

# FABI System Requirements Specification

Version 2

Capstone Project

Team Nova

Tegan Carton-Barber u17034729

Emma Coetzer u17074526

Aeron Land u17140634

Luveshan Marimuthu u17087709

Kendra Riddle u16224940

April 2019



# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	Purpose . . . . .	3
1.2	Scope . . . . .	3
1.2.1	Product Identification . . . . .	3
1.2.2	Product Functionality . . . . .	3
1.2.3	Product Objectives, Goals, and Benefits . . . . .	3
1.3	Definitions, Acronyms, and Abbreviations . . . . .	3
1.4	Overview . . . . .	4
<b>2</b>	<b>User Characteristics</b>	<b>5</b>
<b>3</b>	<b>Constraints</b>	<b>6</b>
3.1	System Constraints . . . . .	6
3.2	Environmental and Technology Constraints . . . . .	6
3.2.1	Software Constraints . . . . .	6
3.2.2	Hardware Constraints . . . . .	6
<b>4</b>	<b>Domain Model</b>	<b>7</b>
<b>5</b>	<b>Architectural Design</b>	<b>8</b>
<b>6</b>	<b>Functional Requirements</b>	<b>9</b>
6.1	Authentication Subsystem . . . . .	9
6.1.1	Use Case . . . . .	9
6.1.2	Requirements . . . . .	10
6.1.3	Subsystem Allocation . . . . .	12
6.2	Staff Handler Subsystem . . . . .	13
6.2.1	Use Case . . . . .	13
6.2.2	Requirements . . . . .	13
6.2.3	Subsystem Allocation . . . . .	14
6.3	Diagnostic Clinic Handler Subsystem . . . . .	15
6.3.1	Use Case . . . . .	15
6.3.2	Requirements . . . . .	15
6.3.3	Subsystem Allocation . . . . .	16
6.4	Database Handler Subsystem . . . . .	17
6.4.1	Use Case . . . . .	17
6.4.2	Requirements . . . . .	17
6.4.3	Subsystem Allocation . . . . .	18
6.5	Culture Collection Handler Subsystem . . . . .	19
6.5.1	Use Case . . . . .	19
6.5.2	Requirements . . . . .	19
6.5.3	Subsystem Allocation . . . . .	20
6.6	Organization Handler Subsystem . . . . .	21
6.6.1	Use Case . . . . .	21
6.6.2	Requirements . . . . .	21
6.6.3	Subsystem Allocation . . . . .	22
6.7	Organization Subsystem . . . . .	23
6.7.1	Use Case . . . . .	23
6.7.2	Requirements . . . . .	23
6.7.3	Subsystem Allocation . . . . .	25
6.8	Organization Member Subsystem . . . . .	26
6.8.1	Use Case . . . . .	26
6.8.2	Requirements . . . . .	26
6.8.3	Subsystem Allocation . . . . .	27
6.9	Porting Subsystem . . . . .	28
6.9.1	Use Case . . . . .	28
6.9.2	Requirements . . . . .	28
6.9.3	Subsystem Allocation . . . . .	29
6.10	Sample Processing Subsystem . . . . .	30
6.10.1	Use Case . . . . .	30

---

6.10.2	Requirements . . . . .	30
6.10.3	Subsystem Allocation . . . . .	32
6.11	Notification Subsystem . . . . .	33
6.11.1	Use Case . . . . .	33
6.11.2	Requirements . . . . .	33
6.11.3	Subsystem Allocation . . . . .	34
6.12	Logging Subsystem . . . . .	35
6.12.1	Use Case . . . . .	35
6.12.2	Requirements . . . . .	35
6.12.3	Subsystem Allocation . . . . .	36
6.13	Reporting Subsystem . . . . .	37
6.13.1	Use Case . . . . .	37
6.13.2	Requirements . . . . .	37
6.13.3	Subsystem Allocation . . . . .	38
<b>7</b>	<b>Quality Requirements</b>	<b>39</b>
7.1	Scalability . . . . .	39
7.2	Extensibility . . . . .	39
7.3	Performance . . . . .	39
7.4	Security . . . . .	39
7.5	Interface . . . . .	39
7.6	Availability . . . . .	39
7.7	Concurrency and Capacity . . . . .	40
7.8	Recoverability . . . . .	40
7.9	Maintainability . . . . .	40
7.10	Integrity . . . . .	40
7.11	Usability . . . . .	40
7.12	Interoperability . . . . .	40
<b>8</b>	<b>Technologies</b>	<b>41</b>
<b>9</b>	<b>Trace-ability Matrix</b>	<b>42</b>

# 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 forestry and agriculture sector. Another aspect of this project is to provide a central means for allowing internal staff within FABI to manage and maintain their databases, staff, and associated organizations. An external management system has been put in place for industry organizations to manage the guests that are associated to them and can in turn make use of the Diagnostic Clinic facilities.

### 1.2.2 Product Functionality

The functionality of FABI Mobile lies within two main areas. The first main area is the web application. The primary intention of the web application is to enable FABI administrators to manage their staff, databases, organizations, and the Diagnostic Clinic. Internal staff members of FABI can submit request and deposit forms for certain culture collections as well submit samples to the Diagnostic Clinic.

The secondary intention of the web application is to provide a centralized and automated means for which organizations and users within those organizations to submit forms to the Diagnostic Clinic as well as track the progress of the samples that they submit.

The mobile application will provide a similar functionality with the exception that the mobile application will not be for the use of FABI administrators.

### 1.2.3 Product Objectives, Goals, and Benefits

The project objectives and goals are as follows:

- Provide a central platform for which administrators of FABI can manage all databases, staff, and organizations.
- Automate all information collection processes.
- Establish a system that can track samples which have been submitted to 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

- FABI - Forestry and Agricultural Biotechnology Institute
- FABI Administrator - This refers to any staff member within FABI that has access to restricted sections or features of the system.
- Internal Staff - This refers to any staff member that FABI wishes to provide access to their system from within their organization.
- Diagnostic Clinic - This refers to a section of the application which facilitates the submission, analysis, tracking, and reporting of plant matter samples submitted from industry organizations to the clinic.
- CRUD - Create, Read, Update, Delete

- 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
- Sample Owner - The owner of a sample refers to one of three types of users. These users can be a FABI staff member, an organization, or an organization member.

## 1.4 Overview

The remainder of this document serves 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.

## 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).

## 2 User Characteristics

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

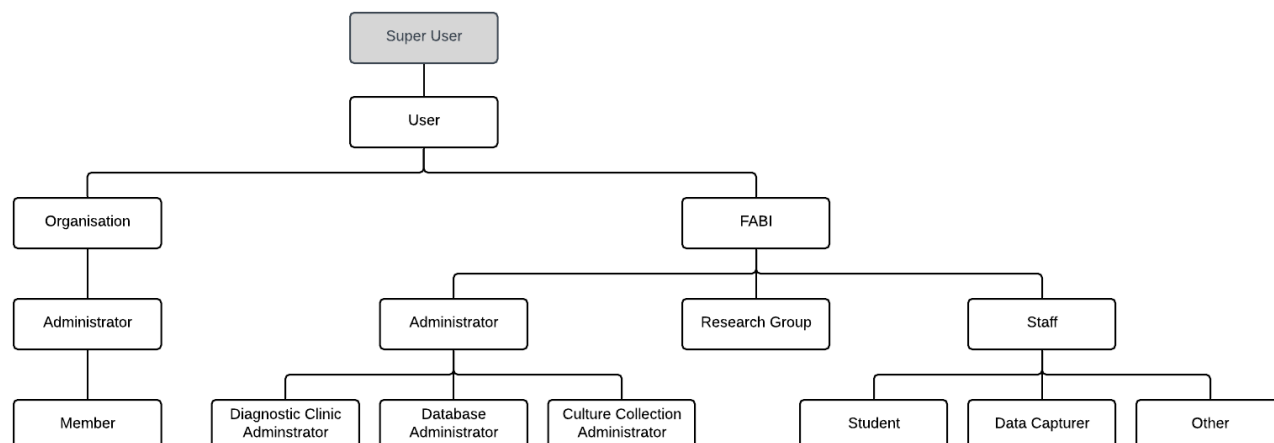


Figure 1: User Hierarchy

**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.

**User:** This group is broken down into two further sub-groups, *Organisation* users and *FABI* users.

**Organisation users,** this user group encapsulates stakeholders and foresters, who will primarily be involved with the Diagnostic Clinic service.

