FABI System Requirements Specification
Capstone Project
Team Nova

Tegan Carton-Barber u17034729
Emma Coetzer u17074526
Aeron Land u17140634
Luveshan Marimuthu u17087709
Kendra Riddle u16224940

April 2019

Contents

1 Introduction

1.1 Purpose

The purpose of this document serves to provide an in-depth analysis into the requirements which surround the FABI Mobile project. This document will outline the requirements which will ensure that the existing system is modernized.

1.2 Scope

1.2.1 Product Identification

FABI Mobile is the primary product. It encapsulates a web and mobile application intended to drastically modernize FABI's existing system as well as to compliment the roll-out of their Diagnostic Clinic to the public. Another aspect of this project is to provide a central means for allowing internal staff within FABI to manage and maintain their databases.

1.2.2 Product Functionality

The functionality of FABI Mobile lies within two main areas. The first main area is the web application. The intention of the web application is to provide a centralized and automated means for which users can submit forms to the Diagnostic Clinic as well as track the progress of the samples that they submit. The mobile application will provide the exact same information, however, this be be done on a mobile platform. The primary difference between the web and mobile application is the fact that administrators within FABI will have access to an admin portal on the web application in order to manage sample tracking, databases, and users.

1.2.3 Product Objectives, Goals, and Benefits

The project objectives and goals are as follows:

- Provide a central platform for which internal staff of FABI can manage and maintain all databases.
- Automate all information collection processes.
- Establish a system that can track samples which have been submitted via the Diagnostic Clinic

The project benefits are far reaching. The implemented system will not only provide a scalable, maintainable solution for FABI's database storage needs, but will also greatly aid in reducing the administrative burden produced by their current paper-based system. This will allow FABI to concentrate on what matters most - Forestry and Agriculture. The deployment of FABI Mobile will provide a vital point of contact between the institute and the industry leaders in the forestry and agriculture sectors in terms of the newly established availability of the Diagnostic Clinic.

1.3 Definitions, Acronyms, and Abbreviations

- CRUD - Create, Read, Update, Delete
- FABI - Forestry and Agricultural Biotechnology Institute
- MTD - Maximum Tolerable Downtime, is the total time the system will be down during recoverability.
- PDF - Portable Document Format
- RPO - Recovery Point Objective
- SQL - Structured Query Language
- UI - User Interface
- UX - User Experience
- WRT - Work Recovery Time, the time it takes to get the system functions running after the restoration is complete.

1.4 References

1.5 Overview

The remainder of this document sets out to provide an overview of the full system, both at a high-level as well as at a subsystem specific level. The first section provides a high-level overview as illustrated by a full system domain model. The next section deals with an in-depth user characteristic analysis of the types of users that will be interacting with the system in its entirety. The document then analyses each subsystem by providing a use case diagram, a detailed list of the functional requirements for each subsystem, as well as subsystem trace-ability matrices to ensure that the requirements have been correctly allocated. The next section provides a discussion of the quality requirements for the system. This hopes to ensure that the system is not only fully functional, but optimally functional. The last section of the document provides a full system trace-ability matrix. This provides an overview of how each requirement has been allocated within the entire system. This provides us with the opportunity to ensure that requirements are evenly distributed and allocated.

2 Domain Model

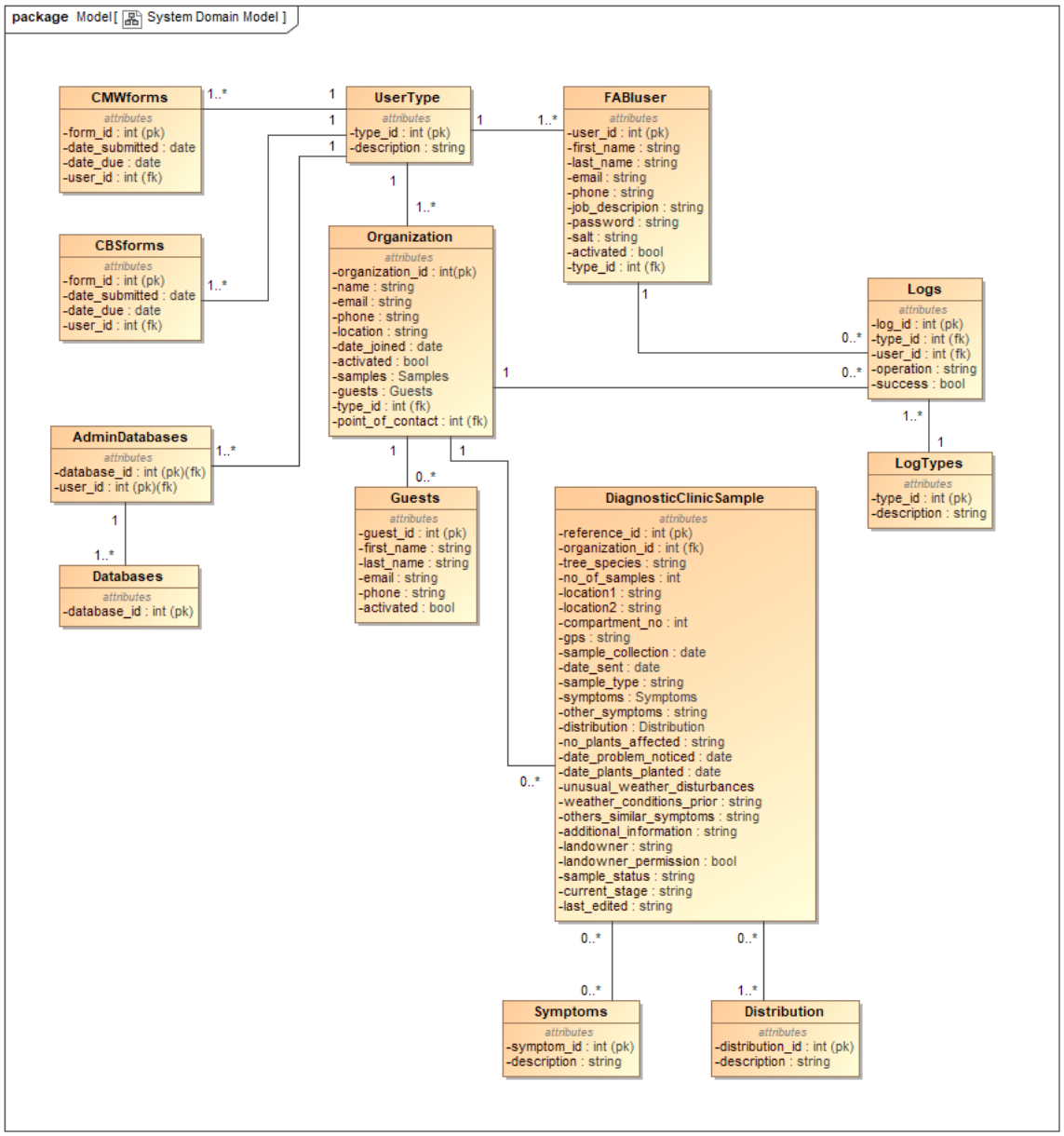

Figure 1: System Domain Model

3 User Characteristics

The system will have a user hierarchy that differentiates users based on their tier within the hierarchy. The hierarchy is as follows:

