void insertDataPerson(PersonModel personObj) {
    SQLiteDatabase sqLiteDatabase = getWritableDatabase();
    ContentValues contentValues = new ContentValues();
    contentValues.put(PERSON_NAME, personObj.getPerson_name());
    contentValues.put(EXIST_IN_REMOTE_SERVER, personObj.getExist_in_remote_server());
    // save to table
    sqLiteDatabase.insert(TABLE_PERSON, nullColumnHack: null, contentValues);
    // close connection
    sqLiteDatabase.close();
}

```

Figure 4.3 Code segment for data insertion to tables in mobile application

```

public List<PersonModel> getAllPersons() {
    List<PersonModel> persons = new ArrayList();
    SQLiteDatabase db = getReadableDatabase();
    String query = "select * from " + TABLE_PERSON + ";";
    Cursor cursor = db.rawQuery(query, selectionArgs: null);
    if (cursor.moveToFirst()) {
        do {
            PersonModel person = new PersonModel();
            person.setPerson_id(cursor.getInt( columnIndex: 0));
            person.setPerson_name(cursor.getString( columnIndex: 1));
            person.setExist_in_remote_server(cursor.getString( columnIndex: 2));

            persons.add(person);
        } while (cursor.moveToNext());
    }
    db.close();
    return persons;
}

```

Figure 4.4 Code segment for querying data from tables in mobile application

```

public int updateSinglePerson(PersonModel personInstance) {
    SQLiteDatabase sqLiteDatabase = getWritableDatabase();
    ContentValues contentValues = new ContentValues();
    contentValues.put(PERSON_NAME, personInstance.getPerson_name());
    int status = sqLiteDatabase.update(TABLE_PERSON, contentValues, whereClause: "person_id=?",
    | new String[]{String.valueOf(personInstance.getPerson_id())});
    return status;
}

```

Figure 4.5 Code segment for updating a table in mobile application

Model Classes

Following figure 4.6, figure 4.7 and figure 4.8 represent a sample model class. Those model classes are used to store data received from application or database. Also, those data can be sent to the database or application. Those model classes consists of attributes, constructor methods, getter methods and setter methods.

```
public class BgTrapModel {
    public String bg_trap_id;
    public String trap_status;
    public String position;
    public String run_name;
    public int person_id;
    public int address_id;
    public String exist_in_remote_server;
```

Figure 4.6 Attributes of model class of mobile application

```
public BgTrapModel(){
}

public BgTrapModel(String bg_trap_id, String trap_status, String position, String run_name,
    int person_id, int address_id, String exist_in_remote_server) {
    this.bg_trap_id = bg_trap_id;
    this.trap_status = trap_status;
    this.position = position;
    this.run_name = run_name;
    this.person_id = person_id;
    this.address_id = address_id;
    this.exist_in_remote_server = exist_in_remote_server;
}
```

Figure 4.7 Constructors of model class of mobile application

```

public void setBg_trap_id(String bg_trap_id) { this.bg_trap_id = bg_trap_id; }

public void setTrap_status(String trap_status) { this.trap_status = trap_status; }

public void setPosition(String position) { this.position = position; }

public void setRun_name(String run_name) { this.run_name = run_name; }

public void setPerson_id(int person_id) { this.person_id = person_id; }

public void setAddress_id(int address_id) { this.address_id = address_id; }

public void setExist_in_remote_server(String exist_in_remote_server) {
    this.exist_in_remote_server = exist_in_remote_server;
}

public String getBg_trap_id() { return bg_trap_id; }

public String getTrap_status() { return trap_status; }

public String getPosition() { return position; }

public String getRun_name() { return run_name; }

public int getPerson_id() { return person_id; }

public int getAddress_id() { return address_id; }

public String getExist_in_remote_server() { return exist_in_remote_server; }

```

Figure 4.8 Getter and setter methods of model class in mobile application

Controller classes

Controller classes are used to control the views (user interfaces) and control the data flow of the application. Following figure 4.9 represents a sample controller class in the proposed mobile application.

```

public class AddMrcMainActivity extends AppCompatActivity {
    final Context context = this;
    EditText txtEdit_TrapId;

    @Override
} protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.add_mrc_main);
    String bg_id = getIntent().getStringExtra( name: "TrapId");
    txtEdit_TrapId = (EditText) findViewById(R.id.editTextTrapId);
    txtEdit_TrapId.setText(bg_id);
}

}

public void goAdditionalMrc(View pView) {
    Intent intent = new Intent(context, AddMrcAdditionalActivity.class);
    startActivity(intent);
}
}
}

```

Figure 4.9 Controller class of mobile application

Code Segments of Web Application

Model Class

Model classes are used to interact with the database. Following code segment in figure 4.10 represents the Maps_Model model class which is the model class spatial data module. That figure represents a function such as saveRecords(\$data). saveRecords(\$data) function is used to save data to map_layer table in MySQL database. Try and Catch clauses are used for exception handling in the code segment.

```

<?php
class Maps_model extends CI_Model
{
    function saveRecords($data)
    {
        try {
            $this->db->insert('map_layer', $data);
            return true;
        }
        catch(Exception $e)
        {
            return false;
        }
    }
}

```

Figure 4.10 Maps_model model class in web application

Controller Class

Following figure 4.11 represents a code segment of goOviPerformanceReportView() function of ReportController class. That function is used to load the OVI performance report. In this function, values of two form inputs such as “from_date” and “to_date” were assigned to an array called to \$data. Also, those values were assigned to a session for further usage. Also, those values were assigned to an array called \$result. Then, displayOviPerformanceReport() function of Report_model model class is called by passing \$data array as a parameter and the return value of those function is assigned to \$result array. Finally, ovi_performance_view screen is loaded by passing \$result array as a parameter.