**FABI users** include all members of FABI, where the members' role in the system vary.

Both the *Organisation* and *FABI* users have access to the mobile application and web application.

**Organisation Administrators:** This user is the administrator of their respective organisation. This user has the ability to add a member of their organisation to the system, as well as maintain the organisation's profile details. This user also has the ability to submit a sample, as well as track all samples associated with their respective organisation.

**Organisation Member:** This user is a member of an organisation registered to the FABI system. This user is limited to submitting samples and tracking all samples they have submitted, as well as maintaining their profile details.

**FABI Administrator:** These users are further divided into different types of administrators, such as the *Database* administrator and the *Diagnostic Clinic*. Each of these administrators only having access to data and services that they are administrators of. These users may add, remove and update their respective data and services.

**Research Group:**

**FABI Staff:** These users are further divided into different types of staff, such as *Students* and *Data Capturers*. These users have very limited access to the system, as they are more concerned with the internal happening of the Diagnostic Clinic, such as culture collection form submissions. These users do not have any data modifying permissions.

## 3 Constraints

The following section provides more details about the constraints that this software solution will need to take into consideration.

### 3.1 System Constraints

Due to the time constraints of this project, the delivered solution will not be able to perform image recognition on any images of insects. This is, however, a desirable attribute that can be added in the future. The system will also not be able to provide real-time editing and assignment of the processing stages of a sample. This is a very complex feature to implement and has not been outlined as a primary point of focus. Another constraint force upon the system is that the mobile application will not be able to provide any porting functionality, and therefore a user will not be able to upload a cvs file via the mobile application

### 3.2 Environmental and Technology Constraints

#### 3.2.1 Software Constraints

Development of the mobile application is constrained to a platform that provides for the development of both an Android and an IOS application. Therefore the mobile application will be ported from the Angular web application created using Apache Cordova.

Database design is a huge aspect of this system. Many of the databases will need to be created, but all databases will be required to be created in such a way so that they are editable and maintainable. The size of the actual databases will also need to be considered.

Network design encapsulates the network to be created in order to facilitate the communication between the server, mobile application, web application, databases, and other system elements. A public domain name will be required for the web application.

#### 3.2.2 Hardware Constraints

FABI has four servers onsite. There is a possibility to use one of their servers to host the databases as well as any other system elements that will be required to be hosted. However, due to The University of Pretoria's proxy, external hosting solutions will be required. This will need to fit within the budget allocated.

The only other hardware constraints for this system include the devices to be used by the users. With regards to the mobile application, users will be required to use smartphones as the mobile application will need to be downloaded from either the Android app store or the IOS app store. The devices accessing the web application will require access to the Internet.

## 4 Domain Model

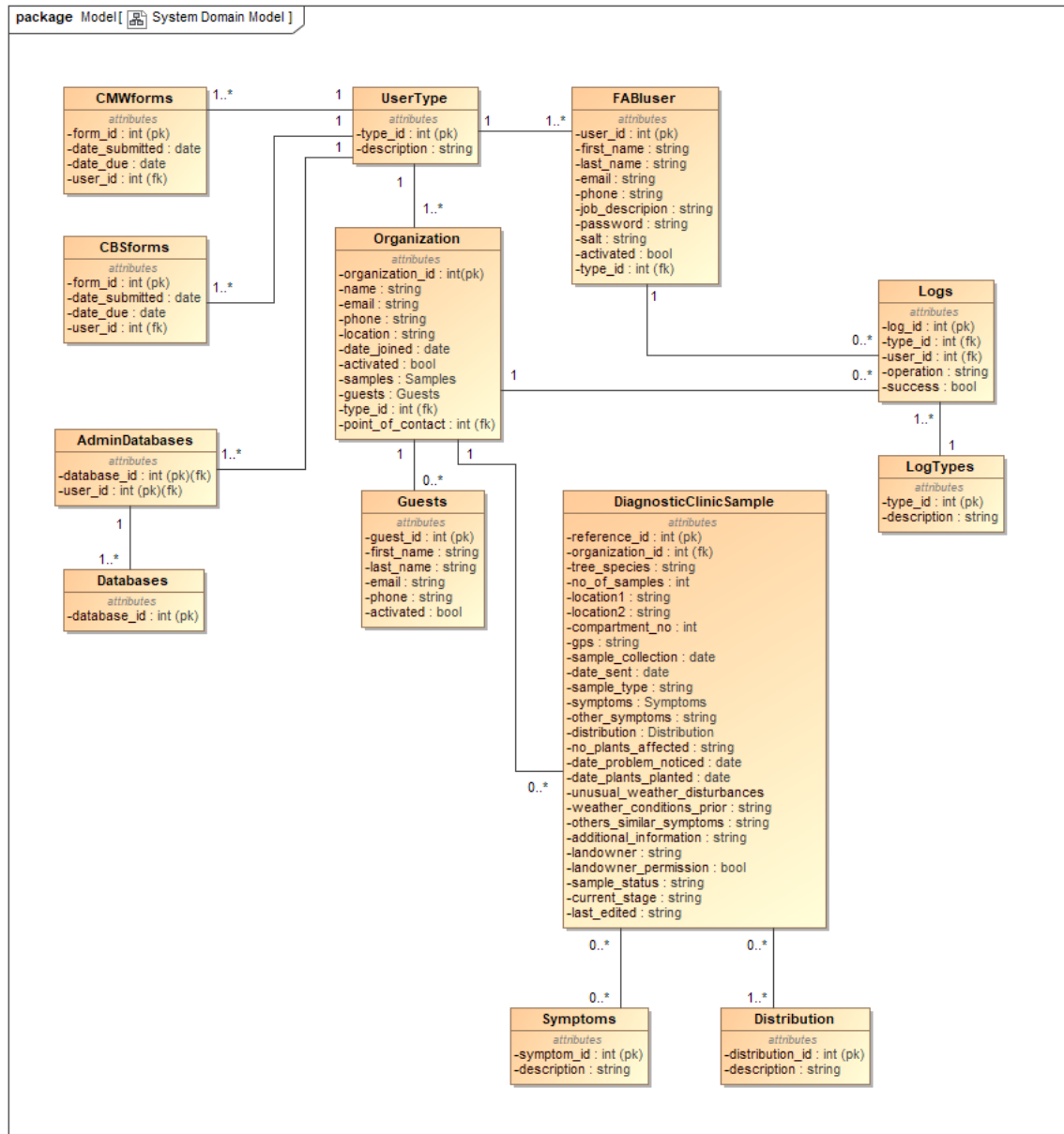


Figure 2: System Domain Model



## 5 Architectural Design

The system will be composed of a set of loosely coupled sub-systems, each built to perform a set of tasks within the overall system. To achieve this, a combination of predefined architectural patterns will be implemented.

The following architectures have been used: Object-Persistent, Microkernel, Component Bases, Client-Server, and Micro-services.

The overall system will be database centric, with multiple database structures and access control systems needing to be implemented. Databases for user management, diagnostic clinic samples as well as databases for any sample collections and other data stored by the organization will be developed. This therefore has the implication that an object-persistent architecture will be used.

The need for strict access control in terms of component access as well as database and API access has the implication that a mandatory access control system is imperative. This therefore requires a variant of a microkernel architecture. However, due to the rigidity of the microkernel architecture the system will "wrap" the microkernel architecture within a micro-service architecture. This idea will enable an access control system which covers all bases, as well as the allowances to adapt the access control should it be necessary.

The front-end systems will use a component based architecture in order to maintain separation of concerns. Strict access control between user types is to be enforced. Each type of user will be presented with a different set of components based on their user type.

The system will use a Client-Server architecture to connect front-end and back-end systems. Server side processes will be hosted on cloud based computing services. The back-end system will make use of a micro-services architecture, where separate components will provide services to front-end processes, as well as providing micro-services to other back-end components. These services will all be provided in the form of RESTful web services.