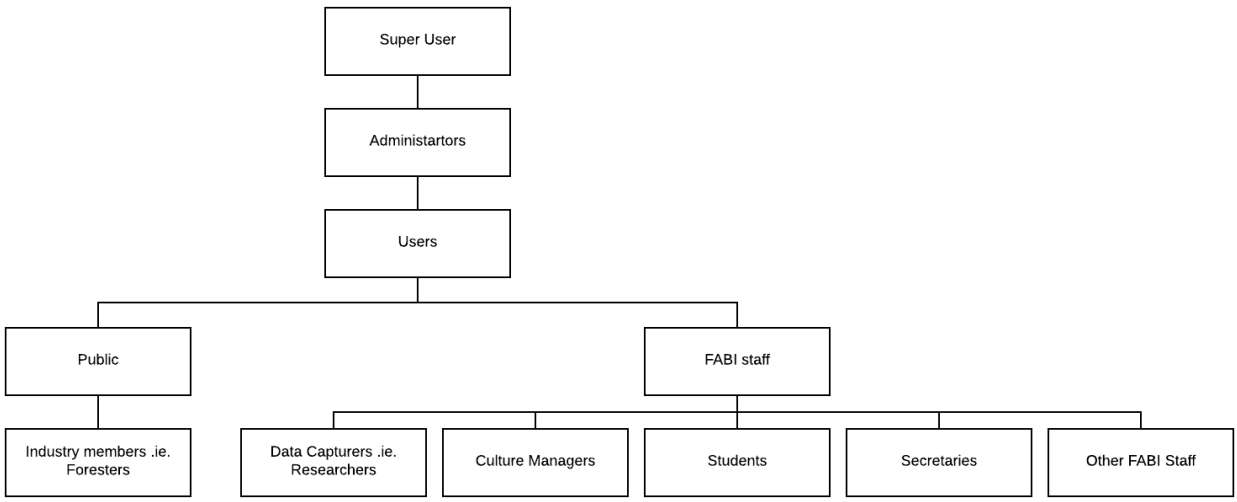

Figure 2: User Hierarchy

Super Users: These users are at the top of the hierarchy and can perform all CRUD operations on the data regardless of which database the data is in. Super users have access to both the mobile application and the web application via the 'login' interface provided by both applications. They are also able to give certain rights to specific users, as well as assign administrators to specified databases. This system will cater for the possibility of having more than one super user.

Administrators: These users may only Create, Read, and Update any data. They do not have the ability to delete data from a database (this ability is solely reserved for the super user), however, the administrators does have the ability to remove data from the database that they are assigned to and not from any other database. Administrators can be assigned to a specific database by a super user, meaning that they will oversee that database as well as who has access to the database. Administrators have access to both the mobile application and the web application via the 'login' interface provided by both applications.

Users: This group is broken down into two further sub-groups, mainly public users and FABI staff. **Public users** have to ability to read data but cannot perform any modifications to the data. This user group encapsulates stakeholders and foresters, who will primarily be involved in sending data to the system by completing the Diagnostic Clinic Form. **FABI staff** have the ability to read data as well as insert data into a database. This user group encapsulates researchers/data capturers, culture managers, students, secretaries, and any other FABI staff members. Both public and private users have access to the mobile application and web application via the 'login' interface provided by both applications.

4 Functional Requirements

The requirements to be fulfilled by the proposed system are complex in that there are many different requirements based on the centralized database, the web application, and the mobile application. Therefore, the requirements have been split up according to the subsystems.

4.1 Administration Subsystem

The purpose of the Administration Subsystem is to solely manage the users of the system as described in Section 3 (user characteristics).

4.1.1 Use Case

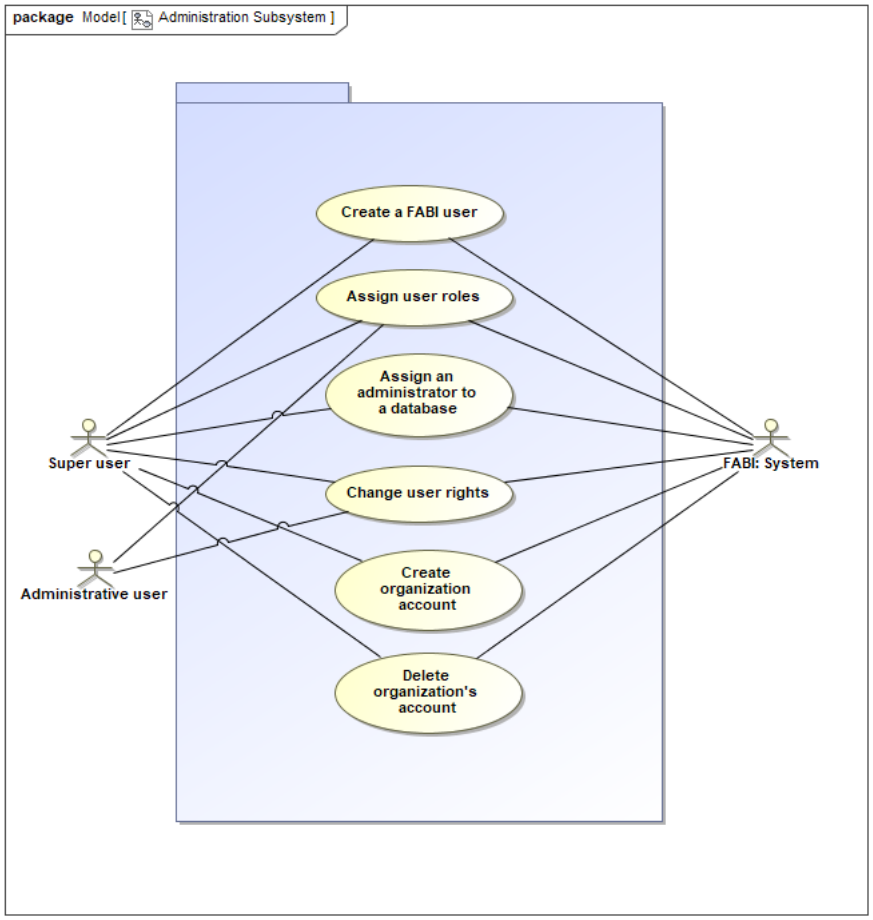

Figure 3: Administration Subsystem Use Case Diagram

4.1.2 Requirements

The following functional requirements should be fulfilled by the Administrative Subsystem:

R1. Create a FABI user

R1.1. The subsystem will provide an option to create a new user.

R1.2. The subsystem will insert the user into the appropriate database.

R2. Assign user roles

R2.1. The subsystem will provide an option to assign a user role to a user currently being created.

R2.2. The subsystem will provide an option to assign user roles to existing users.

R2.3. The subsystem will assign the appropriate rights to the user based on the user role assigned to them.

R2.4. The subsystem will ask for a super user to authorize the user role assignment if the role being assigned is that of a super user.

R2.5. The subsystem will ask for a super user to authorize the user role assignment if the role being assigned is that of a administrator.

R3. Assign an administrator to a database

R3.1. The subsystem will provide an option to assign an administrative user to manage and oversee a specific database.

R3.2. The subsystem will ask for a super user to authorize the database assignment.

R3.3. The subsystem will allocate administrative rights to the administrator of the chosen database once the action has been authorized.

R3.4. The subsystem will inform the super user(s) and other administrative users of the new user assignment to the chosen database.