```

public function goOviPerformanceReportView()
{
    $data['from_date']=$this->input->post('from-date');
    $data['to_date']=$this->input->post('to-date');
    $this -> session -> set_userdata("report_from_date",$data['from_date']);
    $this -> session -> set_userdata("report_to_date",$data['to_date']);
    $result['from_date']=$this->input->post('from-date');
    $result['to_date']=$this->input->post('to-date');
    $result['data'] = $this->Report_model->displayOviPerformanceReport($data);
    $this->load->view( view: 'ovi_performance_view',$result);
}

```

Figure 4.11 goOviPerformanceReportView() function of ReportController class

4.7 Summary

At the beginning of the chapter, implementation environment has been described including minimum hardware and software configurations for the implementation environment. Then, development tools, technologies, reused components, major code segments of mobile application and web application have been described.

Chapter 05: Testing and Evaluation

5.1 Introduction

It is expected to evaluate the developed system focusing on both functional and non-functional requirements. This chapter includes test cases, test results and user evaluation.

5.2 Testing Overview

Software testing can be identified as a process of verifying and validating that the system is bug free, meets all functional and non-functional requirements of the client and meets the technical requirements specified in the design specification. Software testing categorized into two parts such as validation and verification. “Software verification refers to set of activities that ensure that software correctly implemented a specific function. Software validation refers to set of activities that ensures that the software is built traceable to the customer requirements” (GeeksforGeeks, 2019). Field planning and monitoring system for World Mosquito Program, Sri Lankan site has been tested focusing on both functional and non-functional requirements. Test cases are designed to cover all the requirements and finally those test cases are executed using test data. Different defects are identified during the testing process and those identified defects are fixed and res-tested during the testing process. Black box testing has been used to test both functional testing and non-functional testing.

5.3 Test Cases and Test Results

5.3.1 Functional Test Cases and Results

Table 5.1 Test case for add a new MRC location in mobile application

Test case id	T001	Test case name	Add a new MRC location in mobile application.
Module name	Field activities management	Test description	Verifying a user can add a new MRC location in mobile application.
Prerequisites	User should log into the mobile application as a field assistant	Test status	PASSED
Test step		Test data	Expected Result
1. Click on “MRC Field Activities” option in main menu screen.			1. User should be navigated to the MRC field activities screen.
2. Click on “Create a new place” link.			2. User should be navigated to add new MRC wizard.

3.1 Click on next button without filling all required fields.		3.1 Error message should be displayed. Reference: figure 5.1
3.2 Fill all required fields while entering an existing MRC identifier which is already existing in mobile application.	MRC Identifier: cmc-32	3.2 Error message should be displayed. Reference: figure 5.2
3.3 Fill all required fields with valid inputs and click on next button.	MRC Identifier: Cmc-2 MRC Status: Proposed Respond Name: Kamal Silva Location Coordinates: 6.9799959, 79.8727244	3.3 User should be navigated to the next screen of the wizard.
4.0 Fill all required fields and click on done button.	Phone: 0112345678 Address Line1: No.20, Modara Rd Address Line2: Mattakkuliya Location Description: house with white color gate	4.0 New MRC location should be successfully added to the system and user should be navigated to the MRC field activities screen while displaying a success message. Reference: figure 5.3

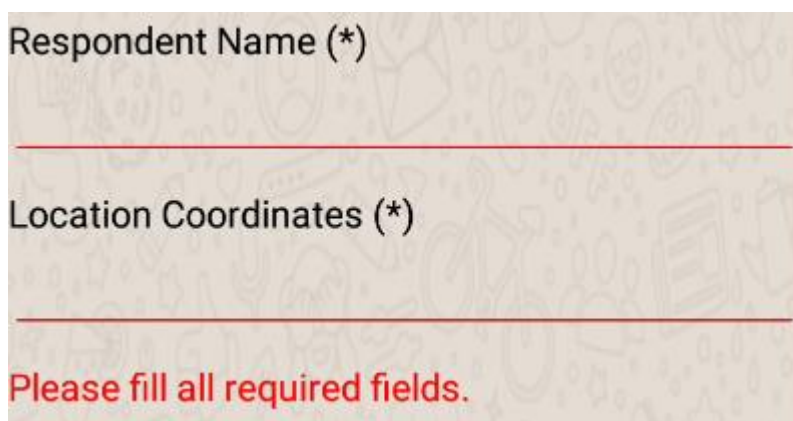


Figure 5.1 Error message for empty required fields

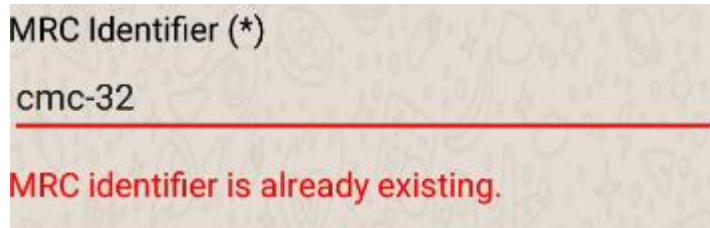


Figure 5.2 Error message for already existing MRC identifier

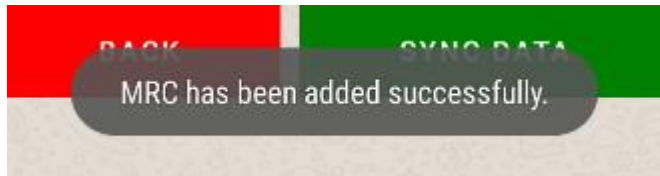


Figure 5.3 Success message after adding new MRC location

Table 5.2 Test case for add a new field incident in mobile application

Test case id	T002	Test case name	Add a new field incident in mobile application.
Module name	Field incidents management	Test description	Verifying a user can add a new field incident in mobile application.
Prerequisites	User should log into the mobile application as a field assistant	Test status	PASSED
Test step		Test data	Expected Result
1. Click on “Field Incidents” option in main menu screen.			1. User should be navigated to the “contact details” screen of add new field incident wizard.
2. 1 Enter valid inputs to fields except email and click on next button.	Email: gihan.gmail		2.1 Error message should be displayed. Reference: figure 5.4
2.2 Enter valid inputs to fields except phone and click on next button.	Phone: 07412		2.2 Error message should be displayed. Reference: figure 5.5
2.3 Enter valid inputs to fields and click on next button.	Community member: gayan gamage Email: gayan@gmail.com Phone: 0773706004		2.3 User should be navigated to the incident details screen of the wizard.
3.0 Enter valid inputs to the fields and click on next button.	Incident Type: Community Enquiry		3.0 User should be navigated to the incident location details screen of the wizard.

	Incident Priority: Medium Description: increment of no of mosquitos Date of Incident: 2021-02-11 Time of Incident: 14:10	
4.0 Enter valid inputs to the fields and click on submit button.	Address: No.5, Kannangara Rd, Kotahena	4.0 Field incident should be successfully added to the system and a success message should be displayed. Reference: figure 5.6



Figure 5.4 Error message for invalid email address

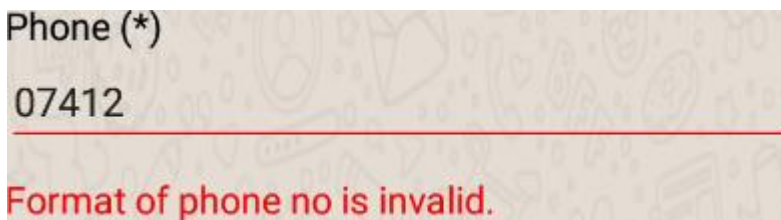


Figure 5.5 Error message for invalid phone no

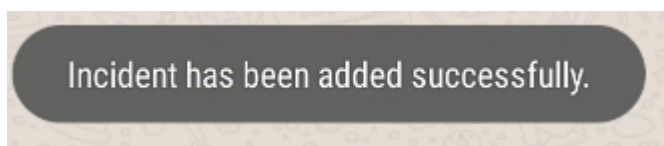


Figure 5.6 Success message after adding new field incident

Table 5.3 Test case for add a new MRC service in mobile application

Test case id	T003	Test case name	Add a new MRC service in mobile application.
Module name	Field activities management	Test description	Verifying a user can add a new MRC service in mobile application.

Prerequisites	User should log into the mobile application as a field assistant	Test status	PASSED
Test step		Test data	Expected Result
1.0 Click on “MRC Field Activities” option in main menu screen.			1.0 User should be navigated into “MRC field activities” screen.
2.0 Select a MRC service run from the field runs list.	Field run: MRC_Nug_1		2.0 All the MRC locations related to selected field run should be displayed.
3.0 select a MRC location.	MRC Location: Nug-12		3.0 User should be navigated to the “Add new MRC service” screen.
4.1 Click on next button by entering existing service id.	Service Id: ser-01		4.1 Error message should be displayed. Reference: figure 5.7
4.2 Click on next button without selecting service status.			4.2 Error message should be displayed. Reference: figure 5.8
4.3 Enter service status, valid service id and click on next button.	Service status: Serviced, Service id: ser-04		4.3 User should be navigated to the next screen of wizard.
5.0 Click on done button.			5.0 MRC service should be successfully added to the system, success message should be displayed and user should be navigated to the MRC activities screen. Reference: figure 5.9

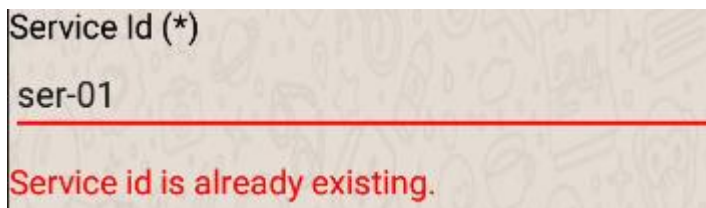


Figure 5.7 Error message for existing service id

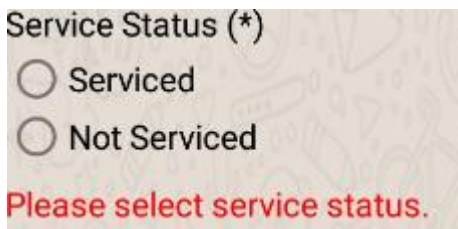


Figure 5.8 Error message for empty service status

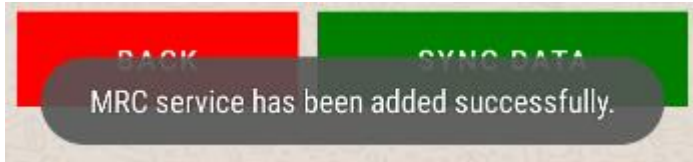


Figure 5.9 Error message after adding MRC service

Table 5.4 Test case for add new MRC release in mobile application

Test case id	T004	Test case name	Add a new MRC release in mobile application.
Module name	Field activities management	Test description	Verifying a user can add a new MRC release in mobile application.
Prerequisites	User should log into the mobile application as a field assistant	Test status	PASSED
Test step		Test data	Expected Result
1.0 Click on “MRC Field Activities” option in main menu screen.			1.0 User should be navigated into “MRC field activities” screen.
2.0 Select a MRC release run from the field runs list.	Field run: MRC_CMC_10		2.0 All the MRC locations related to selected field run should be displayed.
3.0 select a MRC location.	MRC Location: Cmc-22		3.0 User should be navigated to the “Add new MRC release” screen.
4.1 Click on next button by entering existing release id.	Release id: Rel-01		4.1 Error message should be displayed. Reference: figure 5.10
4.2 Click on next button without selecting release status.			4.2 Error message should be displayed. Reference: figure 5.11
4.3 Enter release status, release id and click on next button.	Release status: released Release id: Rel-02		4.3 User should be navigated to the next screen of wizard.
5.0 Click on done button.			5.0 MRC release should be successfully added to the system, success message should be displayed and user should be navigated to the MRC activities screen. Reference: figure 5.12

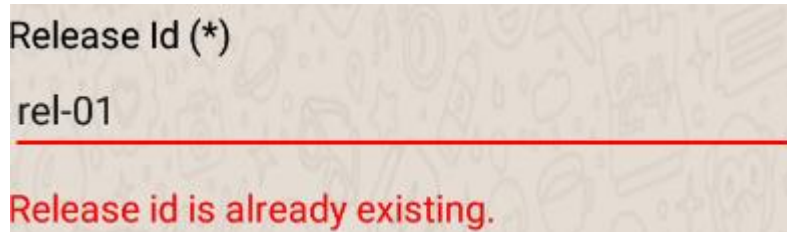


Figure 5.10 Error message for existing release id.



Figure 5.11 Error message for empty release status

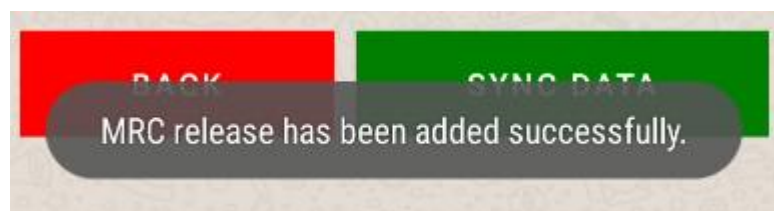


Figure 5.12 Success message after adding MRC release

Table 5.5 Test case for add a new BG collection in mobile application

Test case id	T005	Test case name	Add a new BG collection in mobile application.
Module name	Field activities management	Test description	Verifying a user can add a new BG collection in mobile application.
Prerequisites	User should log into the mobile application as a field assistant	Test status	PASSED
Test step		Test data	Expected Result
1.0 Click on “BG Field Activities” option in main menu screen.			1.0 User should be navigated into “BG field activities” screen.
2.0 Select a BG collection run from the field runs list.	Field run:BG_Nug_4		2.0 All the BG locations related to selected field run should be displayed.