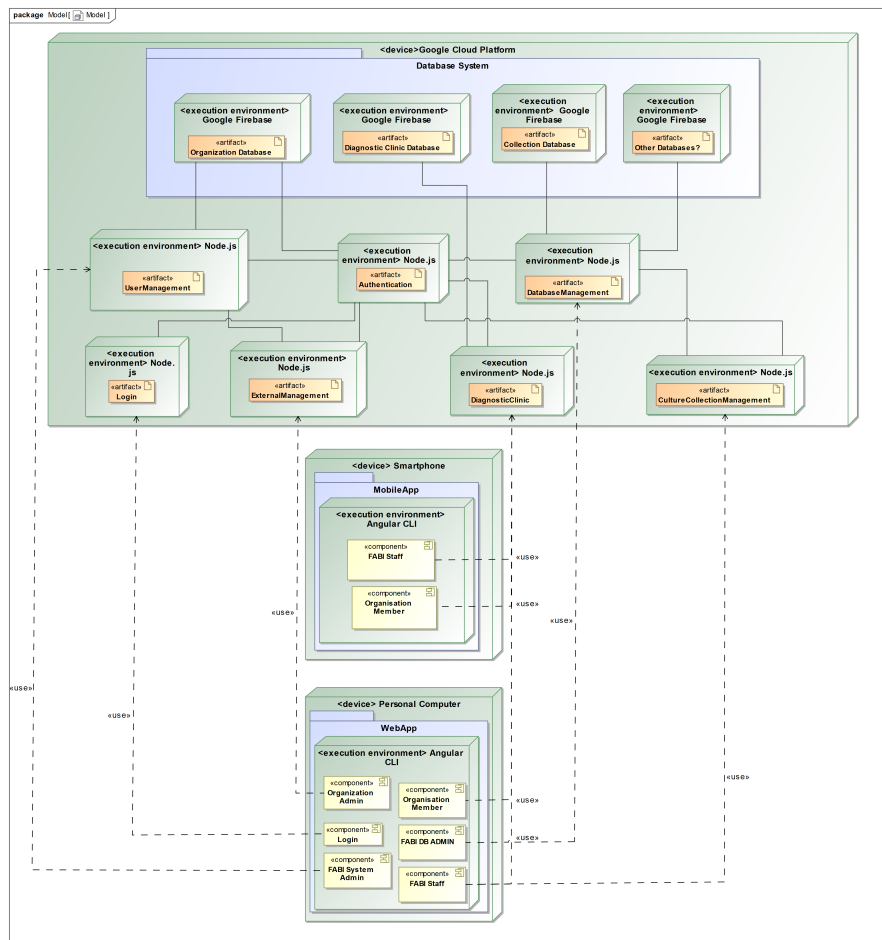


Figure 3: System Deployment Model

## 6 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.

### 6.1 Authentication Subsystem

The purpose of the Authentication Subsystem is to handle all authentication requests as well as the implementation of session control.

#### 6.1.1 Use Case



Figure 4: Authentication Subsystem Use Case Diagram

### 6.1.2 Requirements

The following requirements should be fulfilled by the Authentication Subsystem:

#### **R1. Authenticate FBI Administrator**

R1.1. The subsystem will authenticate the log in details of the FBI administrator against those details contained within the database.

R1.2. The subsystem will notify the user if their log in details are Authenticated or Not-Authenticated.

R1.3. The subsystem will create a session for the user

R1.4. The subsystem will log Authenticated or Not-Authenticated log in attempts

#### **R2. Authenticate FBI Staff**

R2.1. The subsystem will authenticate the log in details of the FBI staff members against those details contained within the database.

R2.2. The subsystem will notify the user if their log in details are Authenticated or Not-Authenticated.

R2.3. The subsystem will create a session for the user

R2.4. The subsystem will log Authenticated or Not-Authenticated log in attempts

#### **R3. Authenticate Organization Administrator**

R3.1. The subsystem will authenticate the log in details of an organization administrator against those details contained within the database.

R3.2. The subsystem will notify the user if their log in details are Authenticated or Not-Authenticated.

R3.3. The subsystem will create a session for the user

R3.4. The subsystem will log Authenticated or Not-Authenticated log in attempts

#### **R4. Authenticate Organization Member**

R4.1. The subsystem will authenticate the log in details of an organization member against those details contained within the database.

R4.2. The subsystem will notify the user if their log in details are Authenticated or Not-Authenticated.

R4.3. The subsystem will create a session for the user

R4.4. The subsystem will log Authenticated or Not-Authenticated log in attempts

#### **R5. Logout FBI Administrator**

R5.1. The subsystem will log the user out

R5.2. The subsystem will terminate the user session

**R6. Logout FBI Staff**

R6.1. The subsystem will log the user out

R6.2. The subsystem will terminate the user session

**R7. Logout Organization Administrator**

R7.1. The subsystem will log the user out

R7.2. The subsystem will terminate the user session

**R8. Logout Organization Member**

R8.1. The subsystem will log the user out

R8.2. The subsystem will terminate the user session

**R9. Terminate User Session**

R9.1. The subsystem will destroy the user's session. This will either be initiated by the user logging out or a session timeout due to inactivity

**R10. Verify User Session**

R10.1. The subsystem will check the user's session to ensure it is still valid

### 6.1.3 Subsystem Allocation

	Authenticate FABI Administrator	Logout FABI Administrator	Authenticate FABI Staff	Logout FABI Staff	Authenticate Organization Administrator	Logout Organization Administrator	Authenticate Organization Member	Logout Organization Member	Terminate User Session	Verify User Session
R1.										
R1.1.	X									
R1.2.	X									
R1.3.	X									
R1.4.	X									
R2.										
R2.1.		X								
R2.2.		X								
R2.3.		X								
R2.4.		X								
R3.										
R3.1.			X							
R3.2.			X							
R3.3.			X							
R3.4.			X							
R4.										
R4.1.				X						
R4.2.				X						
R4.3.				X						
R4.4.				X						
R5.										
R5.1.					X					
R5.2.					X					
R6.										
R6.1.						X				
R6.2.						X				
R7.										
R5.1.							X			
R5.2.							X			
R8.										
R8.1.								X		
R8.2.								X		
R9.										
R5.1.									X	
R9.										
R5.1.										X

Table 1: Authentication Subsystem Traceability Matrix

## 6.2 Staff Handler Subsystem

The purpose of the Staff Handler Subsystem is to enable FBI administrators to manage their staff according to the diagram in the user characteristics section.

### 6.2.1 Use Case

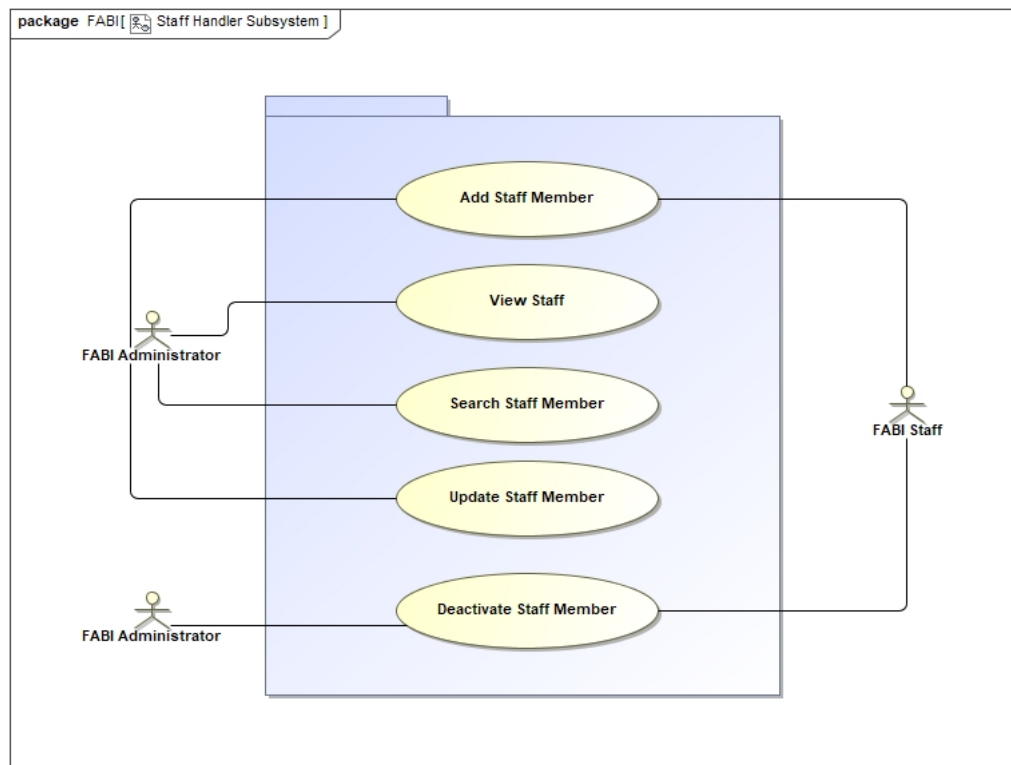


Figure 5: Staff Handler Use Case

### 6.2.2 Requirements

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

#### R1. Add Staff Member

- R1.1. The subsystem will allow an administrator to add a new staff member to the system
- R1.2. The subsystem will insert the staff member into the appropriate database.
- R1.3. The subsystem will send an email to the staff member containing a temporary password