R4. Change user rights

R4.1. The subsystem will allow a super user to change another user's access rights.

R4.2. The subsystem will allow a super user to sign-over their own super user rights.

R4.3. The subsystem will allow a user with administrative rights to change the rights of users using their assigned database.

R4.4. The subsystem will allow a super user to revoke any other user's rights.

R4.5. The subsystem will save all changes made to the appropriate database.

R5. Create organization account

R5.1. The subsystem will provide an option to create a new account for an organization.

R5.2. The subsystem will generate login information for the organization.

R5.3. The subsystem will insert the organization into the appropriate database.

R6. Delete organization's account

R6.1. The subsystem will provide an option deactivate an existing organization's account.

R6.2. The subsystem will only allow a user with super user rights to deactivate an organization's account.

R6.3. The subsystem will deactivate the organization account.

4.1.3 Subsystem Allocation

	Create a FABI user	Assign user roles	Assign an Administrator to a database	Change user rights	Create organization account	Delete organization's account
R1.						
R1.1.	X					
R1.2.	X					
R2.						
R2.1.		X				
R2.2.		X				
R2.3.		X				
R2.4.		X				
R2.5.		X				
R3.						
R3.1.			X			
R3.2.			X			
R3.3.			X			
R3.4.			X			
R4.						
R4.1.				X		
R4.1.				X		
R4.3.				X		
R4.4.				X		
R4.5.				X		
R5.						
R5.1.					X	
R5.1.					X	
R5.3.					X	
R6.						
R6.1.						X
R6.1.						X
R6.3.						X

Table 1: Administration Subsystem Traceability Matrix

4.2 Authentication Subsystem

The purpose of the Authentication Subsystem is to handle all authentication requests and manage any form of authentication required by the system when a user interacts with the system.

4.2.1 Use Case

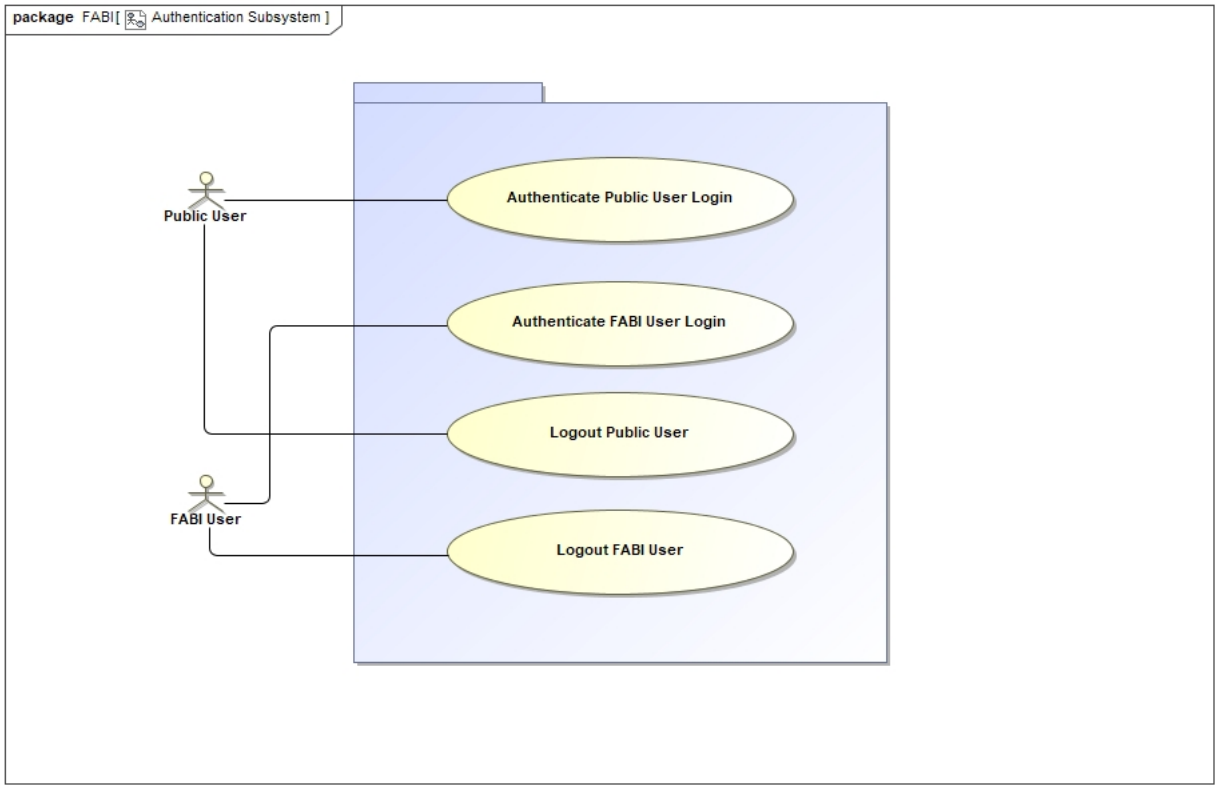

Figure 4: Authentication Subsystem Use Case Diagram

4.2.2 Requirements

The following requirements should be fulfilled by the Authentication Subsystem:

R1. Authenticate public user login

R1.1. The subsystem will authenticate the log in details of a public user who wishes to log into the web or mobile app.

R1.2. The log in details are authenticated with the log in details of the organisation that the public user is a member of.

R1.3. The user is notified if their log in details are Authenticated or Not-Authenticated.

R2. Authenticate FABI user login

R2.1. The subsystem will authenticate the log in details of a FABI user who wishes to log into the web or mobile app.

R2.2. The log in details are authenticated with the log in details of the FABI user's personal account.

R2.3. The user is notified if their log in details are Authenticated or Not-Authenticated.

R3. Logout public user

R3.1. The subsystem will log the public user out of the web or mobile app.

R4. Logout FABI user

R4.1. The subsystem will log the FABI user out of the web or mobile app.

4.2.3 Subsystem Allocation

	Authenticate public user login	Authenticate FABI user login	Logout public user	Logout FABI user
R1.				
R1.1.	X			
R1.2.	X			
R1.3.	X			
R2.				
R2.1.		X		
R2.2.		X		
R2.3.		X		
R3.				
R3.1.			X	
R4.				
R4.1.				X

Table 2: Authentication Subsystem Traceability Matrix

4.3 Diagnostic Clinic Handler Subsystem

The purpose of the Diagnostic Clinic Handler Subsystem is to handle any process that the Diagnostic Clinic supports. The subsystem manages both the internal functions of the Diagnostic Clinic, as well as the functions provided to users who interact with the clinic.