3.0 select a BG location.	BG Location: Nug-11		3.0 User should be navigated to the “Add new BG collection” screen.

4.1 Click on next button by entering existing collection id.	Collection Id: col-087	4.1 Error message should be displayed. Reference: figure 5.13
4.2 Click on next button without entering collection status.		4.2 Error message should be displayed. Reference: figure 5.14
4.3 Enter collection status, collection id and click on next button.	Collection status: Collected, Collection id: Col-099	4.3 User should be navigated to the next screen of wizard.
5.0 Click on done button.		5.0 BG collection should be successfully added to the system, success message should be displayed and user should be navigated to the BG activities screen. Reference: figure 5.15



Figure 5.13 Error message for existing BG collection id

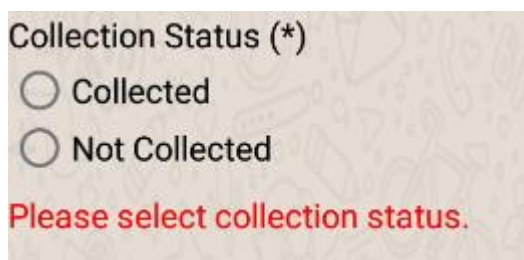


Figure 5.14 Error message for empty collection status

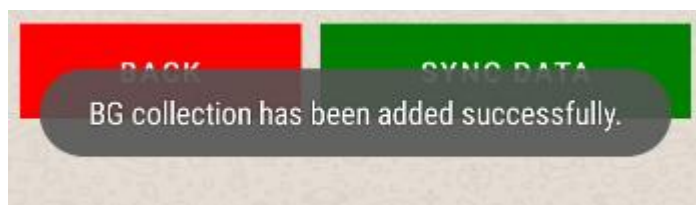


Figure 5.15 Success message after adding BG collection

Table 5.6 Test case for view MRC field locations in map view

Test case id	T006	Test case name	View MRC Field locations in map view
Module name	Field activities management	Test description	Verifying a user can view MRC locations in the map view of mobile application.
Prerequisites	User should log into the mobile application as a field assistant, location service of mobile device should be turned on.	Test status	PASSED
Test step		Test data	Expected Result
1.0 Click on “MRC Field Activities” option in main menu screen.			1.0 User should be navigated into “MRC field activities” screen.
2.0 Select a field run from the field runs list.	Field run:MRC_Nug_4		2.0 All the MRC locations related to selected field run should be displayed.
3.0 Click on map icon			3.0 User should be navigated to the map view and all MRC location related to the selected field run should be visualized on the map. Reference: figure 5.16
4.0 Click on a MRC location displayed on the map.	Nug-92		4.0 User should be navigated to the “Update MRC Collection” wizard and all information of selected MRC location should be displayed in the wizard.

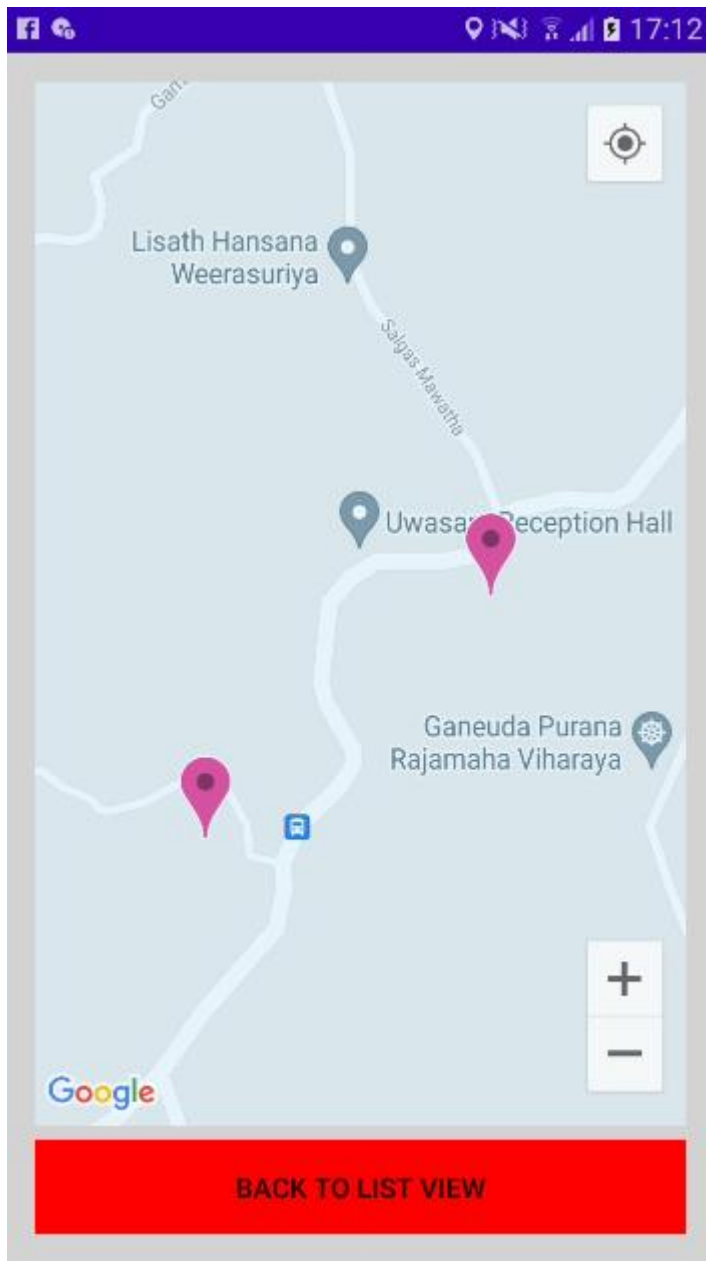


Figure 5.16 Map view of MRC locations

Table 5.7 Test case for add new map layer in web application

Test case id	T007	Test case name	Add new map layer in web application.
Module name	Spatial data management	Test description	Verifying a user can add a new map layer in web application.
Prerequisites	User should log into the web application as a GIS officer.	Test status	PASSED
Test step		Test data	Expected Result

1.0 Click on “Spatial Data” tab in main menu.		1.0 User should be navigated to the spatial list screen.
2.0 Click on “Add new map layer” button.		2.0 User should be navigated to the “Add new map layer” screen.
3.1 Click on save button without entering inputs to all required fields.		3.1 Error message should be displayed. Reference: figure 5.17
3.2 Enter inputs for all required fields while entering map name which is already existing in the system.	Name: CMC DI grids	3.2 Error message should be displayed. Reference: figure 5.18
3.3 Enter valid inputs for all required fields and then click on save button.	Name: Nugegoda GND Map Description: GND Map of Nugegoda MOH Area GeoJson Content: GeoJson content of map	3.3 New map layer details should be added to the system successfully and a success message should be displayed. Then user should be directed to the “spatial list” screen. Reference: figure 5.19

Name:(*)

Enter Map Name

Please enter name

Description:(*)

Enter Map Description

Please enter description

Add GeoJson Content Here:(*)

Please enter GeoJson content

Please fill all required fields.

Figure 5.17 Error message for empty required fields

Name:(*)

CMC D1 grids

Map name is already existing in system

Figure 5.18 Error message for already existing map name

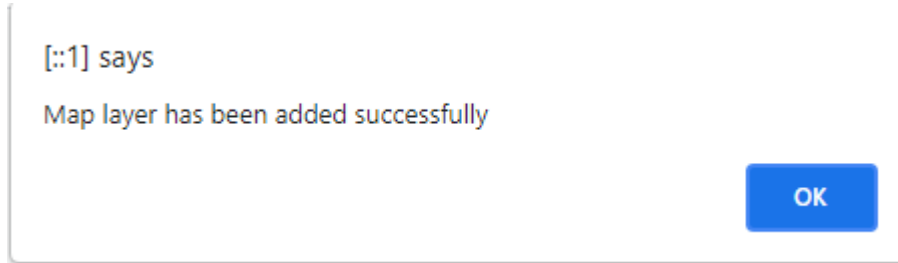


Figure 5.19 Success message after adding new map layer to system

Table 5.8 Test case for add new OVI service run in web application

Test case id	T008	Test case name	Add new OVI service run in web application
Module name	Field planning management	Test description	Verifying a user add new OVI service run in web application
Prerequisites	User should log into the web application as a GIS officer.	Test status	PASSED
Test step		Test data	Expected Result
1.0 Click on "Field Planning" tab in main menu.			1.0 User should be navigated to the "Field Planning" screen.
2.0 Click on "Add New Field Run" button.			2.0 User should be navigated to the "Add New Field Run" screen.
3.1 Enter inputs for all required fields including file other than .csv type file to file uploading field and click on save button.	ans.txt		3.1 Error message should be displayed. Reference: figure 5.20
3.2 Enter a run name which is already existing in system and enter valid inputs for other fields and click on save button.	Name: Mrc_CMC_1		3.2 Error message should be displayed. Reference: figure 5.21
3.3 Enter some invalid OVI trap Ids which are not available in system, in to .csv file and enter valid inputs to all other fields and click on save button.	Invalid OVI Traps: cmc-1200, cmc-1400		3.3 Error message should be displayed. Reference: figure 5.22

3.4 Enter some OVI trap Ids which have invalid status such as set, in to .csv file and enter valid inputs to all other fields and click on save button.	OVI traps with invalid status: cmc-4, cmc-5	3.4 Error message should be displayed. Reference: figure 5.23
3.5 Enter some OVI trap Ids which are already in other OVI field runs with pending status, in to .csv file and enter valid inputs to all other fields and click on save button.	OVI traps: cmc-101, cmc-102	3.5 Error message should be displayed. Reference: figure 5.24
3.6 Enter valid inputs to all fields and click on save button.	Name: Mrc_CMC_10, Type: MRC Service Run, OVI traps in .csv file (cmc-17, cmc-18, cmc-19, cmc-23), Run Description: tenth OVI run in cmc, Date: 2021-01-20, Time: 09:30	3.6 Field run details should be added to the system successfully and a success message box should be displayed. User should be navigated to the “Field Planning” screen. Reference: figure 5.25

Upload .csv File Here(*):

Please upload only .csv files.

Figure 5.20 Error message for uploading a file of invalid file type

Name(*):

Run name is already existing in system.

Figure 5.21 Error message for existing run name

Field Run Error Log

Identifier	Error Description
cmc-1200	invalid identifiers
cmc-1400	invalid identifiers

Figure 5.22 Error message for invalid trap ids which are not existing in system

Field Run Error Log

Identifier	Error Description
cmc-4	invalid status
cmc-5	invalid status

Figure 5.23 Error message for trap ids which have invalid status

Field Run Error Log

Identifier	Error Description
cmc-101	Trap is already in a pending run
cmc-102	Trap is already in a pending run

Figure 5.24 Error message for trap ids which are already in a pending field run

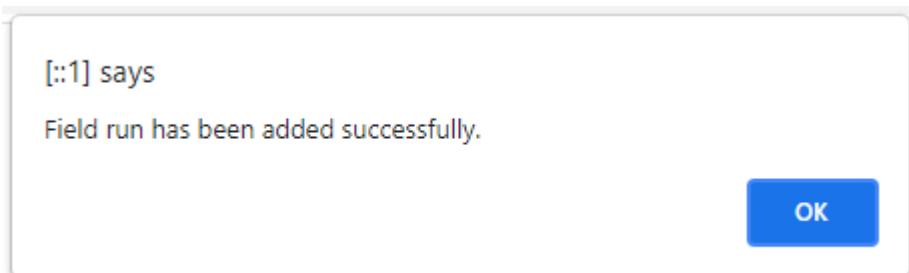


Figure 5.25 Success message after adding OVI service field run

Table 5.9 Test case for add new mosquito diagnostic in web application

Test case id	T009	Test case name	Add new mosquito diagnostic in web application
Module name	Mosquito diagnostic management	Test description	Verifying a user can add new mosquito diagnostic in web application.
Prerequisites	User should log into the web application as a diagnostic officer.	Test status	PASSED
Test step	Test data	Expected Result	
1.0 Click on “Mosquito Diagnostics” tab in main menu.		1.0 User should be navigated to the “Mosquito Diagnostics” screen.	
2.0 Click on “Add New Mosquito Diagnostic” button.		2.0 User should be navigated to the “Add New Mosquito Diagnostic” screen including valid BG collection ids.	
3.1 Enter inputs for all required fields incorporating an existing diagnostic id and click on save button.	Diagnostic Id: Dia-12	3.1 Error message should be displayed. Reference: figure 5.26	
3.2 Enter a date to identified date which is before than the collection date of selected collection id and enter valid inputs to all other fields and click in save button.	Identified date: 2020-08-12	3.2 Error message should be displayed. Reference: figure 5.27	
3.3 Enter valid inputs for all fields and click on save button.		3.3 Diagnostic details should be added to system successfully and a success message should be displayed. User should be navigated to the “Mosquito Diagnostics” screen. Reference: figure 5.28	

Diagnostic Id(*):

Dia-12

Diagnostic id is already existing

Figure 5.26 Error message for already existing diagnostic id

Identified Date(*):

12/08/2020

Identified date should be greater than collected date 14/08/2020

Figure 5.27 Error message for invalid identified date

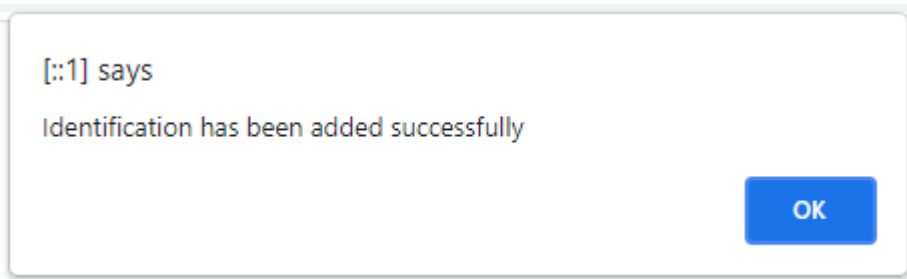


Figure 5.28 Success message after adding identification details

Table 5.10 Test case for add new mosquito screening result in web application

Test case id	T010	Test case name	Add new mosquito screening result in web application
Module name	Mosquito screening management	Test description	Verifying a user can add mosquito screening result in web application.
Prerequisites	User should log into the web application as a molecular biologist.	Test status	PASSED
Test step		Test data	Expected Result
1.0 Click on “Screening Results” tab in main menu.			1.0 User should be navigated to the “Screening Results” screen.
2.0 Click on “Add New Screening Result” button.			2.0 User should be navigated to the “Add New Screening Result” screen including valid diagnostic ids.
3.1 Enter inputs for all required fields incorporating an existing screening id and click on save button.	Screening Id: Scr-14		3.1 Error message should be displayed. Reference: figure 5.29
3.2 Enter a date to screening date which is before than the identified date of selected diagnostic id and enter valid inputs to all other fields and click in save button.	Screened Date: 2020-11-10		3.2 Error message should be displayed. Reference: figure 5.30
3.3 Enter valid inputs for all fields and click on save button.			3.3 Screening details should be added to system successfully and a success message should be displayed. User should be navigated to the “Mosquito Screening” screen. Reference: figure 5.31