#### R2. View Staff Compliment

- R2.1. The subsystem will display all staff members currently active within FBI

### R3. Search Staff Member

R3.1. The subsystem will allow an administrator to search for a particular staff member

R3.2. The subsystem will display the appropriate information regarding the staff member

R3.3. The subsystem will display an appropriate message if the user is not found

### R4. Update Staff Member

R4.1. The subsystem will allow an administrator to edit information regarding the staff member

R4.2. The subsystem will update the appropriate database

### R5. Deactivate Staff Member

R5.1. The subsystem will allow an administrator to deactivate a staff member

R5.2 The subsystem will update the status of the staff member in the database to deactivated, rather than removing the user from the database

R5.3. The subsystem will subsequently not allow this user to log in

R5.4 The subsystem will notify the respective staff member of their account deactivation

#### 6.2.3 Subsystem Allocation

	Add Staff Member	View Staff	Search Staff Member	Update Staff Member	Deactivate Staff Member
R1.					
R1.1.	X				
R1.2.	X				
R1.3.	X				
R2.					
R2.1.		X			
R3.					
R3.1.			X		
R3.2.			X		
R3.3.			X		
R4.					
R4.1.				X	
R4.2.				X	
R5.					
R5.1.					X
R5.1.					X
R5.3.					X
R5.4.					X

Table 2: Staff Handler Subsystem Traceability Matrix

## 6.3 Diagnostic Clinic Handler Subsystem

The purpose of the Diagnostic Clinic Handler Subsystem is to handle the maintenance of the Diagnostic Clinic in terms of the processing cycle that sample's undergo.

### 6.3.1 Use Case

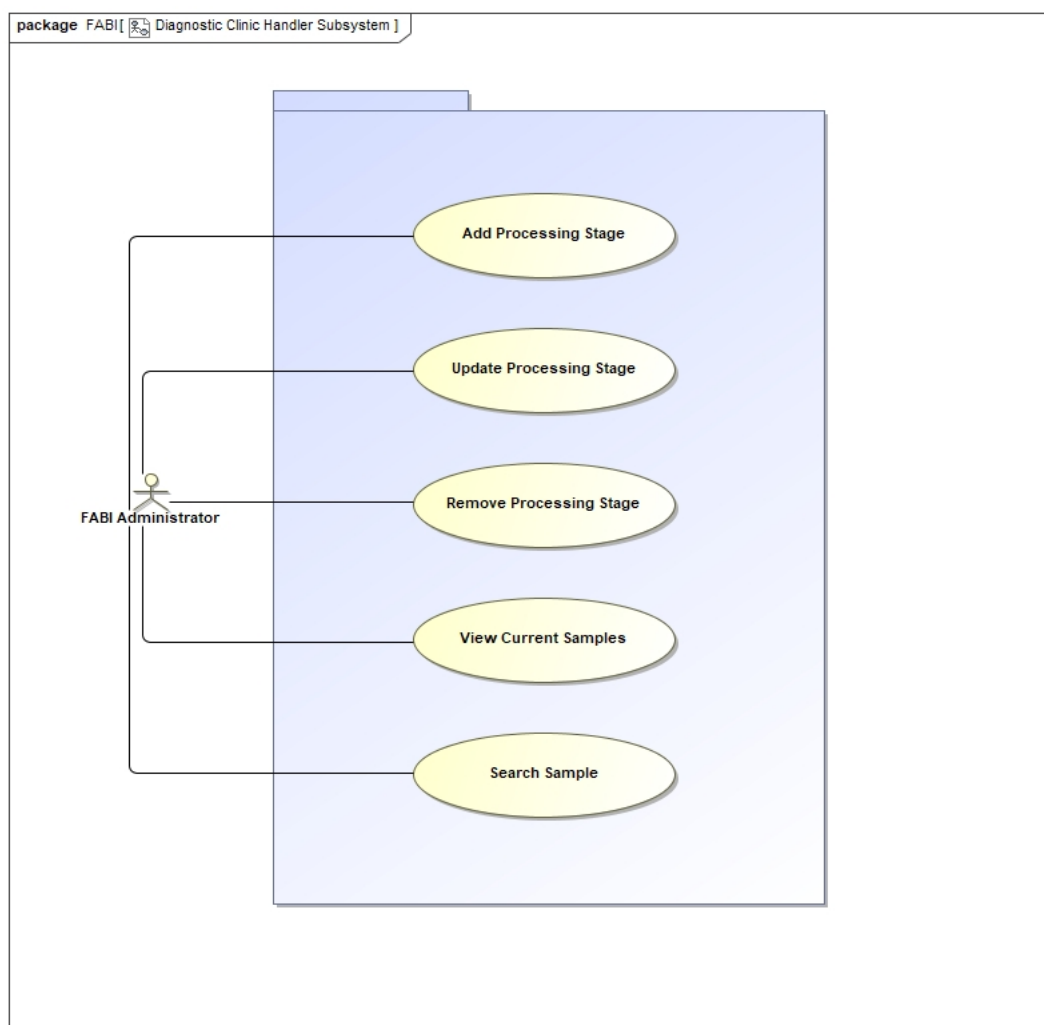


Figure 6: Diagnostic Clinic Handler Subsystem Use Case Diagram

### 6.3.2 Requirements

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

#### R1. Add Processing Stage

R1.1. The subsystem will allow the administrator of the Diagnostic Clinic to add a new processing stage to the clinic's diagnostic process.

#### R2. Update Processing Stage

R2.1. The subsystem will allow the administrator of the Diagnostic Clinic to update a processing stage's details



### R3. Remove Processing Stage

R3.1. The subsystem will allow the administrator of the Diagnostic Clinic to remove a processing stage from the clinic's diagnostic process.

### R4. View Current Samples

R4.1. The subsystem will display all the samples that the clinic is currently processing

### R5. Search Sample

R5.1. The subsystem will allow an administrator to search for a particular sample

R5.2. The subsystem will display the appropriate information regarding the sample

R5.3. The subsystem will display an appropriate message if the sample is not found

#### 6.3.3 Subsystem Allocation

	Add Processing Stage	Update Processing Stage	Remove Processing Stage	View Current Samples	Search Samples
R1.					
R1.1.	X				
R2.					
R2.1.		X			
R3.					
R3.1.			X		
R4.					
R4.1.				X	
R5.					
R5.1.					X
R5.2.					X
R5.3.					X

Table 3: Diagnostic Clinic Handler Subsystem Traceability Matrix

## 6.4 Database Handler Subsystem

The purpose of the Database Handler Subsystem is to enable FABI administrators to manage their databases from a central point.