4.3.1 Use Case

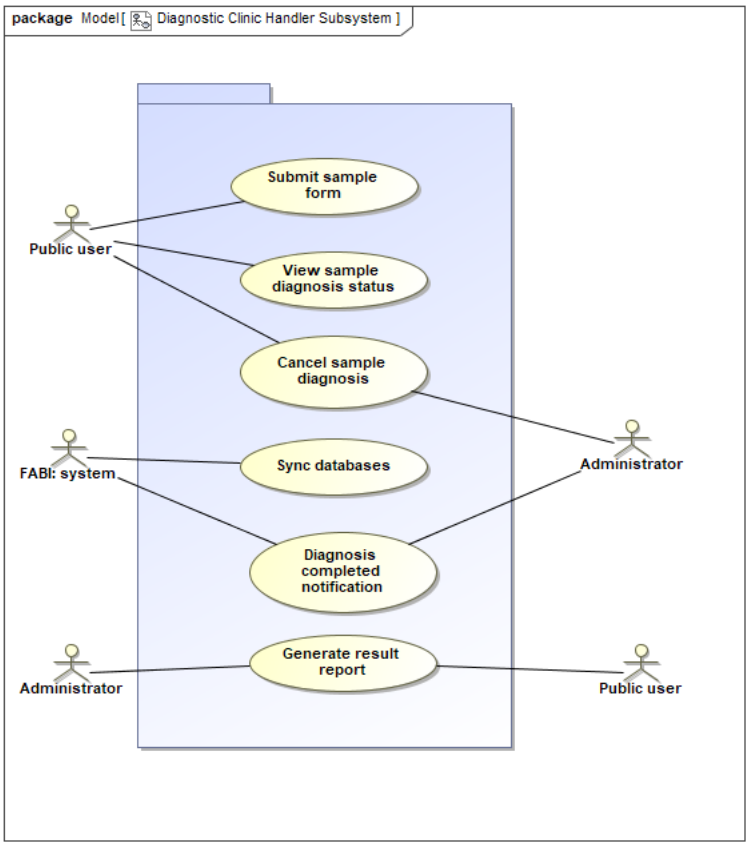

Figure 5: Diagnostic Clinic Handler Subsystem Use Case Diagram

4.3.2 Requirements

The following functional requirements should be fulfilled by the Diagnostic Clinic Handler Subsystem:

R1. Submit sample form

R1.1. The subsystem generates a reference number that uniquely identifies the sample in the Diagnostic Clinic based on the sample submitted.

R1.2. The subsystem will notify the public user of the reference number.

R2. View sample diagnosis status

R2.1. The subsystem will return the status and progress of the sample's diagnosis.

R3. Cancel sample diagnosis

R3.1. The subsystem will cancel the diagnosis of the the sample.

R3.2. The subsystem will notify the public user of the cancellation of the sample's diagnosis.

R3.3. The subsystem will notify the administrator of the cancellation of the sample's diagnosis.

R4. Sync databases

R4.1. Throughout the process of diagnosing the sample, the subsystem will update the respective databases with data retrieved during the process. This ensures that the databases and the data being processed are synced at all times.

R5. Diagnosis completed notification

R5.1. Once the diagnosis is complete, the subsystem will notify the administrator that the diagnosis is complete.

R6. Generate result report

R6.1. Once the administrator has received the diagnosis completion notification, the subsystem will generate a report with the result of the diagnosis. This step is initiated by the administrator.

R6.2 The subsystem will style the report with an existing report template specific to the organization that the public user is a member of.

R6.3. The subsystem will send the report to the public user who submitted the sample.

4.3.3 Subsystem Allocation

	Submit sample form	View sample diagnosis status	Cancel sample diagnosis	Sync databases	Diagnosis completed notification	Generate result report
R1.						
R1.1.	X					
R1.2.	X					
R2.						
R2.1.		X				
R3.						
R3.1.			X			
R3.2.			X			
R3.3.			X			
R4.						
R4.1.				X		
R5.						
R5.1.					X	
R6.						
R6.1.						X
R6.2.						X
R6.3.						X

Table 3: Diagnostic Clinic Handler Subsystem Traceability Matrix

4.4 Central Data Access Point Subsystem

This subsystem will be used by admin users of the application to curate and manage databases within the system.

4.4.1 Use Case

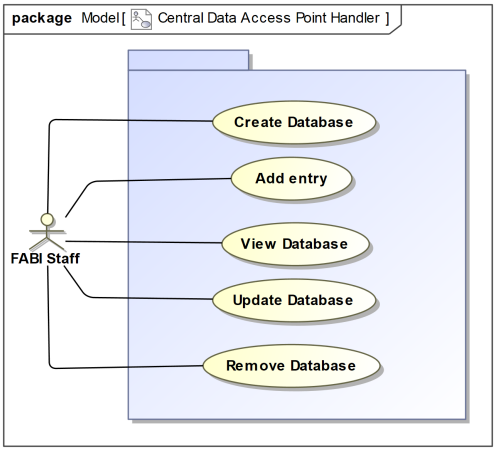

Figure 6: Central Data Access Point Handler Subsystem User Case Diagram

4.4.2 Requirements

The following functional requirements should be fulfilled by the Central Data Access Point Subsystem:

R1. Create database

R1.1 The subsystem will allow admins to create a new database.

R2.Add entry

R2.1. The subsystem will allow a user add a single entry into one of the databases.

R2.2. The subsystem will allow a user to import multiple entries into a given database.

R3. View database

R3.1. The subsystem will allow a user to search the database for a given entry or entries.

R3.2. The subsystem will provide a simple interface to view the full database.

R4. Update database

R4.1. The subsystem will allow a user to update an entry found in the database if necessary.

R5. Remove from Database

R5.1. The subsystem will allow admin users to remove an entry in a database.

R5.2. The subsystem will give admin users the ability to drop a database, with ample warning and verification.

4.4.3 Subsystem Allocation

	Create database	Add entry	View database	Update database	Remove database
R1.					
R1.1.	X				
R2.					
R2.1.		X			
R2.2.		X			
R3.					
R3.1.			X		
R3.2.			X		
R4.					
R4.1.				X	
R5.					
R5.1.					X
R5.1.					X

Table 4: Central Data Access Subsystem Traceability Matrix

4.5 FABI Internal Form Handler Subsystem

This subsystem will manage and store the forms submitted to FABI and allow FABI to generate a list of forms submitted within a time period.

4.5.1 Use Case

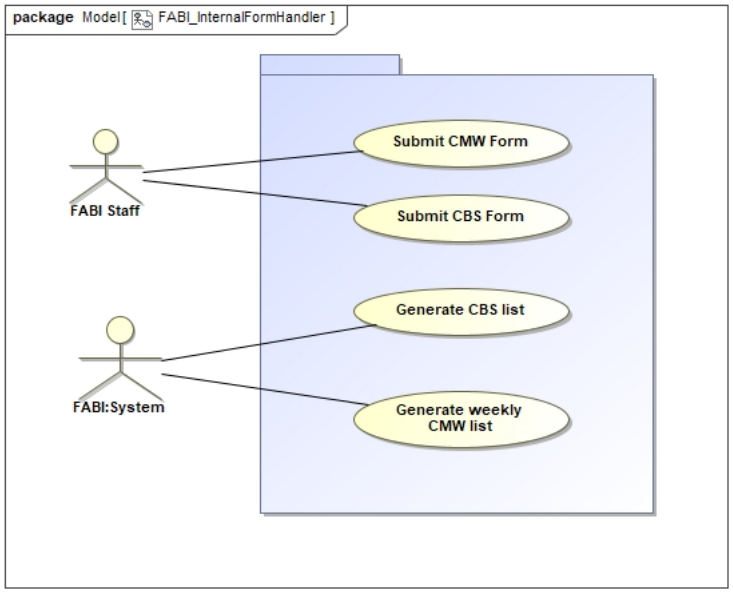

Figure 7: Internal Form Handler Use Case Diagram

4.5.2 Requirements

The following functional requirements should be fulfilled by the Internal Form Handler Subsystem:

R1. Submit CMW form

R1.1. The subsystem will allow FABI staff members to submit a CMW form to the FABI culture collection.