Screening Id(*):

Screening id is already existing in system.

Figure 5.29 Error message for existing screening id.

Screened Date(*):

Screened date should be greater than identified date 12/11/2020

Figure 5.30 Error message for invalid screened date

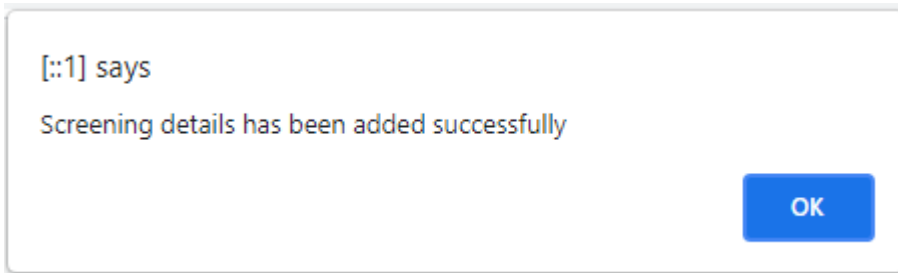


Figure 5.31 Success message after adding screening result

Table 5.11 Test case for add new bio-banking export in web application

Test case id	T011	Test case name	Add new bio-banking export in web application
Module name	Bio-banking exports management	Test description	Verifying a user can add bio-banking export in web application.
Prerequisites	User should log into the web application as a molecular biologist.	Test status	PASSED
Test step		Test data	Expected Result
1.0 Click on “Bio-Banking Exports” tab in main menu.			1.0 User should be navigated to the “Bio Banking Exports” screen.
2.0 Click on “Add New Bio Banking Export” button.			2.0 User should be navigated to the “Add New Bio Banking Export” screen including valid OVI collection ids.
3.1 Enter inputs for all required fields incorporating an existing export id and click on save button.	Export Id: Ex-19		3.1 Error message should be displayed. Reference: figure 5.32
3.2 Enter a date to export date which is before than the collected date of selected OVI collection id and enter valid inputs to all other fields and click in save button.	Export Date: 2021-01-12		3.2 Error message should be displayed. Reference: figure 5.33

3.3 Enter valid inputs for all fields and click on save button.		3.3 Export details should be added to system successfully and a success message should be displayed. User should be navigated to the “Bio Banking Exports” screen. Reference: figure 5.34
---	--	--

Export Id(*):

Export id is already existing

Figure 5.32 Error message for already existing export id

Export Date(*):

Export date should be greater than collection date 14/01/2021

Figure 5.33 Error message for invalid export date

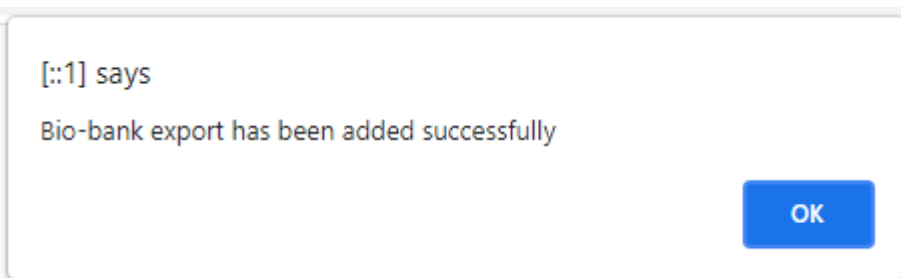


Figure 5.34 Success message after adding bio-bank export

Table 5.12 Test case for generate OVI performance report in web application

Test case id	T012	Test case name	Generate OVI performance report in web application
Module name	Report generation	Test description	Verifying a user can generate OVI performance report in web application.
Prerequisites	User should log into the web application as a GIS officer.	Test status	PASSED
Test step	Test data	Expected Result	
1.0 Click on “Reports” tab in main menu.		1.0 User should be navigated to the “Reports Menu” screen.	

2.0 Click on “OVI Performance Report” link.		2.0 User should be navigated to the “OVI Performance Report Generation” screen.
3.1 Enter invalid date ranges to date fields and click on “Generate Report” button.	Date From: 2021-03-05 Date To: 2021-01-20	3.1 Error message should be displayed. Reference: figure 5.35
3.2 Enter date ranges which does not have data for report generation.	Date From: 2024-01-01 Date To: 2025-01-01	3.2 Error message should be displayed. Reference: figure 5.36
3.3 Enter valid date ranges to fields and click on “Generate Report” button.	Date From: 2020-01-01 Date To: 2021-09-05	3.3 OVI performance report should be generated and displayed to the user. Reference: figure 5.37
4.0 Click on “Print Report” button.		4.0 Print window should be displayed to user.

Date From: Date To:

Invalid date range selected. Please select valid date range.

Figure 5.35 Error message for invalid date range

Date From: Date To:

System does not have enough data for selected date range to generate report.

Figure 5.36 Error message for not having enough data for selected date range

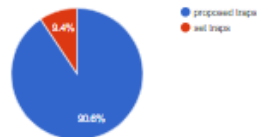
OVI Performance Report

Date From: 2020-01-01

Date To: 2021-09-05

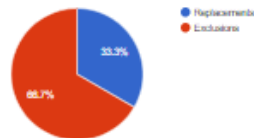
New Ovi Trap Locations

	Count	Percentage
Proposed traps	222	90.612244800
Set traps	23	9.387755100



Ovi Exclusions and Replacements

	Count	Percentage
Replacements	1	33.333333000
Exclusions	2	66.666666000



Print Report

Figure 5.37 Generated OVI performance report

Table 5.13 Test case for display locations of field run in GIS dashboard in web application

Test case id	T013	Test case name	Display locations of relevant BG field run in GIS dashboard in web application
Module name	Dashboard	Test description	Verifying a user can display locations of relevant BG field run in GIS dashboard in web application.
Prerequisites	User should log into the web application as a GIS officer or field coordinator.	Test status	PASSED
Test step		Test data	Expected Result
1.0 Click on “Dashboards” tab in main menu.			1.0 User should be navigated to the “Dashboards” screen.

2.0 Click on “GIS Dashboard” link.		2.0 User should be navigated to the “GIS Dashboard” screen.
3.0 Enter valid inputs to the “Field Location Type” field.	Field Location Type: BG Locations	3.0 Field runs related to selected location type should be loaded in to the “Run Name” field.
4.0 Select a run name from “Run Name” field.	Run Name: BG_CMC_1	4.0 Field locations related to the selected field run should be displayed on the map. Reference: figure 5.38

GIS Dashboard

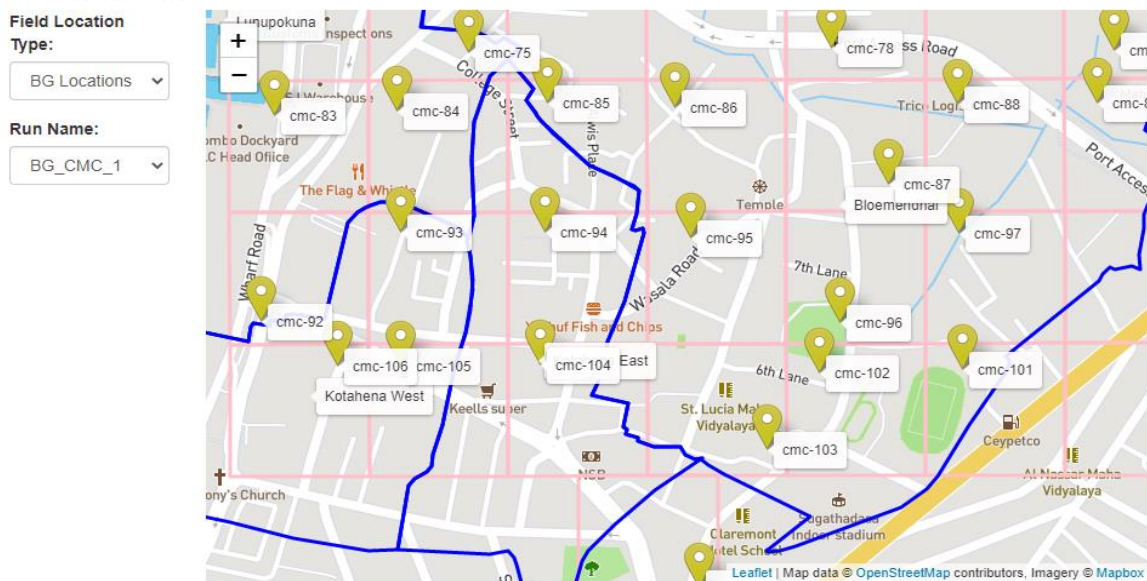


Figure 5.38 Location visualization in GIS dashboard

Please refer Appendix E – Test Cases and Test Results for the rest of functional test cases and results.

5.3.2 Non-Functional Test Cases and Results

Non-functional testing is defined as a type of software testing to check non-functional requirements such as performance, usability and security of the software application. Non-functional testing is done with non-functional parameters which are not focused in functional testing (guru99, 2021).

Security Testing

“The purpose of Security Tests is to identify all possible loopholes and weaknesses of the software system which might result in a loss of data, information, revenue and unauthorized

access of outsiders of the organization” (guru99, 2021). Following table 5.14 represents the test case for saving encrypted passwords.

Table 5.14 Test case for save encrypted passwords

Test case id	T025	Test case name	Passwords of mobile application logins are stored encrypted in database
Module name	Users Management	Test description	Verifying a user can display locations of relevant BG field run in GIS dashboard in web application.
Prerequisites	User should log into the web application as administrator.	Test status	PASSED
Test step		Test data	Expected Result
1.0 Click on “Users Management” tab in main menu.			1.0 User should be navigated to the “Users Management Menu” screen.
2.0 Click on “Mobile application logins” link.			2.0 User should be navigated to the “Mobile Application Logins” screen.
3.0 Click on “Add New Mobile Login” link.			3.0 User should be navigated to the “Add New Mobile Login” screen.
4.0 Enter valid inputs to fields and click on save button.	Username: Fieldofficer, Password: fieldman		4.0 Login details should be added to the system successfully and password should be stored encrypted in the database.

Usability Testing

“Usability testing mainly focuses the user friendliness of the system, flexibility of the system and ability of the system to meet its objectives” (guru99, 2021). Below table 5.15 represents ad hoc test scenarios that have been executed.

Table 5.15 Test cases for usability testing

Test ID	Description	Expected Result	Test Status
T026	Verifying the user can search map layers by typing map layer name in web application.	User should be able to search map layers by typing map layer name in web application.	PASSED

T027	Verify the user can search field incidents by typing member name, address, date and time in web application.	User should be able to search field incidents by typing member name, address, date and time in web application.	PASSED
T028	Verify the user can display diagnostic records with a pagination in web application.	User should be able to display diagnostic records with a pagination in web application.	PASSED
T029	Verify the user can display screening records with a pagination in web application.	User should be able to display screening records with a pagination in web application.	PASSED

5.4 User Evaluation

User evaluation is the final stage of the testing process. This focuses on how well users can learn and use the application to perform the actions/ functionalities. It also refers to how satisfied users are with the application developed. Questionnaires were used to gather feedbacks from the users about implemented system. Both open-ended and closed-ended questions are included in the questionnaire targeting all the user roles that will be associated with the application. Below are some major questions that are included in the questionnaire. This will be given to a selected number of end users to gather information regarding the field planning and monitoring system of World Mosquito Program, Sri Lankan Site.

- Please select the category that best describes your role:
 Field Assistant Field Coordinator GIS Officer Diagnostic Officer
 Molecular Biologist
- How easy is the application to use?
 Poor Not Bad Good Excellent
- The content of the application clearly represented?
 Poor Not Bad Good Excellent

4. The application navigation is easy to understand:

Poor Not Bad Good Excellent

5. What is your preferred mapping technology used in the system?

Google Maps Open Street Maps

6. As a field assistant, how mobile application is useful in data collection in the field activities?

Poor Not Bad Good Excellent

7. What do you like least/most about the application?

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

8. Which features of the application are most important to you?

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

9. What is the most important feature you think that should be added to the application?

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

10. Overall, how would you rate our site?

Poor Not Bad Good Excellent

Also, following suggestions were provided by the users.

- Design more attractive user interfaces for mobile application.
- Include help option in web application.