### 6.4.1 Use Case

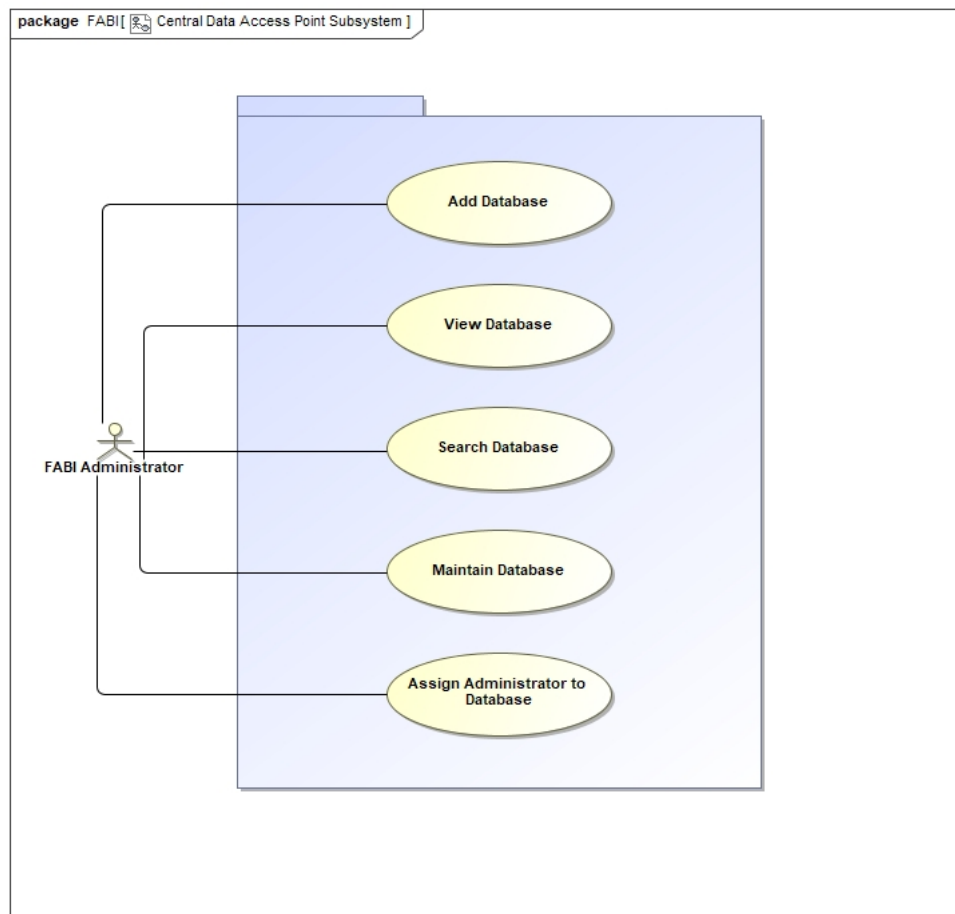


Figure 7: Database Handler Subsystem User Case Diagram

### 6.4.2 Requirements

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

#### R1. Add Database

R1.1 The subsystem will allow administrators to add a new database manually.

R1.2 The subsystem will allow administrators to add a new database via the porting subsystem.

#### R2. View Databases

R2.1. The subsystem will display all databases that FABI maintains

### R3. Search Databases

R3.1. The subsystem will allow an administrator to search for a particular database

R3.2. The subsystem will display the appropriate information regarding the database

R3.3. The subsystem will display an appropriate message if the database is not found

### R4. Maintain Database

R4.1. The subsystem will allow an administrator to select a database.

R4.2. The subsystem will allow an administrator add,update, or remove entries within the database.

R4.3. The subsystem will display confirmation dialogues for the administrator regarding any changes that occur.

### R5. Assign Administrator To Database

R5.1. The subsystem will allow an administrator to select a database.

R5.2. The subsystem will allow an administrator to assign a FBI member of staff to the selected database

R5.3. The subsystem will send an email to the member of staff informing them of what database they are now an administrator of

#### 6.4.3 Subsystem Allocation

	Add Database	View Database	Search Database	Maintain Database	Assign Administrator to Database
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	
R4.2.				X	
R4.3.				X	
R5.					
R5.1.					X
R5.2.					X
R5.3.					X

Table 4: Database Handler Subsystem Traceability Matrix

## 6.5 Culture Collection Handler Subsystem

This purpose of the Culture Collection Handler Subsystem is that it will manage and store all forms submitted intended for the culture collection.

### 6.5.1 Use Case

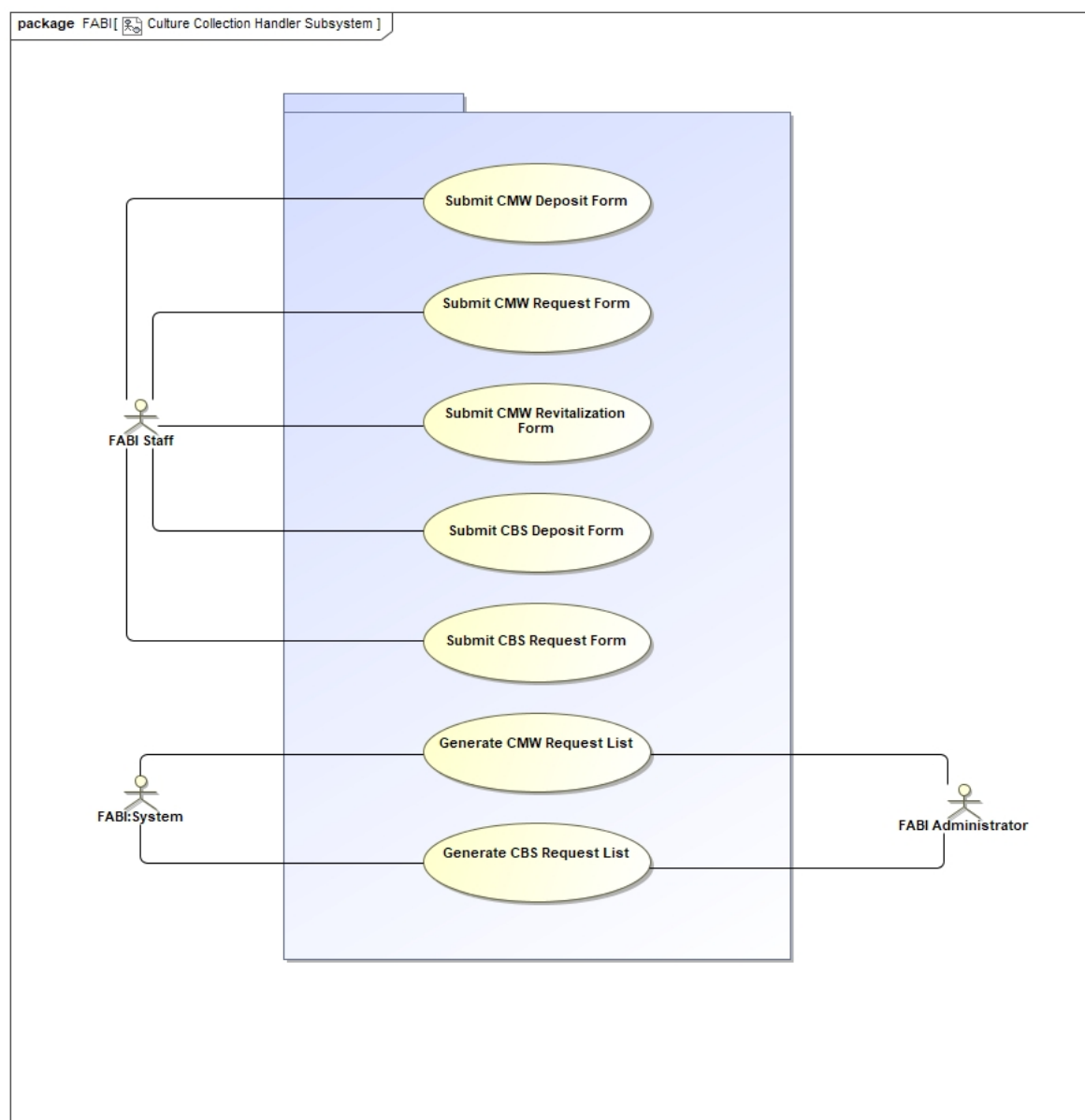


Figure 8: Culture Collection Use Case Diagram

### 6.5.2 Requirements

The following functional requirements should be fulfilled by the Culture Collection Handler Subsystem:

#### R1. Submit CMW Request Form

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

**R2. Submit CMW Deposit Form**

R2.1. The subsystem will allow FBI staff members to submit a CMW deposit form to the CMW culture collection.

**R3. Submit CMW Revitalization Form**

R3.1. The subsystem will allow FBI staff members to submit a CMW revitalization form to the CMW culture collection.

**41. Submit CBS Request Form**

R4.1. The subsystem will allow FBI staff members to submit a CBS request form to the CBS culture collection.

**R5. Submit CBS Deposit Form**

R5.1. The subsystem will allow FBI staff members to submit a CBS deposit form to the CBS culture collection.

**R6. Generate CMW Request List**

R6.1. The subsystem will generate a list of all CMW requests that need to be sent to the FBI culture collection .

**R7. Generate CBS Request List**

R7.1. The subsystem will generate a list of all CBS requests that need to be sent to the international CBS culture collection.

**6.5.3 Subsystem Allocation**

	Submit CMW Deposit Form	Submit CMW Request Form	Submit CMW Revitalization Form	Submit CBS Deposit Form	Submit CBS Request Form	Generate CMW Request List	Generate CBS Request List
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 5: Culture Collection Handler Subsystem Traceability Matrix

## 6.6 Organization Handler Subsystem

The purpose of the Organization Handler Subsystem is to enable FBI administrators to manage the organizations that are able to access the system. This therefore manages the associated organizations and their details.