R2. Submit CBS form

R2.1. The subsystem will allow FABI staff members to submit a CBS form that will be sent by FABI to the international culture collection.

R3. Generate CBS list

R3.1. The subsystem will generate a list of all CBS forms that need to be sent to the international culture collection.

R4. Generate weekly CMW list

R4.1. The subsystem will generate a list of all CMW forms every week to be sent to the FABI culture collection .

4.5.3 Subsystem Allocation

	Submit CMW form	Submit CBS form	Generate CBS list	Generate weekly CMW list
R1.				
R1.1.	X			
R2.				
R2.1.		X		
R3.				
R3.1.			X	
R4.				
R4.1.				X

Table 5: Internal Form Handler Subsystem Traceability Matrix

4.6 Organization Information Subsystem

The purpose of the Organization Information Subsystem is to manage the organizations that make use of the FABI mobile system.

4.6.1 Use Case

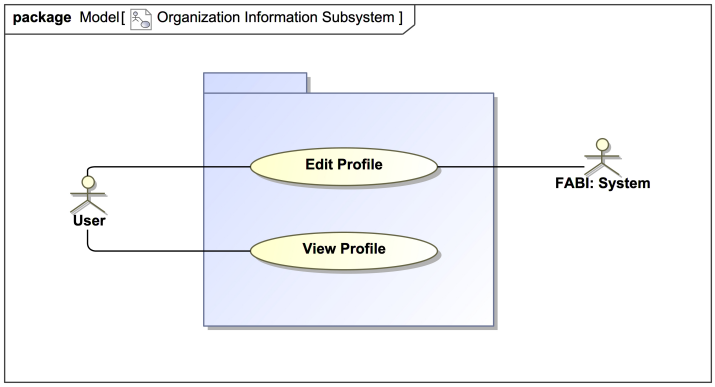

Figure 8: Organization Information Use Case Diagram

4.6.2 Requirements

The following functional requirements should be fulfilled by the Organization Information Subsystem:

R1. Edit Details

R1.1. The subsystem will allow the organization to edit their details.

R1.2. The subsystem will register these changes with the system.

R2. View Details

R2.1. The subsystem will allow the organization to view their details.

4.6.3 Subsystem Allocation

	Edit Details	View Details
R1.		
R1.1.	X	
R1.2.	X	
R2.		
R2.1.		X

Table 6: Organization Information Subsystem Traceability Matrix

4.7 Porting Subsystem

This subsystem will be used to transfer data in large quantities from legacy databases which already exist. It will provide administrators functionality to upload CSV files, which can then be processed and imported into the relevant database.

4.7.1 Use Case

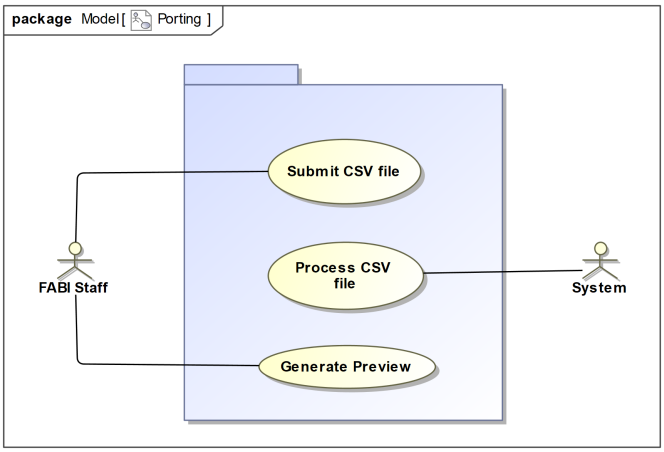

Figure 9: Porting Subsystem Use Case Diagram

4.7.2 Requirements

The following functional requirements should be fulfilled by the Porting Subsystem:

R1. CSV submission

R1.1 The subsystem will allow administrators to submit a CSV file to be processed into a relevant database.

R2. Process the submitted CSV file

R2.1. The subsystem will process the data from the file into a usable data format.

R2.2. The subsystem will import this processed data into the relevant database.

R3. Preview Submission

R3.1. The subsystem will show a preview of the data to be submitted.

R3.2. The subsystem will provide users functionality to commit the data to the database.

4.7.3 Subsystem Allocation

	CSV submission	Process the submitted CSV file	Preview submission
R1.			
R1.1.	X		
R2.			
R2.1.		X	
R2.2.		X	
R3.			
R3.1.			X
R3.2.			X

Table 7: Porting Subsystem Traceability Matrix

4.8 Sample Tracking Subsystem

The purpose of the Sample Tracking Subsystem is to manage the processing stages of the Diagnostic Clinic. These stages are the stages that the sample goes through when in the Diagnostic Clinic. The subsystem will also manage the tracking of the sample during it's diagnosis and which processing stage it is in.

4.8.1 Use Case

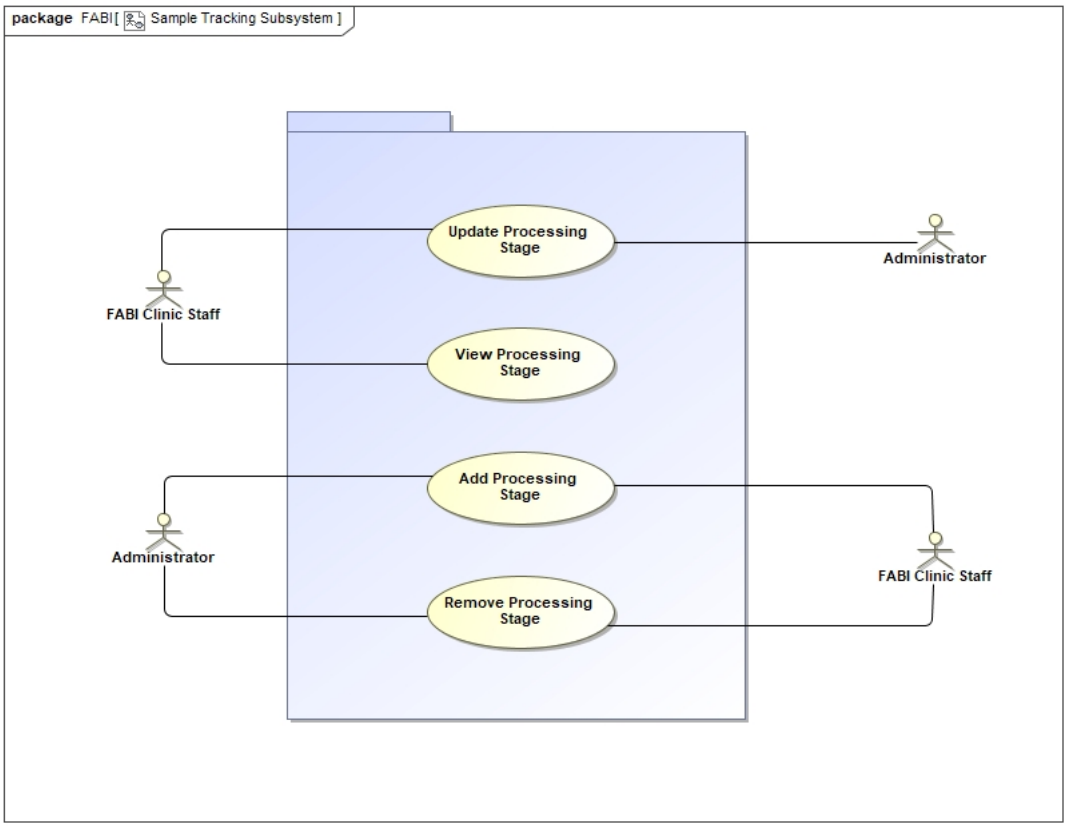

Figure 10: Sample Tracking Subsystem Use Case Diagram

4.8.2 Requirements

The following functional requirements should be fulfilled by the Sample Tracking Subsystem:

R1. Update processing stage

R1.1. The subsystem will update which stage of the diagnostic process the sample is in.