- Integrate scientific calculation tools for mosquito diagnostics management module and mosquito screening management module in web application.

5.5 Summary

At the beginning of the chapter, testing overview has been described. Then, functional test cases and test results have been described. Then, non-functional test cases and results have been described including security testing and usability testing. Finally, user evaluation has been represented including sample questionnaires.

Chapter 06 – Conclusion

6.1 Introduction

World Mosquito Program, Sri Lankan site is one of the leading companies in Sri Lanka involved in protecting people from mosquito borne diseases. Because of the drawbacks and weaknesses of the existing semi-automated field planning and monitoring system, the company has taken a decision to implement a new field planning and monitoring system to cover its all field planning and monitoring tasks. That was the basis for proposed field planning and monitoring system to be implemented.

The developed system consists of a mobile application (for Android operating system) and a web application. Mobile application is used as a data collection tool inside the field in field activities. Web application is used as a data management and data administration tool for both field planning activities and field monitoring activities. The system is allowed to manipulate information related to spatial data administration, field planning, field activities, field incidents, mosquito diagnostics, screening results, bio-banking exports. Also, separate administration module was implemented to manage users. Also, system produces meaningful managerial reports for the top-level and middle-level managers to take managerial decisions. Also, user interfaces of both mobile and web application was designed user friendly to increase the usability of the system. Also, both client-side validations and server-side validations with meaningful error messages are done to prevent user and system from mistakes and errors. Moreover, security and integrity of the system were considered in the implementation. Also, developed web application allows employees to access the system without depending on the location and time. Finally, it can be concluded that developed field planning and monitoring system has a high positive impact on current field planning and monitoring operations of World Mosquito Program, Sri Lankan site.

6.2 Objectives Achieved

The developed system has achieved all the objectives of the project. One of the objectives of the project is to handle data collections related to all kind of field activities through the mobile application. That objective has been achieved by implementing functions related to OVI field activities, BG field activities and MRC field activities in mobile application. One of the objectives of the project is to record field incident details through the mobile application and manage those details through the web application. That objective has been achieved by implementing real time field incident insertion facility in mobile application and implementing

field incidents module in web application with relevant functions such as add field incidents, update field incidents, delete field incidents and view field incidents. One of the objectives of the project is to administrate field planning activities through the web application. That objective has been achieved by implementing field planning module in web application including adding new field runs and viewing existing field runs. One of the objectives of the project is to handle report generation. That objective has been achieved by implementing report generation module in web application by facilitating to generate different reports such as OVI performance report, BG performance report, MRC performance report, mosquito diagnostic report and mosquito screening report. One of the objectives of the project is to record and maintain bio-banking exports through the web application. That objective has been achieved by implementing bio-banking exports module in web application by facilitating to add, delete, update and view bio-banking exports. One of the objectives of the project is to record and maintain mosquito identifications through web application. That objective has been achieved by implementing mosquito diagnostics module in web application by facilitating to add, delete, update and view mosquito diagnostics. One of the objectives of the project is to record and maintain screening results through web application. That objective has been achieved by implementing mosquito screening results module in web application by facilitating to add, delete, update and view mosquito screening results. One of the objectives of the project is to manage spatial data related to field planning through web application. That objective has been achieved by implementing spatial data management module in web application by facilitating to add, delete, update and display different map layers to the system. Therefore, it has been proved that all the objectives of the project have been achieved.

6.3 Problems Encountered

At the beginning of the project, the main problem was the unfamiliarity with the problem domain due to its definitions, unique business process and lack of biology related scientific knowledge to understand scientific processes of the organization. That problem was resolved with the help of the chief project manager of World Mosquito Program, Sri Lankan site. Furthermore, in requirements gathering process, scheduled meetings with the employees were rescheduled several times because of very busy lifestyles of the employees. Also, initially, employees did not like to provide official documents and information because of the confidentiality. But after some period, that situation was changes and employees agreed to provide their official documents and information for reviewing.

6.4 Lessons Learnt

The knowledge and experience taken from this project is very valuable. For every task of this project, it was essential to learn every phases of software development life cycle. Also, managing a kind of project on a given timeline is not a simple task. In this project, I learnt technologies such as CSS, JavaScript, Bootstrap, Codeigniter, PHP, MySQL, Git and GIS. Also, I learnt MVC architecture and RUP process model in this project. Also, I learnt project management principles in this project. In this project, I was able to apply theories and principles learnt from MIT degree program to real world problem solving scenario.

6.5 Future Improvements

After successfully implementing and deploying the entire system and based on the positive customer feedbacks received from the different stakeholders of the system, it is confirmed that system consists all functions and features expected by client. Also, there are some future improvements which can add more value to developed system. The followings are some future enhancements which are to be included into developed field planning and monitoring system for World Mosquito Program, Sri Lankan Site.

- Build an official web site for World Mosquito Program, Sri Lankan Site to provide awareness on the project operations for local citizens of Sri Lanka.
- Integrate an optical scan feature to extract data from collection stickers, releasing stickers, diagnostic stickers and screening stickers.
- Enhance security features of the web application using digital certificates and firewalls.
- Incorporate GIS based location based client-side validations to both mobile and web application.
- Incorporate server-side GIS services to the web application.

References

- Dimagi. (2020) *About Commcare*. [Online] Available at: <https://www.dimagi.com/commcare/>. [Accessed 12 2021].
- FusionCharts. (2020) *Build beautiful web & mobile dashboards*. [Online] Available at: <https://www.fusioncharts.com/>. [Accessed 09 April 2021].
- GeeksForGeeks. (2019) *Software Testing Basics*. [Online] Available at: <https://www.geeksforgeeks.org/software-testing-basics/>. [Accessed 10 June 2021].
- GeeksForGeeks. (2020) *Functional vs Non Functional Requirements*. [Online] Available at: <https://www.geeksforgeeks.org/functional-vs-non-functional-requirements/>. [Accessed 28 January 2021].
- GeeksforGeeks. (2020) *RUP and its Phases*. [Online] Available at: <https://www.geeksforgeeks.org/rup-and-its-phases/>. [Accessed 10 March 2021].
- GeeksForGeeks. (2020) *Types of Feasibility Study in Software Project Development*. [Online] Available at: <https://www.geeksforgeeks.org/types-of-feasibility-study-in-software-project-development>. [Accessed 28 January 2021].
- GitHub. (2020) *Dompdf*. [Online] Available at: <https://github.com/dompdf/dompdf>. [Accessed 08 April 2021].
- GitHub. (2021) *GitHub Guides*. [Online] Available at: <https://guides.github.com/>. [Accessed 4 August 2021].
- Google. (2020) *About My Maps*. [Online] Available at: <https://www.google.com/maps/about/mymaps/>. [Accessed 1 February 2021].
- Google. (2020) *Maps SDK for Android Overview*. [Online] Available at: <https://developers.google.com/maps/documentation/android-sdk/overview>. [Accessed 12 April 2021].
- Google. (2021) *Using Google Charts*. [Online] Available at: <https://developers.google.com/chart/interactive/docs>. [Accessed 4 August 2021].
- Guru99. (2021) *What is Non Functional Testing?*. [Online] Available at: <https://www.guru99.com/non-functional-testing.html>. [Accessed 11 June 2021].
- Guru99. (2021) *What is Security Testing?*. [Online] Available at: <https://www.guru99.com/what-is-security-testing.html>. [Accessed 11 June 2021].
- Guru99. (2021) *What is Usability Testing?*. [Online] Available at: <https://www.guru99.com/usability-testing-tutorial.html>. [Accessed 11 June 2021].
- Kobo. (2021) *Features of KoBo Toolbox*. [Online] Available at: <https://www.kobotoolbox.org/>. [Accessed 2 February 2021].
- Leaflet. (2020) *An open source JavaScript library for mobile-friendly interactive maps*. [Online] Available at: <https://leafletjs.com/>. [Accessed 10 April 2021].

Marston, T. (2018) *The Model-View-Controller (MVC) Design Pattern for PHP*. [Online] Available at: <http://www.tonymarston.net/php-mysql/model-view-controller.html> . [Accessed 07 March 2021].