### 6.6.1 Use Case

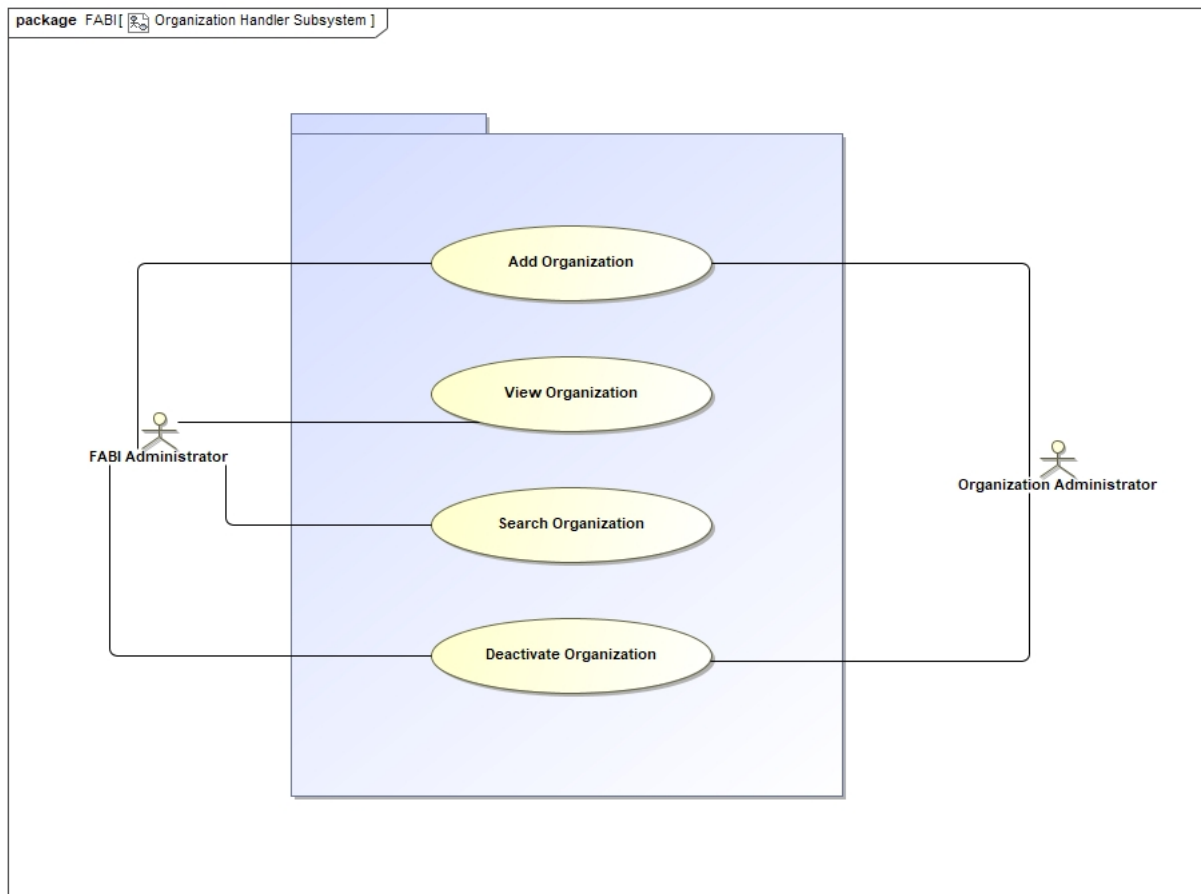


Figure 9: Organization Handler Use Case Diagram

### 6.6.2 Requirements

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

#### R1. Add Organization

- R1.1. The subsystem will allow a FBI administrator to add a new organization
- R1.2. The subsystem will insert this organization into the appropriate database
- R1.3. The subsystem will email the administrator of the organization a temporary password

#### R2. View Organization

- R2.1 The subsystem will allow the administrator to view all associated organizations

### R3. Search Organization

R3.1. The subsystem will allow an administrator to search for a particular organization

R3.2. The subsystem will display the appropriate information regarding the organization

R3.3. The subsystem will display an appropriate message if the organization is not found

### R4. Deactivate Organization

R4.1. The subsystem will allow an administrator to deactivate an organization

R4.2. The subsystem will subsequently deactivate all members that are apart of that organization

R4.3. The subsystem will subsequently not allow the administrator of the organization or any associated members to log in (\*access the system?\*)

#### 6.6.3 Subsystem Allocation

	Add Organization	View Organization	Search Organization	Deactivate Organization
R1.				
R1.1.	X			
R1.2.	X			
R1.3.	X			
R2.				
R2.1.		X		
R3.				
R3.1.			X	
R3.2.			X	
R3.3.			X	
R4.				
R4.1.				X
R4.2.				X
R4.3.				X

Table 6: Organization Handler Subsystem Traceability Matrix

## 6.7 Organization Subsystem

The purpose of the Organization Subsystem is to enable associated organizations to manage the members that they have as well as the overall ability to submit and monitor all samples submitted by a particular organization and its members.

### 6.7.1 Use Case

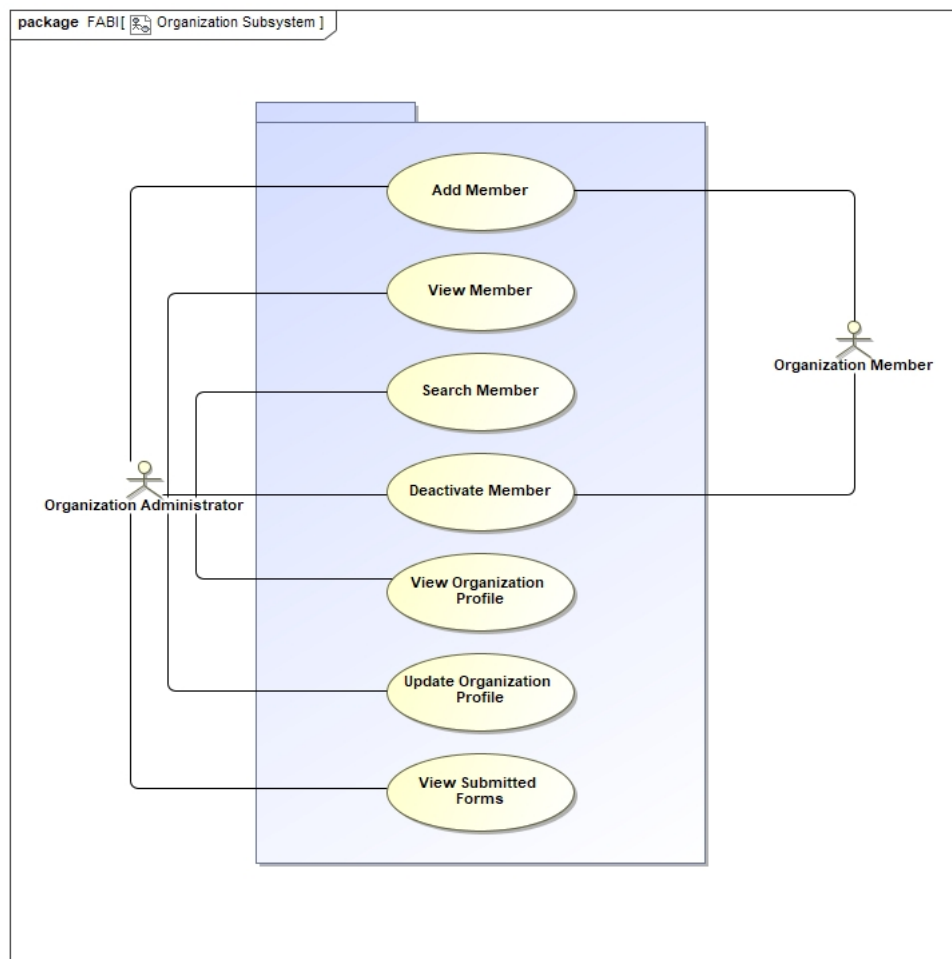


Figure 10: Organization Use Case Diagram

### 6.7.2 Requirements

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

#### R1. Add Member

- R1.1. The subsystem will allow an administrator of the organization to add a new member
- R1.2. The subsystem will insert this member into the appropriate database
- R1.3. The subsystem will create an association between the registered member and the organization
- R1.4. The subsystem will email the member of the organization a temporary password.



**R2. View Member**

R2.1 The subsystem will allow the administrator to view all associated organization members.

**R3. Search Member**

R3.1. The subsystem will allow an administrator to search for a particular member

R3.2. The subsystem will display the appropriate information regarding the member

R3.3. The subsystem will display an appropriate message if the member is not found.