R1.2. The subsystem will notify the administrator that the
stage of the diagnostic process that the sample is in has been updated.

R2. View processing stage

R2.1. The subsystem will return which stage of the diagnostic process the sample is currently in.

R3. Add processing stage

R3.1. The subsystem will add a processing stage to the diagnostic process.

R3.2. The subsystem will notify all the FABI clinic staff that the specific stage has been added to the diagnostic process.

R4. Remove processing stage

R4.1. The subsystem will remove a processing stage from the diagnostic process.

R4.2. The subsystem will notify all the FABI clinic staff that the specific stage has been removed from the diagnostic process.

4.8.3 Subsystem Allocation

	Update processing stage	View processing stage	Add processing stage	Remove processing stage
R1.				
R1.1.	X			
R1.2.	X			
R2.				
R2.1.		X		
R3.				
R3.1.			X	
R3.2.			X	
R4.				
R4.1.				X
R4.2.				X

Table 8: Sample Tracking Subsystem Traceability Matrix

4.9 Notification Subsystem

This subsystem will be used to send email notifications to researchers and clients for certain events in the system.

4.9.1 Use Case

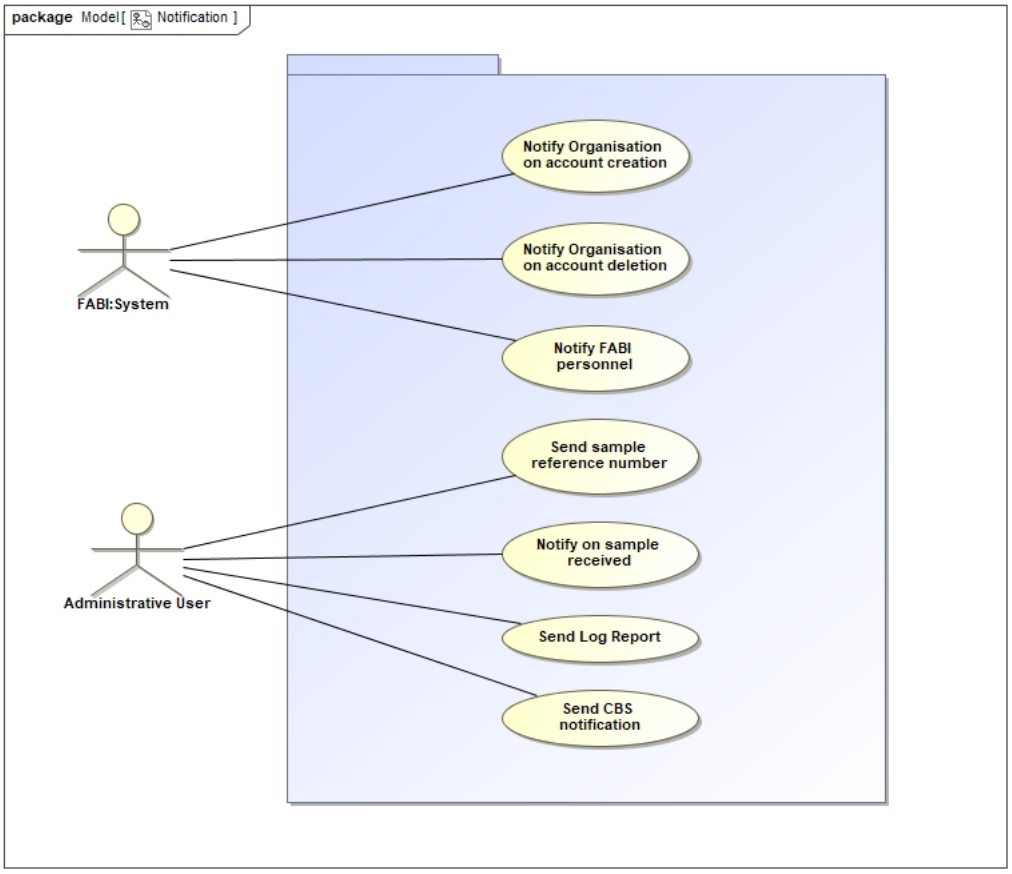

Figure 11: Notification Use Case Diagram

4.9.2 Requirements

The following functional requirements should be fulfilled by the Notification Subsystem:

R1. Notify organisation on account creation

R1.1. The subsystem will send a notification to organisation when an account is created for them.

R2. Notify organisation on account deletion

R2.1. The subsystem will send a notification to organisation when their account is deleted.

R3. Notify FABI personnel

R3.1. Send notification to FABI personnel when their account details change.

R4. Send sample reference number

R4.1. The subsystem will send a notification to client with a reference number for submitting samples.

R5. Notify on sample received

R5.1. The subsystem will send a notification to client when their sample has been received by FABI.

R6. Send log report

R6.1. The subsystem will send a structured report of the logs of the system.

R7. Send CBS notification

R7.1. The subsystem will send a notification when the CBS form list becomes too large i.e batch needs to be sent.

4.9.3 Subsystem Allocation

	Account creation	Account deletion	FABI personnel	Send reference number	Sample received	Send log report	Send CBS notification
R1.							
R1.1.	X						
R2.							
R2.1.		X					
R3.							
R3.1.			X				
R4.							
R4.1.				X			
R5.							
R5.1.					X		
R6.							
R6.1.						X	
R7.							
R7.1.							X

Table 9: Notification Subsystem Traceability Matrix

4.10 Logging Subsystem

The purpose of the Logging Subsystem is to keep record of certain activities that occur on the FABI Mobile system. These logs are used to generate various reports as set out by the reporting subsystem (Section 4.11).