Object Management Group. (2021) *Unified Modelling Language*. [Online] Available at: <https://www.omg.org/spec/UML/About-UML/#spec-versions-formal> . [Accessed 08 March 2021].

SitePoint. (2020) *Sending Emails with PHPMailer*. [Online] Available at: <https://www.sitepoint.com/sending-emails-php-phpmailer/>. [Accessed 08 April 2021].

Summerfield, J. (2020) *Mobile Website vs. Mobile App: Which is Best for Your Organization?*. [Online] Available at: <https://www.hswsolutions.com/services/mobile-web-development/mobile-website-vs-apps/> . [Accessed 09 March 2021].

TutorialsPoint. (2021) *Object Oriented Approach*. [Online] Available at: https://www.tutorialspoint.com/system_analysis_and_design/system_analysis_and_design_object_oriented_approach.htm . [Accessed 08 March 2021].

TutorialsPoint. (2021) *SDLC-Overview*. [Online] Available at: https://www.tutorialspoint.com/sdlc/sdlc_overview.htm. [Accessed 09 March 2021].

World Mosquito Program. (2021) *About us*. [Online] Available at: <https://www.worldmosquitoprogram.org/en/work/about-us>. [Accessed 28 January 2021].

Appendix A – System Manual

This documentation provides guidelines for the installation process of both mobile application and web application of field planning and monitoring system for World Mosquito Program, Sri Lankan Site. Also, this document includes the minimum hardware and software requirements required for the installation process. This documentation is divided into two sections such as system manual of the mobile application and system manual of web application.

System Manual of Mobile Application

Hardware Requirements

Table A.1 Hardware requirements for mobile application

Hardware	Minimum Requirements
Processor	Quad-core 1.5 GHz Cortex-A7
Memory	1.5GB RAM
Free storage space	100MB of free storage space
Internet	Minimum 512 kbps mobile broadband connection

Software Requirements

Table A.2 Software requirements for mobile application

Software	Minimum Requirements
Operating system	Android 6.0.2 Marshmallow

Installation and Launching

- 1) Access Google shared drive of World Mosquito Program and download signed .apk file through a mobile device.
- 2) Open the downloaded .apk file and run the installation process.
- 3) After the installation process, go to settings->applications and enable location permission for the installed application.
- 4) Go to the main menu and open the installed application to launch it.

System Manual of Web Application

Hardware Requirements

Table A.3 Hardware requirements for web application

Hardware	Minimum Requirements
Processor	Intel Core i3 processor 1.80 GHz, 2.40 GHz
Memory	4GB or higher
Display	1024x768 of resolution
Hard disc space	1GB minimum or higher
Printer	Laser printer
Internet	100Mbps home broadband connection

Software Requirements

Table A.4 Software requirements for web application

Software	Minimum Requirements
Operating system	Microsoft Windows 8/10, Ubuntu 9.0
Bundle packages	XAMPP server 1.8.3
Web browser	Google Chrome/Mozilla Firefox/Internet Explorer

Installation and Launching

- 1) Access Google shared drive of World Mosquito Program and download **CoreDataWeb** folder through a computer.
- 2) Copy and paste the downloaded folder into following paths.
 - In Windows: - C:\xampp\htdocs
 - In Ubuntu: - /opt/lampp/htdocs
- 3) Access Google shared drive of World Mosquito Program and download **Backup.sql** file through a computer.
- 4) Open PhpMyAdmin and import **Backup.sql** file.
- 5) Open the web browser and visit [http://\[::1\]/CoreDataWeb/CoreDataWeb/index.php/MainMenuController](http://[::1]/CoreDataWeb/CoreDataWeb/index.php/MainMenuController) to launch the web application.

Appendix B – Design Documentation

Use-case Diagrams and Use-Case Narratives for Module Design

Mosquito Diagnostic Management

In this module, entomology officer should be able to add, delete, update and display mosquito diagnostic results to the system through web application. Following figure B.1 is the use-case diagram of mosquito diagnostic management module.

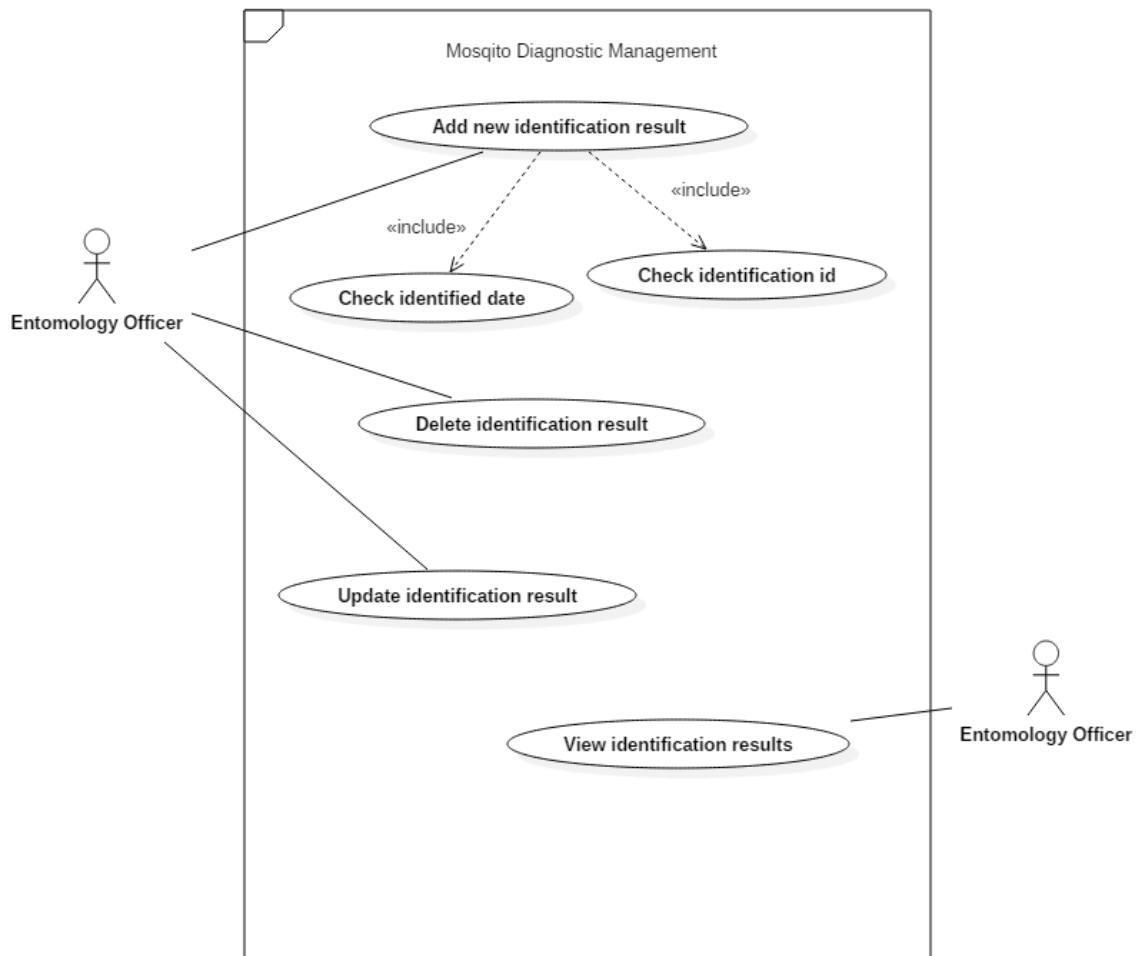


Figure B.1 Use-case diagram of mosquito diagnostic management module

Following table B.1 is the use-case narrative of add new identification use case.

Table B.1 Use-case narrative of add new identification result

Use case name	Add new identification result
Actors	Entomology officer

Description	Describes how entomology officer adds a new identification result to the system through the web application.
Normal sequence of events	<ol style="list-style-type: none"> 1) Entomology officer fills the form and submits it. 2) System checks identification id. 3) System checks the identified date. 4) New identification result is added to the system.
Alternative sequence of events	<ol style="list-style-type: none"> 3) If identification id is already existing in system then, user will be notified an error message. 4) If identified date is invalid then, user will be notified an error message.
Pre-conditions	<ul style="list-style-type: none"> • Identification id should not be existing in the system. • Identified date should be a valid date.
Post-conditions	<ul style="list-style-type: none"> • Relevant tables of the remote database should be updated.
Assumptions	<ul style="list-style-type: none"> • User may enter an identification id which is existing in the system. • User may enter an invalid identified date.

Following table B.2 is the use-case narrative of delete identification result use case.

Table B.2 Use-case narrative of delete identification result

Use case name	Delete identification result
Actors	Entomology officer
Description	Describes how entomology officer deletes an identification result from the system through web application.

Normal sequence of events	<ol style="list-style-type: none"> 1) Entomology officer views specific identification result. 2) Entomology officer clicks delete button. 3) Identification result is deleted from the system.
Post-conditions	<ul style="list-style-type: none"> • Relevant tables of the remote database should be updated.

Screening Results Management

In this module, molecular biologist should be able to add, delete, update and view screening results in the system through the web application. Following figure B.2 is the use-case diagram of screening results management module.

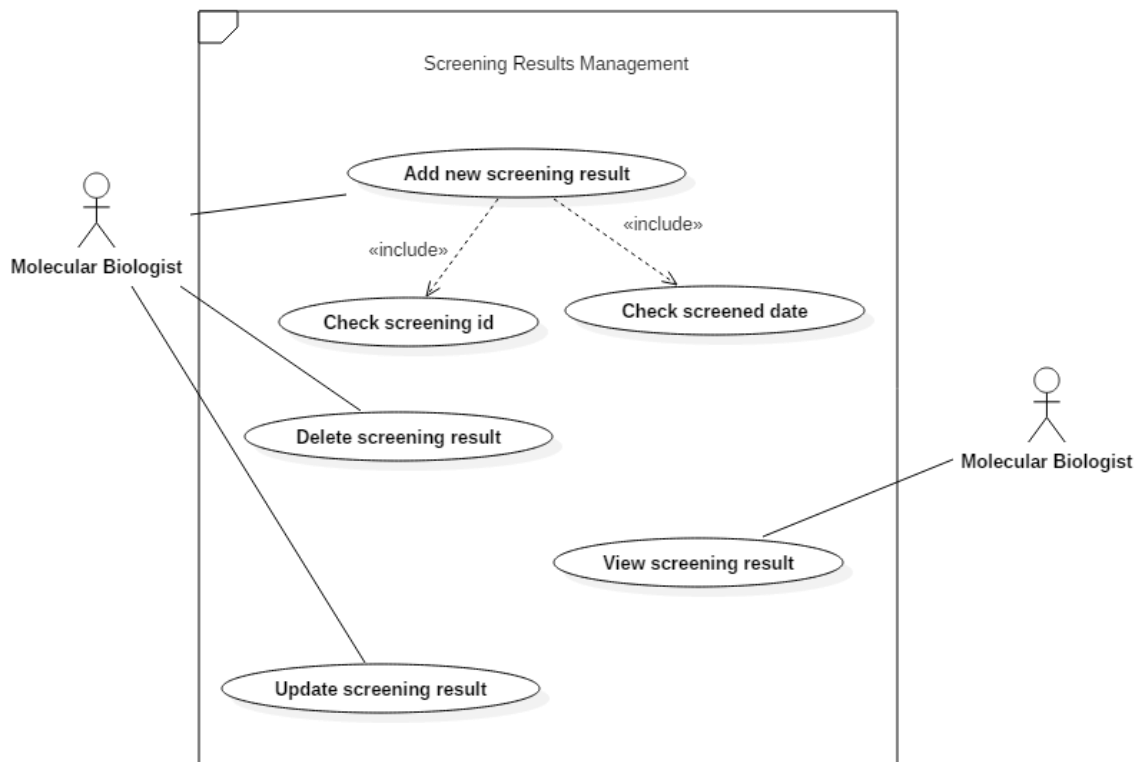


Figure B.2 Use-case diagram of screening results management module

Following table B.3 is the use-case narrative of add new screening result of screening results management module.

Table B.3 Use-case narrative of add new screening result

Use case name	Add new screening result
Actors	Molecular biologist
Description	Describes how molecular biologist add a new screening result to the system through web application.
Normal sequence of events	<ol style="list-style-type: none"> 1) Molecular biologist fills the form and submits it. 2) System checks screening id. 3) System checks screened date. 4) New screening result is added to system.
Alternative sequence of events	<ol style="list-style-type: none"> 3) If screening id is already existing in system then, user will be notified an error message. 4) If screened date is invalid then, user will be notified an error message.
Pre-conditions	<ul style="list-style-type: none"> • Screening id should not be existing in the system. • Screened date should be a valid date.
Post-conditions	<ul style="list-style-type: none"> • Relevant tables of the remote database should be updated.
Assumptions	<ul style="list-style-type: none"> • User may enter a screening id which is already existing in the system. • User may enter invalid screened date.

Bio-banking Export Management Module

In this module, molecular biologist should be able to add and update bio-banking export details in the system through the web application. Following figure B.3 is the use-case diagram of bio-banking export management module.

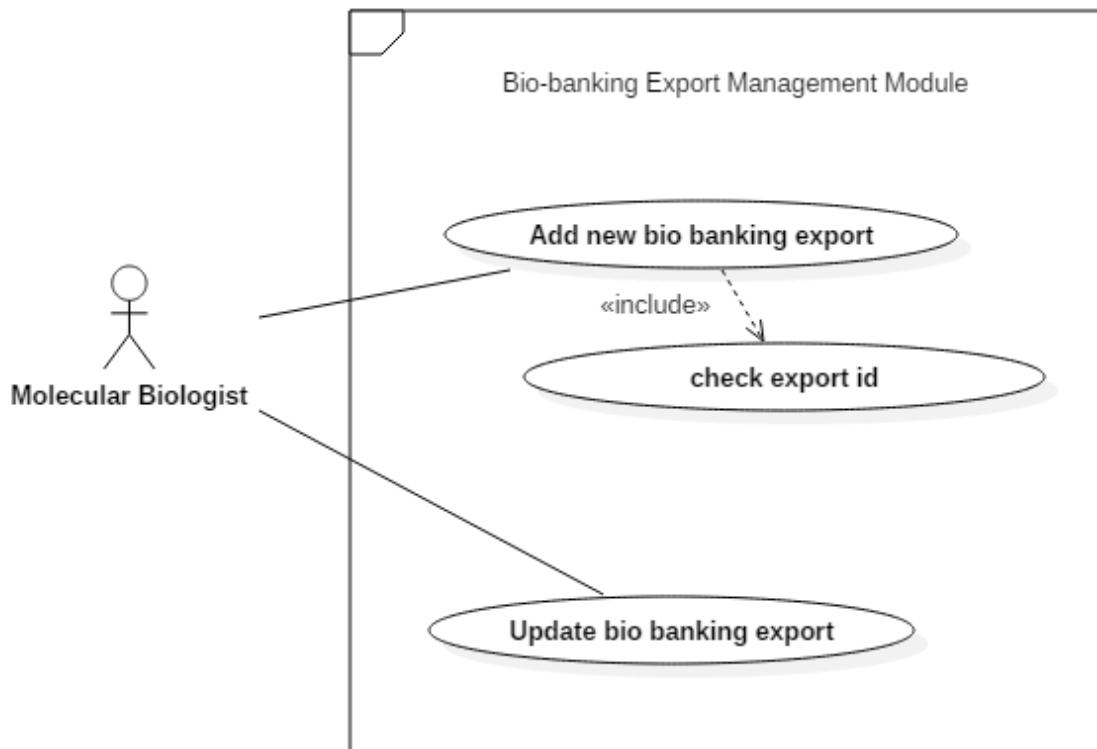


Figure B.3 Use-case diagram of bio-banking export management module

Following table B.4 is the use-case narrative of add new bio banking export use-case.

Table B.4 Use-case narrative of add new bio banking export

Use case name	Add new bio banking export
Actors	Molecular biologist
Description	Describes how molecular biologist add a new bio bank export to the system through web application.
Normal sequence of events	<ol style="list-style-type: none"> 1) Molecular biologist fills the form and submits it. 2) System checks export id. 3) New bio-banking export is added to the system.
Alternative sequence of events	3) If export id is already existing in system then, user will be notified an error message.
Pre-conditions	<ul style="list-style-type: none"> • Export id should not be existing in system.

Post-conditions	<ul style="list-style-type: none"> Relevant tables of the remote database should be updated.
Assumptions	<ul style="list-style-type: none"> User may enter already existing export id.

Report Generation Module

In this module, GIS officer should be able to generate several reports such as performance report OVI traps, performance report of BG traps and performance report of MRC releases. Also, entomology officer should be able to generate analysis report of mosquito diagnostics. Also, molecular biologist should be able to generate analysis report of mosquito screening. Following figure B.4 is the use-case diagram of report generation module.

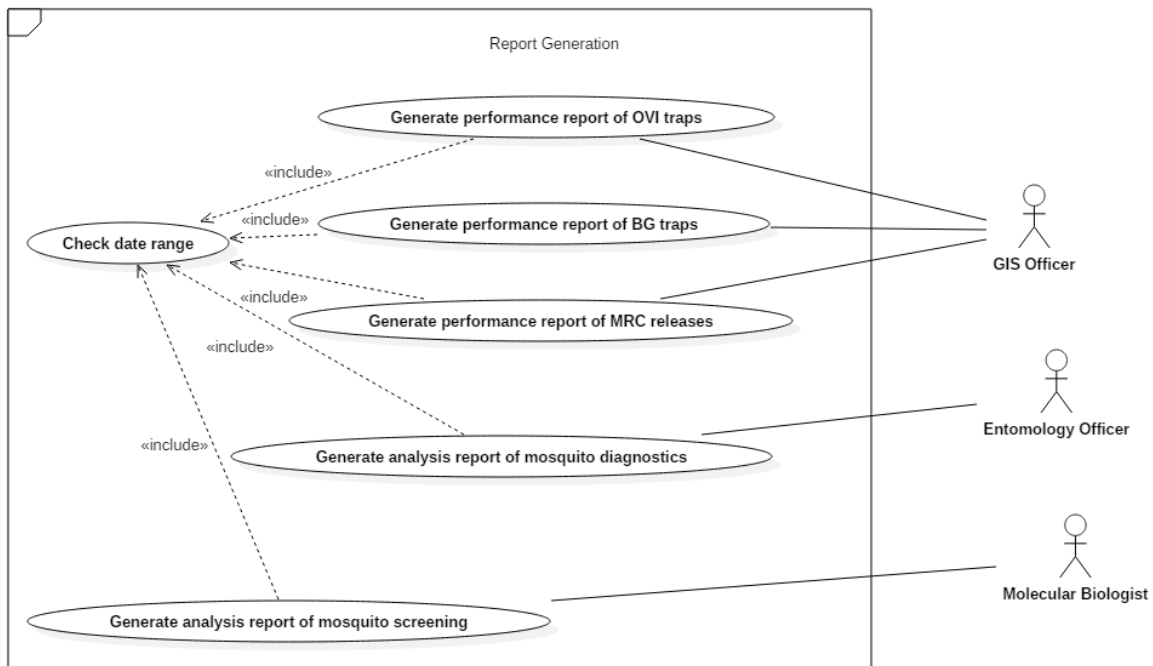


Figure B.4 Use-case diagram of report generation module

Following table B.5 is the use-case narrative of generate performance report of BG traps use-case.

Table B.5 Use-case narrative of generate performance report of BG traps

Use case name	Generate performance report of BG traps
Actors	GIS officer

Description	Describes how GIS officer generates performance report of BG traps through the web application.
Normal sequence of events	<ol style="list-style-type: none"> 1) GIS officer fills the form and submits it. 2) System checks the date range. 3) BG performance report is generated.
Alternative sequence of events	<ol style="list-style-type: none"> 3) If the date range is not valid then, user will be notified an error message.