**R4. Deactivate Member**

R4.1. The subsystem will allow an administrator to deactivate a member

R4.3. The subsystem will subsequently not allow the member of that organization to log in.

**R5. View Organization Profile**

R5.1. The subsystem will display all information associated with the organization.

**R6. Update Organization Profile**

R6.1. The subsystem will allow the administrator of the organization to edit the organization's details

**R7. View Submitted Samples**

R7.1. The subsystem will display all submitted samples associated with a particular organization.

### 6.7.3 Subsystem Allocation

	Add Member	View Member	Search Member	Deactivate Member	View Organization Profile	Update Organization Profile	View Submitted Samples
R1.							
R1.1.	X						
R1.2.	X						
R1.3.	X						
R1.4.	X						
R2.							
R2.1.		X					
R3.							
R3.1.			X				
R3.2.			X				
R3.3.			X				
R4.							
R4.1.				X			
R4.2.				X			
R5.							
R5.1.					X		
R6.							
R6.1.						X	
R7.							
R7.1.							X

Table 7: Organization Subsystem Traceability Matrix

## 6.8 Organization Member Subsystem

The purpose of the Organization Member Subsystem is to enable organization members to maintain their profiles and sample submissions.

### 6.8.1 Use Case

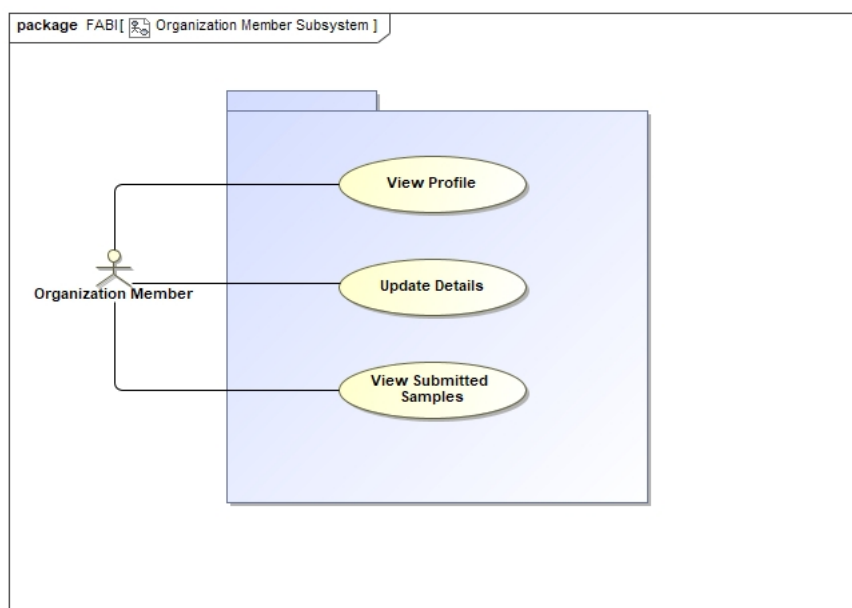


Figure 11: Organization Member Use Case Diagram

### 6.8.2 Requirements

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

#### R1. View Profile

R1.1. The subsystem will allow the member to view all their details

#### R2. Update Profile

R2.1 The subsystem will allow the member to edit their details

#### R3. View Submitted Samples

R3.1. The subsystem will display all submitted samples associated with a particular member.

### 6.8.3 Subsystem Allocation

	View Profile	Update Profile	View Submitted Samples
R1.			
R1.1.	X		
R2.			
R2.1.		X	
R3.			
R3.1.			X

Table 8: Organization Member Subsystem Traceability Matrix

## 6.9 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.

### 6.9.1 Use Case

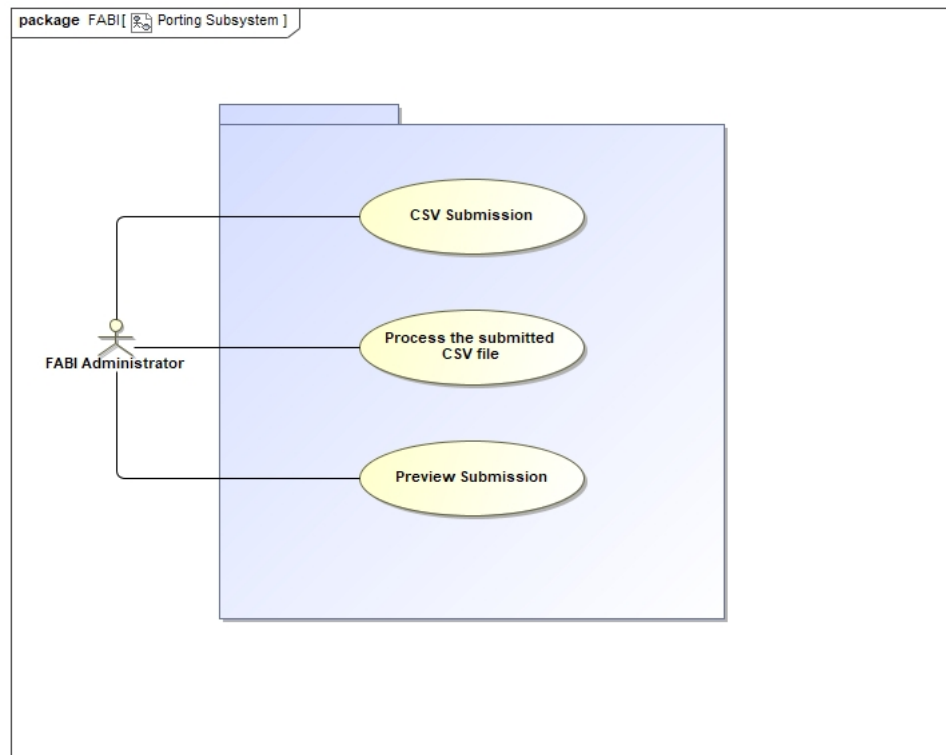


Figure 12: Porting Subsystem Use Case Diagram

### 6.9.2 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.

### 6.9.3 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 9: Porting Subsystem Traceability Matrix

## 6.10 Sample Processing Subsystem

The purpose of the Sample Processing Subsystem is to enable the monitoring and management of a sample during the processing cycle.