4.10.1 Use Case

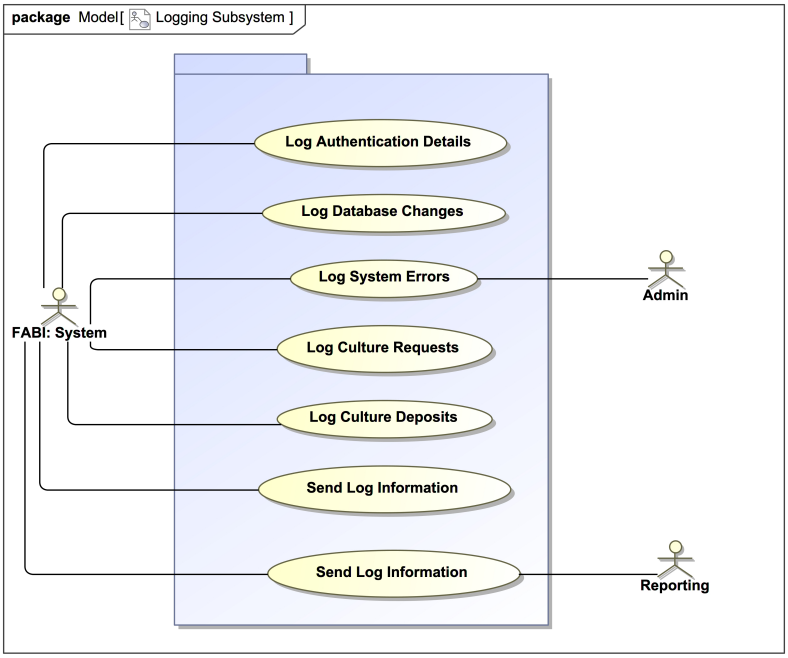

Figure 12: Logging Use Case Diagram

4.10.2 Requirements

The following functional requirements should be fulfilled by the Logging Subsystem:

R1. Log authentication details

R1.1. The subsystem will log information relating to how many successful or failed authentications there have been.

R2. Log database changes

R2.1. The subsystem will log information relating to any changes that take place in the database

R3. Log system errors

R3.1. The subsystem will log information relating to any errors that occur within the system

R4. Log Culture Requests

R4.1. The subsystem will log information relating to requests that have been submitted for culture collections

R5. Log Culture Deposits

R5.1. The subsystem will log information relating to deposits that have been submitted for culture collections

R6. Log Sample Submissions

R6.1. The subsystem will log information relating to sample submissions that are made for the diagnostic clinic

R7. Send Log Information

R7.1. The subsystem will send all log information to the reporting subsystem

4.10.3 Subsystem Allocation

	Log Authentication Details	Log Database Changes	Log System Errors	Log Culture Requests	Log Culture Deposits	Log Sample Submissions	Send Log Information
R1.							
R1.1.	X						
R2.							
R2.1.		X					
R3.							
R3.1.			X				
R4.							
R4.1.				X			
R5.							
R5.1.					X		
R6.							
R6.1.						X	
R7.							
R7.1.							X

Table 10: Logging Subsystem Traceability Matrix

4.11 Reporting Subsystem

The purpose of the Reporting Subsystem is to generate reports on demand. These reports pull data from the logging subsystem in order to generate the reports.

4.11.1 Use Case

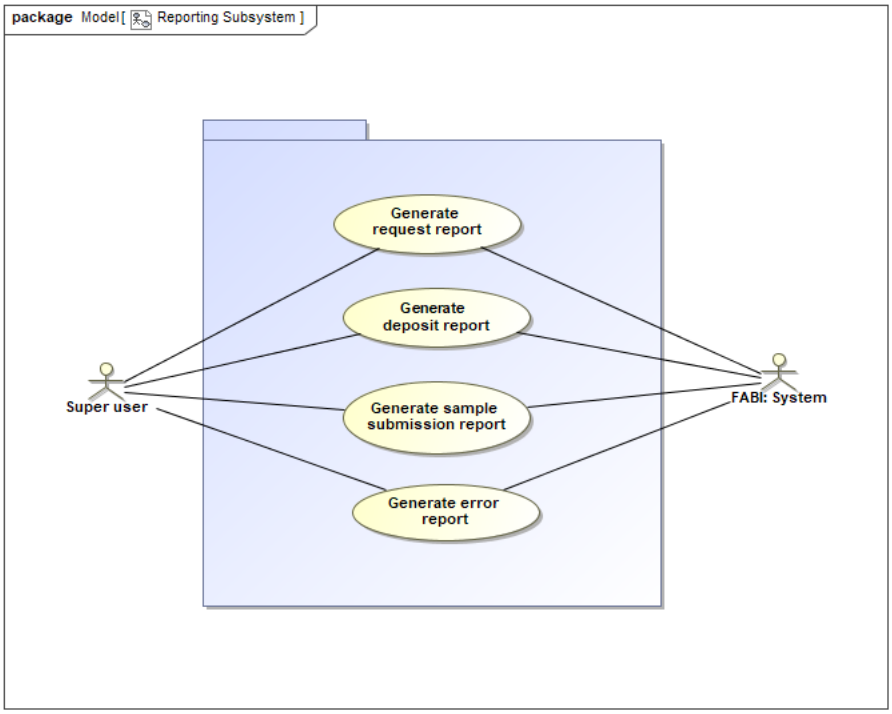

Figure 13: Reporting Subsystem Use Case Diagram

4.11.2 Requirements

The following requirements should be fulfilled by the Reporting Subsystem:

R1. Generate request report

R1.1. The subsystem will provide an option to generate a request report on demand.

R1.2. The subsystem will pull data from the report information logged in order to generate the report.

R1.3. The subsystem will provide an option to download the report as a PDF.

R2. Generate deposit report

R2.1. The subsystem will provide an option to generate a deposit report on demand.

R2.2. The subsystem will pull data from the deposit information logged in order to generate the report.

R2.3. The subsystem will provide an option to download the report as a PDF.

R3. Generate sample submission report

R3.1. The subsystem will provide an option to generate a sample submission report on demand.

R3.2. The subsystem will pull data from the sample submission information logged in order to generate the report.

R3.3. The subsystem will provide an option to download the report as a PDF.

R4. Generate error report

R4.1. The subsystem will provide an option to generate an error report on demand.

R4.2. The subsystem will pull data from the error information logged in order to generate the report.

R4.3. The subsystem will provide an option to download the report as a PDF.

4.11.3 Subsystem Allocation

	Generate request report	Generate deposit report	Generate sample submission report	Generate error report
R1.				
R1.1.	X			
R1.2.	X			
R1.3.	X			
R2.				
R2.1.		X		
R2.2.		X		
R2.3.		X		
R3.				
R3.1.			X	
R3.2.			X	
R3.3.			X	
R4.				
R4.1.				X
R4.2.				X
R4.3.				X

Table 11: Reporting Subsystem Traceability Matrix

5 Quality Requirements

5.1 Scalability

Scalability is an attribute that describes the ability of a software system to grow and manage increased demand. [5] The crux of FABI Mobile in terms of scalability requirements lies in the scalability of their database system and the ability of which this system can cope when multiple requests operations are performed on these databases, whilst still maintaining optimal performance. This system requires a scalable database storage solution which is why the choice of a NoSQL database has been chosen.

5.2 Extensibility

Extensibility refers to the ability to extend the system's functionality with little to no affect on other internal functions.[1] In order to produce an extensible system for FABI the system will be built in a modular manner which promotes loose coupling. This will enable FABI to easily add or remove databases from the system. Another extensible solution is that FABI Mobile administrators will be able to dynamically build or change their submission forms in order to accommodate changing needs.

5.3 Performance

The performance of the system describes the overall responsiveness and capability of the system. It also defines how well the system performs its function under all conditions. The performance of the system is of vital importance, as poor performance leads to a negative user experience and may jeopardise the overall safety and throughput of the system. The system should be able to process images quickly and effectively via the mobile application. The system will integrate information efficiently and effectively. The system should be available everywhere, regardless of the users' location. Thus the system will provide an offline feature for users to still have access when they do not have internet connectivity. The up-time of the system has not yet been specified.