Pre-conditions	<ul style="list-style-type: none"> • User should enter valid date range.
Assumptions	<ul style="list-style-type: none"> • User may enter invalid date ranges.

Appendix C – User Manual

This document represents how to use main functions of both mobile application and web application. This document is divided into two sections such as user manual for mobile application and user manual for web application.

User Manual for Mobile Application

Following figure C.1 represents the login screen of the mobile application. Users who have authorization, can access the application by providing valid username and password. If user provides valid username and password then, user will be redirected to the figure C.2 menu screen.



Figure C.1 Login screen of mobile application



WMP Field

Menu

MRC Field Activities

BG Field Activities

OVI Field Activities

Field Incidents

Figure C.2 Menu screen of mobile application

Through the menu screen, users can navigate to forms of different field activities and field incident forms.

User Manual for Web Application

Before access the web application, users should log into a valid Gmail address to which administrator has granted privileges to access the system. If user log into a correct Gmail account and access the URL of the web application then, user will be navigated to the main menu screen. Following figure C.3 represents the main menu screen of administrator's view. That screen consists links to all available modules of the system.

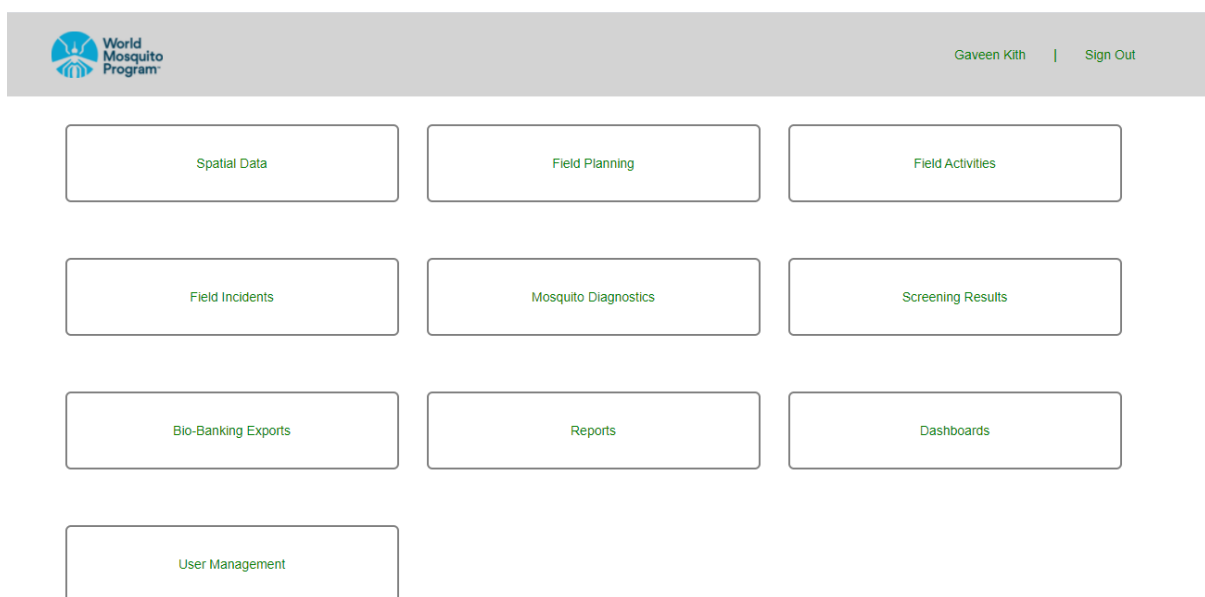


Figure C.3 Main menu screen of web application

Appendix D – MIS Reports

Mosquito Diagnostic Report

Following figure D.1 represents the template of mosquito diagnostic report. That report provides an overall summarized idea about mosquito diagnostics for a given period of time. In this report, identified total male mosquitos, identified total female mosquitos, total of both male and female mosquitos are represented statistically for different mosquito categories. Also, those information are represented graphically using a bar column chart.

Mosquito Diagnostic Report

Date From: 2020-01-01 Date To: 2021-10-01

Mosquito Category	Male Total	Female Total	Total
Aedes aegypti	392	334	726
Anopheles	258	241	499
Culex	268	241	509

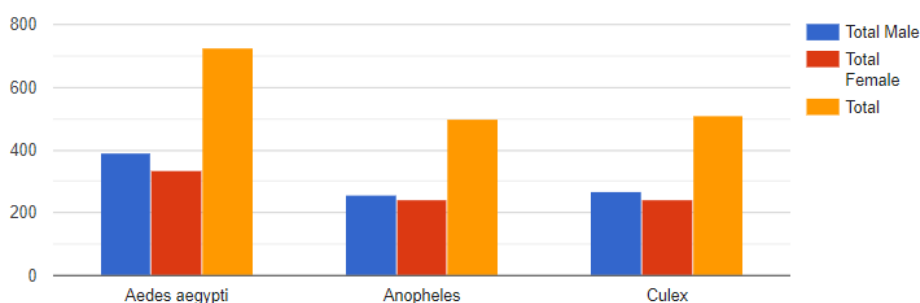


Figure D.1 Mosquito diagnostic report

BG Performance Report

Following figure D.2 represents the template of BG performance report. That report is used to get an overall summarized idea about field activities related to BG traps for a given time period. Date range used to represent generate the report is displayed in top left corner and top right corner of the report template. Information which are provided by that report, is mainly divided for two parts such as new BG trap locations and BG exclusions and replacements. In new BG trap locations part, count and percentage of count of both proposed traps and set traps are visualized statistically. Also, those information are visualized graphically using a pie chart. In BG exclusions and replacements part, count and percentage of both replacements and exclusions are visualized statistically. Also, those information are visualized graphically using a pie chart.

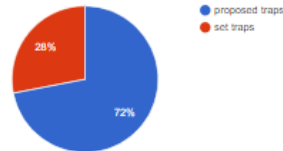
BG Performance Report

Date From: 2020-01-01

Date To: 2021-10-28

New BG Trap Locations

	Count	Percentage
Proposed traps	77	71.962616800
Set traps	30	28.037383100



BG Exclusions and Replacements

	Count	Percentage
Replacements	1	50
Exclusions	1	50

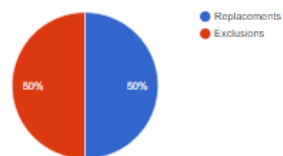


Figure D.2 BG performance report

MRC Performance Report

Following figure D.3 represents the template of MRC performance report. That report is used to get an overall summarized idea about field activities related to MRC traps for a given time period. Date range used to represent generate the report is displayed in top left corner and top right corner of the report template. Information which are provided by that report, is mainly divided for two parts such as new MRC locations and MRC exclusions and replacements. In new MRC locations part, count and percentage of count of both proposed locations and set locations are visualized statistically. Also, those information are visualized graphically using a pie chart. In MRC exclusions and replacements part, count and percentage of both replacements and exclusions are visualized statistically. Also, those information are visualized graphically using a pie chart.

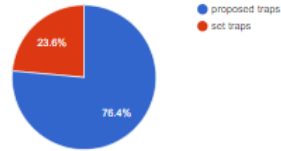
MRC Performance Report

Date From: 2020-01-01

Date To: 2021-09-01

New MRC Locations

	Count	Percentage
Proposed locations	81	76.415094300
Set locations	25	23.584905600



MRC Exclusions and Replacements

	Count	Percentage
Replacements	2	40
Exclusions	3	60

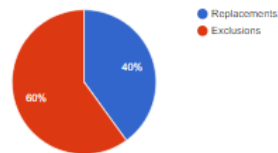


Figure D.3 MRC performance report

Appendix E – Test Cases and Test Results

Functional Test Cases and Results

Table E.1 Test case for add a new BG location in mobile application

Test case id	T014	Test case name	Add a new BG location in mobile application.
Module name	Field activities management	Test description	Verifying a user can add a new BG location in mobile application.
Prerequisites	User should log into the mobile application as a field assistant	Test status	PASSED
Test step		Test data	Expected Result
1. Click on “BG Field Activities” option in main menu screen.			1. User should be navigated to the BG field activities screen.
2. Click on “Create a new place” link.			2. User should be navigated to add new BG wizard.
3.1 Click on next button without filling required fields.			3.1 Error message should be displayed. Reference: figure E.1
3.2 Fill all required fields while entering an existing BG identifier which is already existing in mobile application.	BG Identifier: nug-45		3.2 Error message should be displayed. Reference: figure E.2
3.3 Fill all required fields with valid inputs and click on next button.	BG Identifier: Cmc-5 BG Status: Proposed Respond Name: Sudesh Bandara Location Coordinates: 6.9766959, 79.8927244		3.3 User should be navigated to the next screen of the wizard.
4.0 Fill all required fields and click on done button.	Phone: 0112345978 Address Line1: No.20, Mahawatta Rd Address Line2: Mattakkuliya		4.0 New BG location should be successfully added to the system and user should be navigated to the BG field activities screen while displaying a success message. Reference: figure E.3

	Location Description: house with gren color gate	
--	--	--



Figure E.1 Error message for empty required fields

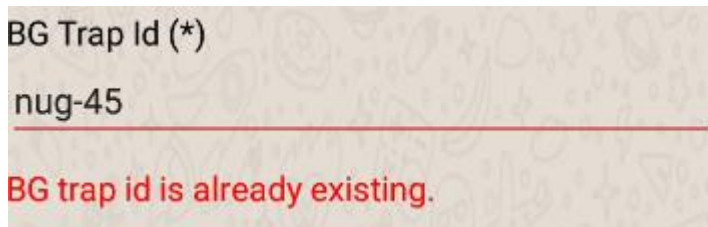


Figure E.2 Error message for existing BG trap id



Figure E.3 Success message after adding BG trap location

Table E.2 Test case for add new OVI location in mobile application

Test case id	T015	Test case name	Add a new OVI location in mobile application.
Module name	Field activities management	Test description	Verifying a user can add a new OVI location in mobile application.
Prerequisites	User should log into the mobile application as a field assistant	Test status	PASSED
Test step		Test data	Expected Result
1. Click on “OVI Field Activities” option in main menu screen.			1. User should be navigated to the OVI field activities screen.

2. Click on “Create a new place” link.		2. User should be navigated to add new OVI wizard.
3.1 Click on next button without filling required fields.		3.1 Error message should be displayed. Reference: figure E.4
3.2 Fill all required fields while entering an existing OVI identifier which is already existing in mobile application.	OVI Identifier: nug-45	3.2 Error message should be displayed. Reference: figure E.5
3.3 Fill all required fields with valid inputs and click on next button.	OVI Identifier: Cmc-10 OVI Status: Proposed Respond Name: Kanchana Perera Location Coordinates: 6.9966959, 79.8927244	3.3 User should be navigated to the next screen of the wizard.
4.0 Fill all required fields and click on done button.	Phone: 0772345978 Address Line1: No.20, Kotahena Rd Address Line2: Modara Location Description: house with pink color door	4.0 New OVI location should be successfully added to the system and user should be navigated to the OVI field activities screen while displaying a success message. Reference: figure E.6



Figure E.4 Error message for empty required fields



Figure E.5 Error message for already existing OVI trap id

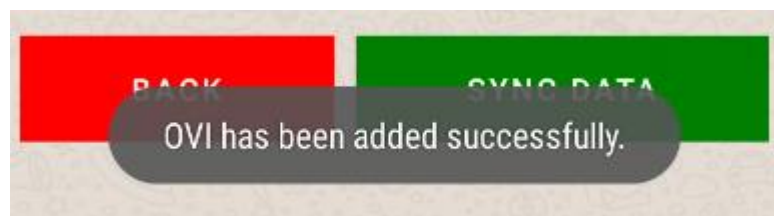


Figure E.6 Success message after adding OVI trap location

Table E.3 Test case for add a new BG service in mobile application

Test case id	T016	Test case name	Add a new BG service in mobile application.
Module name	Field activities management	Test description	Verifying a user can add a new BG service in mobile application.
Prerequisites	User should log into the mobile application as a field assistant	Test status	PASSED
Test step		Test data	Expected Result
1.0 Click on “BG Field Activities” option in main menu screen.			1.0 User should be navigated into “BG field activities” screen.
2.0 Select a BG service run from the field runs list.	Field run:BG_Nug_2		2.0 All the BG locations related to selected field run should be displayed.
3.0 select a BG location.	BG Location: Nug-14		3.0 User should be navigated to the “Add new BG service” screen.
4.1 Click on next button by entering existing service id.	Service Id: ser-02		4.1 Error message should be displayed. Reference: figure E.7
4.2 Click on next button without selecting service id.			4.2 Error message should be displayed. Reference: figure E.8
4.3 Enter service status, service id and click on next button.	Service status: Serviced, Service Id: ser-09		4.3 User should be navigated to the next screen of wizard.
5.0 Click on done button.			5.0 BG service should be successfully added to the system, success message should be displayed and user should be navigated to the BG activities screen.

		Reference: figure E.9
--	--	-----------------------

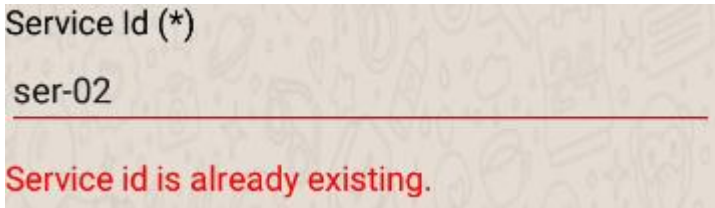


Figure E.7 Error message for existing service id

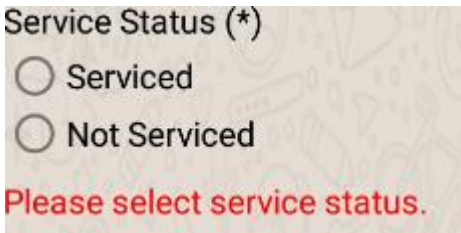


Figure E.8 Error message for empty service status

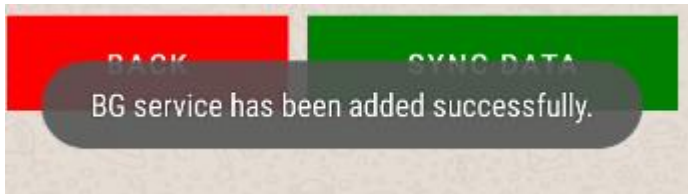


Figure E.9 Success message after adding BG service

Table E.4 Test case for add a new OVI service in mobile application

Test case id	T017	Test case name	Add a new OVI service in mobile application.
Module name	Field activities management	Test description	Verifying a user can add a new OVI service in mobile application.
Prerequisites	User should log into the mobile application as a field assistant	Test status	PASSED
Test step		Test data	Expected Result
1.0 Click on “OVI Field Activities” option in main menu screen.			1.0 User should be navigated into “OVI field activities” screen.
2.0 Select an OVI service run from the field runs list.	Field run:OVI_Nug_12		2.0 All the OVI locations related to selected field run should be displayed.
3.0 select an OVI location.	OVI Location: Nug-14		3.0 User should be navigated to the “Add new OVI service” screen.
4.1 Click on next button by entering existing service id.	Service Id: ser-10		4.1 Error message should be displayed. Reference: figure E.10

4.2 Click on next button without selecting service status.		4.2 Error message should be displayed. Reference: figure E.11
4.3 Enter service status, service id and click on next button.	Service status: Serviced, Service Id: ser-06	4.3 User should be navigated to the next screen of wizard.
5.0 Click on done button.		5.0 OVI service should be successfully added to the system, success message should be displayed and user should be navigated to the OVI activities screen. Reference: figure E.12

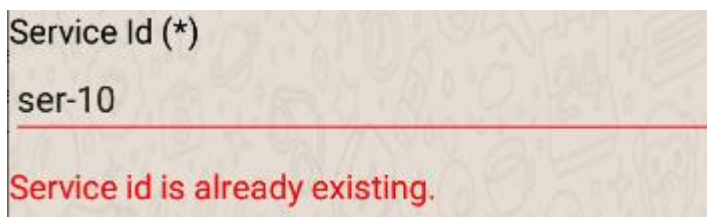


Figure E.10 Error message for existing service id

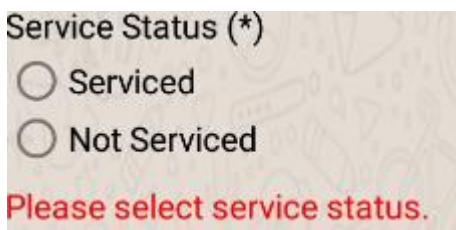


Figure E.11 Error message for empty service status

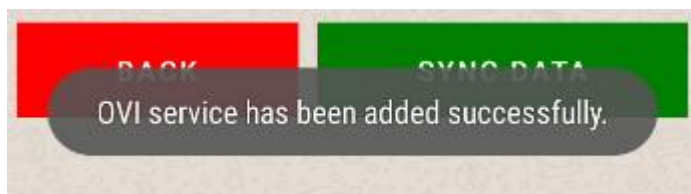


Figure E.12 Success message after adding OVI service

Table E.5 Test case for add a new OVI collection in mobile application

Test case id	T018	Test case name	Adding a new OVI collection in mobile application.
Module name	Field activities management	Test description	Verifying a user can add a new OVI collection in mobile application.
Prerequisites	User should log into the mobile application as a field assistant	Test status	PASSED
Test step		Test data	Expected Result

1.0 Click on “OVI Field Activities” option in main menu screen.		1.0 User should be navigated into “OVI field activities” screen.
2.0 Select an OVI collection run from the field runs list.	Field run:OVI_Nug_7	2.0 All the OVI locations related to selected field run should be displayed.
3.0 select an OVI location.	OVI Location: Nug-17	3.0 User should be navigated to the “Add new OVI collection” screen.
4.1 Click on next button by entering existing collection id.	Collection id: Col-050	4.1 Error message should be displayed. Reference: figure E.13
4.2 Click on next button without selecting collection status.		4.2 Error message should be displayed. Reference: figure E.14
4.3 Enter collection status, collection id and click on next button.	Collection status: Collected, Collection id: Col-054	4.3 User should be navigated to the next screen of wizard.
5.0 Click on done button.		5.0 OVI collection should be successfully added to the system, success message should be displayed and user should be navigated to the OVI activities screen. Reference: figure E.15

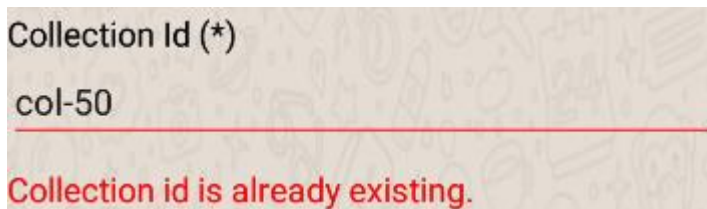


Figure E.13 Error message for existing OVI collection id

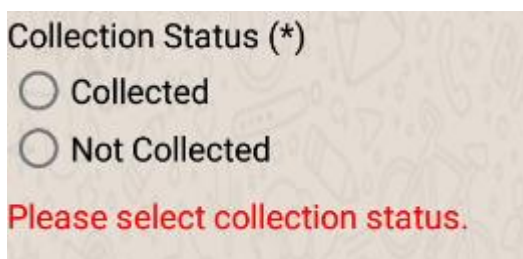


Figure E.14 Error message for empty collection status



Figure E.15 Error message after adding OVI collection

Table E.6 Test case for view current location of mobile device in map view

Test case id	T019	Test case name	View current location of the mobile device in map view.
Module name	Field activities management	Test description	Verifying a user can view the current location of the mobile device in the map view of MRC locations, OVI locations and BG locations.
Prerequisites	User should log into the mobile application as a field assistant, location service of mobile device should be turned on.	Test status	PASSED
Test step		Test data	Expected Result
1.0 Click on “MRC Field Activities” option in main menu screen.			1.0 User should be navigated into “MRC field activities” screen.
2.0 Select a field run from the field runs list.	Field run:MRC_Nug_4		2.0 All the MRC locations related to selected field run should be displayed.
3.0 Click on map icon			3.0 User should be navigated to the map view and all MRC location related to the selected field run should be visualized on the map.
4.0 Click on the current location icon displayed on the map.			4.0 Current location of the device should be visualized on the map using marker.
5.0 Navigate to the map view of OVI locations and click on the current location icon displayed on the map.	Field run: OVI_Nug_2		5.0 Current location of the device should be visualized on the map using marker. Reference: figure E.16
6.0 Navigate to the map view of BG locations and click on the current location icon displayed on the map.	Field run: BG_Nug_4		6.0 Current location of the device should be visualized on the map using marker.

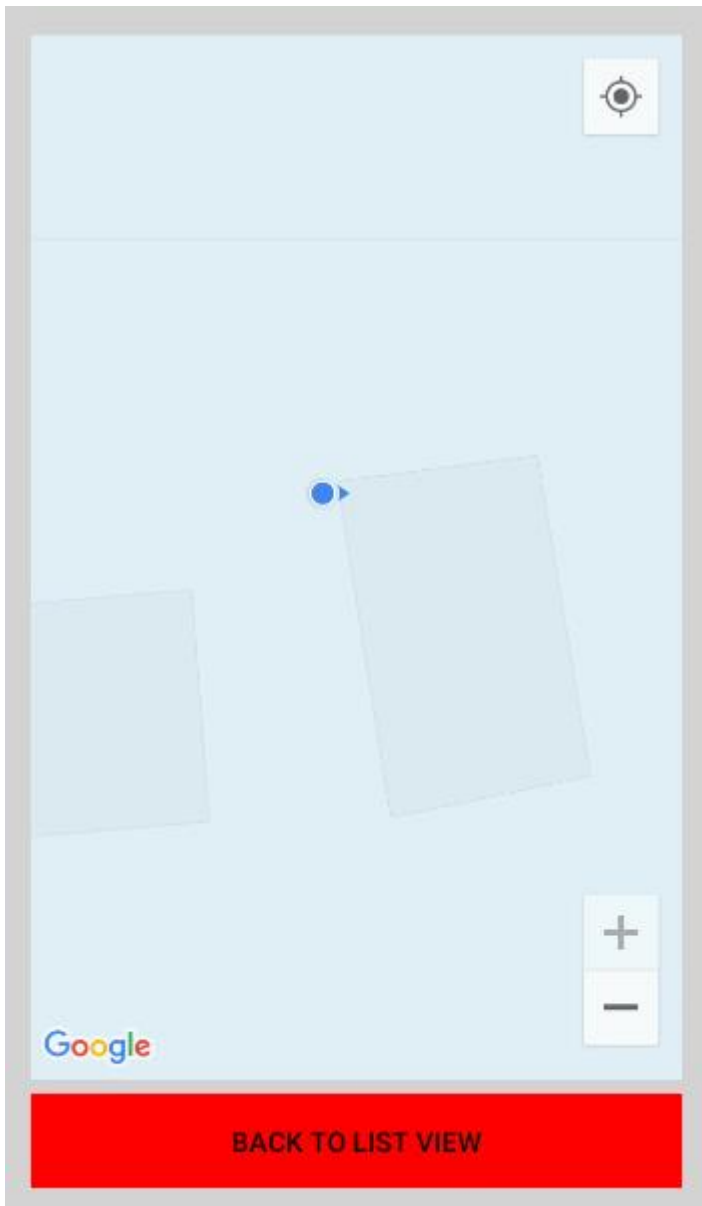


Figure E.16 Map visualization of the current location

Table E.7 Test case for update new OVI location in mobile application

Test case id	T020	Test case name	Update new OVI location in mobile application
Module name	Field activities management	Test description	Verifying a user can update the details of newly added OVI location in mobile application.
Prerequisites	User should log into the mobile application as a field assistant.	Test status	PASSED
Test step		Test data	Expected Result

1.0 Click on “OVI Field Activities” option in main menu.		1.0 User should be navigated into “OVI field activities” screen.
2.0 Select “Dry Run” as the field run from field runs list.	Field Run: Dry Run	2.0 New OVI Traps should be displayed.
3.0 Click on a OVI trap	OVI Trap: CMC-20	3.0 User should be navigated to the update OVI location wizard.
4.0 Enter valid inputs by replacing existing inputs and click on submit button.	Trap Position: flower tree Respond Name: Sunethra Perera	4.0 Details of OVI trap should be updated successfully, success message should be displayed and user should be navigated to the OVI field activities screen.

Table E.8 Test case for update BG collection in mobile application

Test case id	T021	Test case name	Update BG collection in mobile application
Module name	Field activities management	Test description	Verifying a user can update the details of BG collection in mobile application.
Prerequisites	User should log into the mobile application as a field assistant.	Test status	PASSED
Test step		Test data	Expected Result
1.0 Click on “BG Field Activities” option in main menu.			1.0 User should be navigated into BG field activities screen.
2.0 Select a BG collection field run from field runs list.	Field Run: BG_Nug_8		2.0 BG traps related to selected BG collection run should be displayed.
3.0 Click on a BG trap	BG Trap: NUG-23		3.0 User should be navigated to the update BG collection wizard.
4.0 Enter valid inputs by replacing existing inputs and click on submit button.	Trap Position: kitchen Respond Name: Kamala Fernando		4.0 Details of BG collection should be updated successfully, success message should be displayed and user should be navigated to the BG field activities screen.

Table E.9 Test case for view map layers in map view in web application

Test case id	T022	Test case name	View map layers in map view in web application
Module name	Spatial data management	Test description	Verifying a user can view map layers in a map view in web application.

Prerequisites	User should log into the web application as a GIS officer.	Test status	PASSED
Test step		Test data	Expected Result
1.0 Click on “Spatial Data” tab in main menu.			1.0 User should be navigated to the spatial list screen with a list of map layers available in the system.
2.0 Click on the “Map View” link of any map layer available in the map layer list.	Name: Nugegoda GND Map		2.0 User should be directed to the “Map View” screen and the map layer should be displayed on a map.

Table E.10 Test case for delete a map layer in web application

Test case id	T023	Test case name	Delete a map layer in web application
Module name	Spatial data management	Test description	Verifying a user can delete a map layer in web application.
Prerequisites	User should log into the web application as a GIS officer.	Test status	PASSED
Test step		Test data	Expected Result
1.0 Click on “Spatial Data” tab in main menu.			1.0 User should be navigated to the spatial list screen with a list of map layers available in the system.
2.0 Click on the “Delete” link of any map layer available in the map layer list.	Map layer name: NDCU GND Map v3		2.0 Confirm box should be displayed. Reference: figure E.17
3.0 Click on the “ok” button of confirm box			3.0 Selected map layer should be deleted from the system successfully and a success message should be displayed. Reference: figure E.18

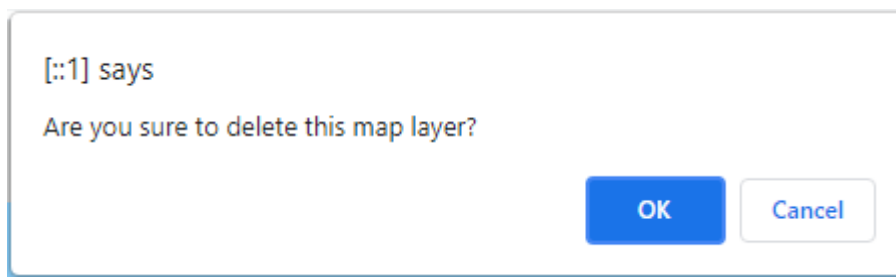


Figure E.17 Confirm box for deleting map layer

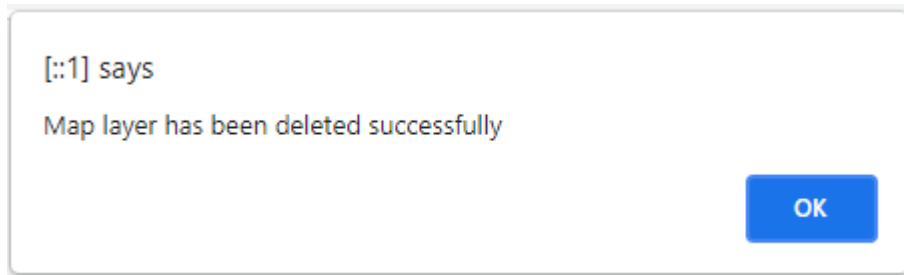


Figure E.18 Success message box after deleting map layer

Table E.11 Test case for update a map layer in web application

Test case id	T024	Test case name	Update a map layer in web application
Module name	Spatial data management	Test description	Verifying a user can update a map layer in web application.
Prerequisites	User should log into the web application as a GIS officer.	Test status	PASSED
Test step		Test data	Expected Result
1.0 Click on “Spatial Data” tab in main menu.			1.0 User should be navigated to the spatial list screen with a list of map layers available in the system.
2.0 Click on the “Update” link of any map layer available in the map layer list.	Name: Nugegoda GND Map		2.0 User should be directed to the “Update Map Layer” screen and all details related to the selected map layer should be displayed in relevant fields.
3.0 Change values of relevant fields and click on “save” button.	Description: GND Map of Nugegoda MOH area updated in 2016		3.0 Details of the map layer should be updated in system successfully and a success message box should be displayed. Then user should be navigated to the “Spatial Data” screen.