5.4 Security

The security of the system describes the overall user-access, session control, data integrity and data security of the system. The system will provide intensive user-access control. The users will only be authorised to access certain features/functions of the system based on their user type. These user types are described in the user characteristics (part 3). Only specified user types and thus authorised personnel will be allowed to add, remove and specifically update information in the database. The other user types' access will be controlled, by allowing only certain user types to view certain data and by restricting which user types may add and update data. The system will ensure that all the data is securely handled, as it deals with and stores very sensitive and important data. This will, along with the next point, ensure the integrity of the data. The system will store the data in a secure location and in a secure format. The specifications of these security features have not yet been identified. One data storing format to ensure security is that encryption will be used for important and sensitive data.

5.5 Interface

Interface requirements are hugely important to reaching the goals of the proposed software system. It involves the consideration of those who will be interacting with the system. The primary interface requirements for this system is that the system is easy and simple to use. This is, because many of the users interacting with the system currently are users which do not have a high exposure to technology. This is primary goal. The interface must also provide ways which automates as much of the process as possible as to reduce the administrative burden.

5.6 Availability

Availability refers to the ability of a user to access information or resources in a specified location and in the correct format.[3] In terms of FABI Mobile, availability is essential. Especially when it comes to the correctness of the information. The institute deals with information which is technically specific. It is therefore of utmost importance that the information is presented accordingly. In terms of access, it is important that the mobile application work offline. This is because many of the submissions will be made from people working the field where there is often very little signal. An offline capability will greatly improve the systems availability.

5.7 Concurrency and Capacity

The concurrency of the system describes the ability of the system to handle multiple concurrent connections and requests. The system will handle concurrent connections and requests. The number of requests and the number of user connections that the system can handle has not yet been identified. These specifications will be identified later to ensure sufficient functionality is delivered to the user. Capacity defines the ways in which the systems may be expected to scale-up by increasing hardware capacity based on the organisation's volume projections. FABI has specified that a large volume of data has to be stored by the system. The system will be able to store large amounts of data and the system will store data as long as FABI deems it necessary. The specifications as to how much data has not yet been identified, as the organisation's hardware capacity and architecture has not been specified. The system will be able to deal with the transmission of data between functions within the system, as well as external systems. A data threshold will be defined later to ensure that the system has optimal performance and throughput. Other data transmission specifications have not yet been identified.

5.8 Recoverability

Recoverability refers to ability of the system to respond to failure and restore the system to a functioning state. Recovery is of vital importance in this system, as the system holds very valuable information. Thus, the system will have backup databases to store backup data. Data will be backed-up every week. Using the concept of the Recovery Point Objective (RPO), this will ensure that the system's data recoverability is high. The system will provide recoverability for server failures, storage failures and application failures. The system will not provide recoverability for administrative failures, such as an administrator mistakenly deleting data or performing some other irreversible action. The recovery process' time has not yet been identified, but the system will account for Maximum Tolerable Downtime (MTD). The recoverability process consists of the Recovery Time Object (RTO), which is the time it will take to get the backups and restore the system to an operational state. The process also consists of the Work Recovery Time (WRT). MTD = RTO + WRT

5.9 Maintainability

Maintainability is defined as the degree to which an application is understood, repaired, or enhanced[2] It is crucial that the system is maintainable. This is due to the fact that the system will continually be growing as new entries are added to the database. This in turn will need to be maintained. The interfaces will need to undergo maintenance as well.

5.10 Integrity

Integrity refers to methods of ensuring that data is real, accurate and safeguarded from unauthorized user modification.[4] In terms of FABI Mobile data integrity is vital. A user management system which enforces a strict user hierarchy is used to ensure no unauthorized access occurs. Interface forms are protected against SQL injections to ensure that sensitive information from within the database is not extracted. All information that is added, edited, or removed from the database has to undergo authorization from the admin of that database.

5.11 Usability

The usability of the system defines the user experience(UX) and user interface (UI) of the system. This is to ensure that the system is usable by all users and ensure the end users' satisfaction.
The system will provide for an easy to navigate and understand interface. The interface will be designed with the colours of FABI and will keep a simple layout. The interfaces of the web application and mobile application will be very similar in design, with only size and layout varying to accommodate the platform the application is on. The system will use English as its primary language and design the app so that it can be used by both local and international users.

5.12 Interoperability

The interoperability of the system defines the ability of the system to exchange and share information with internal and external systems.
The system will provide for the sharing of data internally between systems within FABI, as well as other external systems. The data formats, transport protocols, encoding and other such aspects have not yet been identified but will be used to ensure the system is interoperable.

6 Trace-ability Matrix

	Administration Subsystem	Authentication Subsystem	Diagnostic Clinic Handler Subsystem	Central Data Access Point Subsystem	FABI Internal Form Handler Subsystem	User Information Subsystem	Porting Subsystem	Sample Tracking Subsystem	Notification Subsystem	Logging Subsystem	Reporting Subsystem
Administration Subsystem											
R1.	X										
R2.	X										
R3.	X										
R4.	X										
R5.	X										
R6.	X										
Authentication Subsystem											
R1.		X									
R2.		X									
R3.		X									
R4.		X									
Diagnostic Clinic Handler Subsystem											
R1.			X								
R2.			X								
R3.			X								
R4.			X								
R5.			X								
R6.			X								
Central Data Access Point Subsystem											
R1.				X							
R2.				X							
R3.				X							
R4.				X							
R5.				X							
FABI Internal Form Handler Subsystem											
R1.					X						
R2.					X						
R3.					X						
R4.					X						
Organization Information Subsystem											
R1.						X					
R2.						X					
Porting Subsystem											
R1.							X				
R2.							X				
R3.							X				
Sample Tracking Subsystem											
R1.								X			
R2.								X			
R3.								X			
R4.								X			
Notification Subsystem											
R1.									X		
R2.									X		
R3.									X		
R4.									X		
R5.									X		
R6.									X		
R7.									X		

	Administration Subsystem	Authentication Subsystem	Diagnostic Clinic Handler Subsystem	Central Data Access Point Subsystem	FABI Internal Form Handler Subsystem	User Information Subsystem	Porting Subsystem	Sample Tracking Subsystem	Notification Subsystem	Logging Subsystem	Reporting Subsystem
Logging Subsystem											
R1.										X	
R2.										X	
R3.										X	
R4.										X	
R5.										X	
R6.										X	
R7.										X	
Reporting Subsystem											
R1.											X
R2.											X
R3.											X
R4.											X

Table 12: System Traceability Matrix

References

[1] *Software Design - Extensibility (Modularity without compilation).* URL: https://gerardnico.com/system/property/extensibility. (accessed: 27.04.2019).

[2] *Software Maintainability: 75% of Your Budget Is Dedicated to Software Maintenance.* URL: https://www.castsoftware.com/glossary/software-maintainability. (accessed: 27.04.2019).

[3] *What does Availability mean?* URL: https://www.techopedia.com/definition/990/availability. (accessed: 27.04.2019).

[4] *What does Integrity mean?* URL: https://www.techopedia.com/definition/10284/integrity. (accessed: 27.04.2019).

[5] *What does Scalability mean?* URL: https://www.techopedia.com/definition/9269/scalability. (accessed: 27.04.2019).