### 6.10.1 Use Case

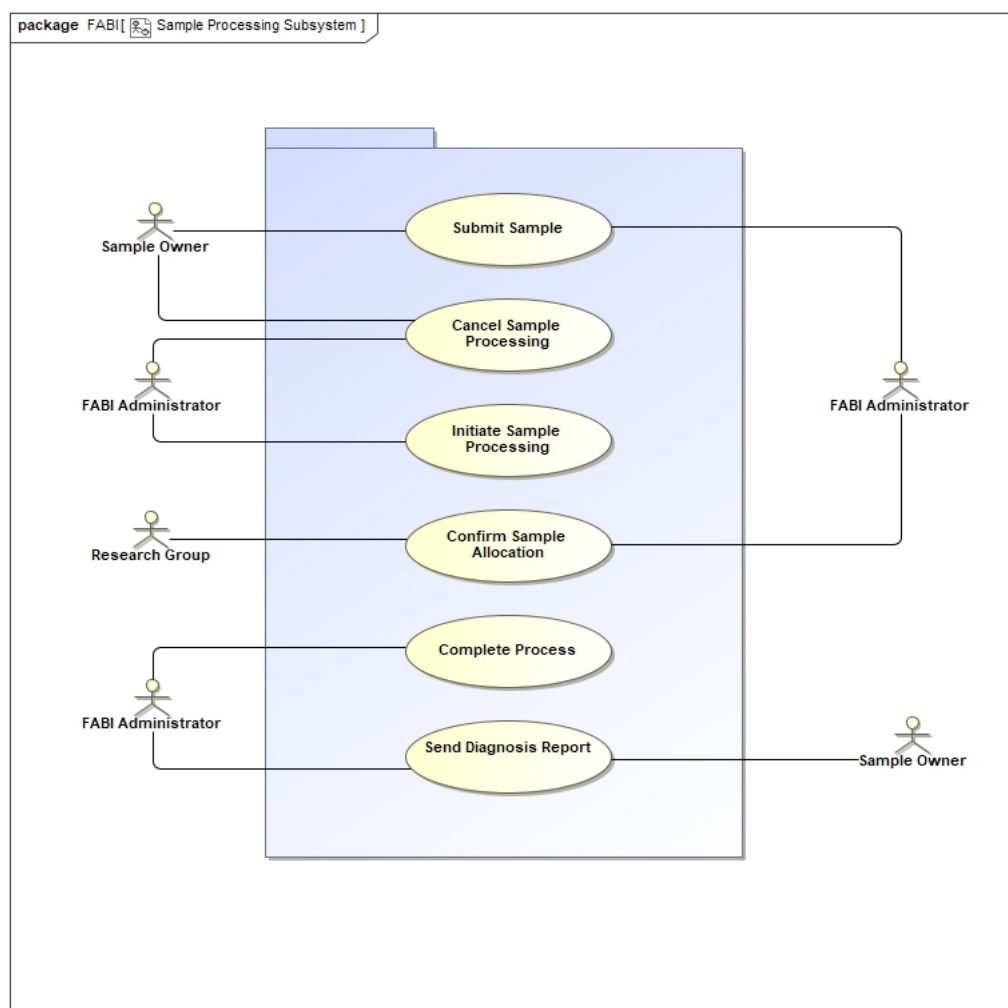


Figure 13: Sample Processing Subsystem Use Case Diagram

### 6.10.2 Requirements

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

#### R1. Submit Sample

- R1.1. The subsystem will allow the user to submit a sample submission form, as well as the physical sample, which will be done manually
- R1.2. The subsystem will validate the form before submission is successful
- R1.3. The subsystem will attach known information to the form that the user did not explicitly supply
- R1.4. The subsystem will notify the administrator of the Diagnostic Clinic of a new sample submission
- R1.5. The subsystem will update the sample status in the database to "pending"

**R2. Cancel Sample Processing**

R2.1. The subsystem will allow the administrator of the Diagnostic Clinic to cancel a sample processing

**R3. Initiate Sample Processing**

R3.1. The subsystem will allow the administrator of the Diagnostic Clinic to allocate a sample submission to a particular research group

R3.2. The subsystem will send a confirmation email to the research group

R3.3. The subsystem will update the sample status in the database to "initiated"

**R4. Confirm Sample Allocation**

R4.1. The research group will confirm the sample allocation

R4.2. The subsystem will update the sample status in the database to "analysis"

**R5. Complete Process**

R5.1. The subsystem will allow the Diagnostic Clinic administrator to mark a particular sample as completed

R5.2. The subsystem will update the sample status in the database to "completed"

**R6. Send Diagnosis Report**

R6.1. The subsystem will allow the Diagnostic Clinic administrator to upload a report

R6.2. The subsystem will send the report to the sample owner associated to that particular sample.



### 6.10.3 Subsystem Allocation

	Submit Sample	Cancel Sample Processing	Initiate Sample Processing	Confirm Sample Allocation	Complete Process Processing	Send Diagnosis Report Processing
R1.						
R1.1.	X					
R1.2.	X					
R1.3.	X					
R1.4.	X					
R1.5.	X					
R2.						
R2.1.		X				
R3.						
R3.1.			X			
R3.2.			X			
R3.3.			X			
R4.						
R4.1.				X		
R4.2.				X		
R5.						
R5.1.					X	
R5.2.					X	
R6.						
R6.1.						X
R6.2.						X

Table 10: Sample Processing Subsystem Traceability Matrix

## 6.11 Notification Subsystem

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

### 6.11.1 Use Case

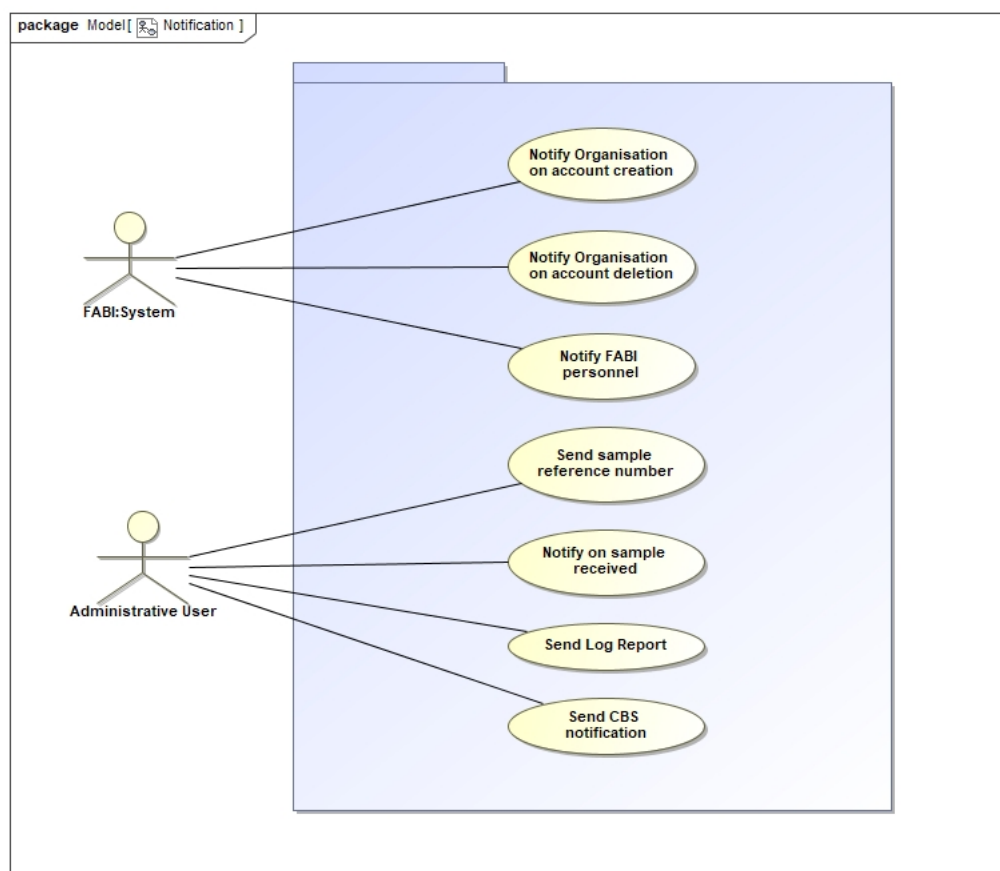


Figure 14: Notification Use Case Diagram

### 6.11.2 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 FAB: personnel

R3.1. Send notification to FAB: 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 FBI.

**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.

**6.11.3 Subsystem Allocation**

	Notify organization on account creation	Notify organization on account deletion	Notify FBI personnel	Send sample reference number	Notify on 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 11: Notification Subsystem Traceability Matrix

## 6.12 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).

### 6.12.1 Use Case

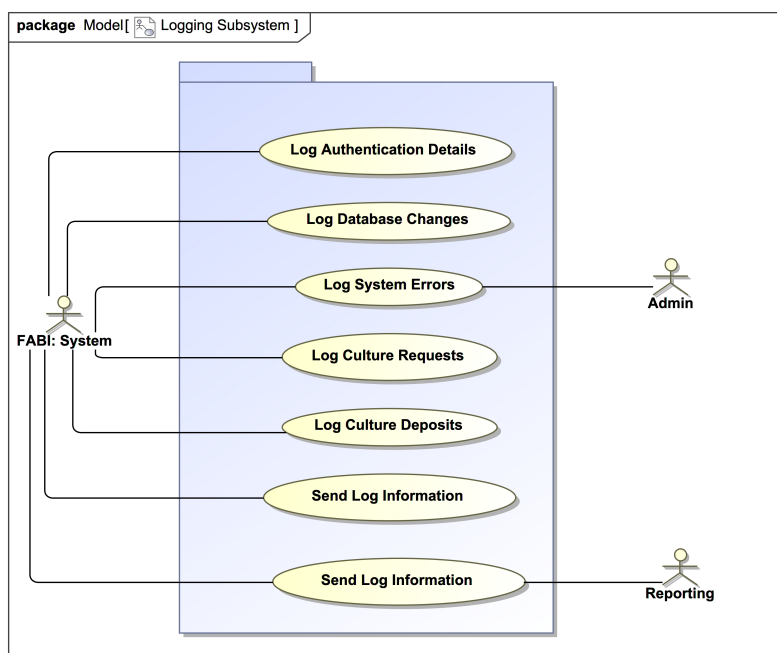


Figure 15: Logging Use Case Diagram

### 6.12.2 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

**6.12.3 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 12: Logging Subsystem Traceability Matrix

## 6.13 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.

### 6.13.1 Use Case

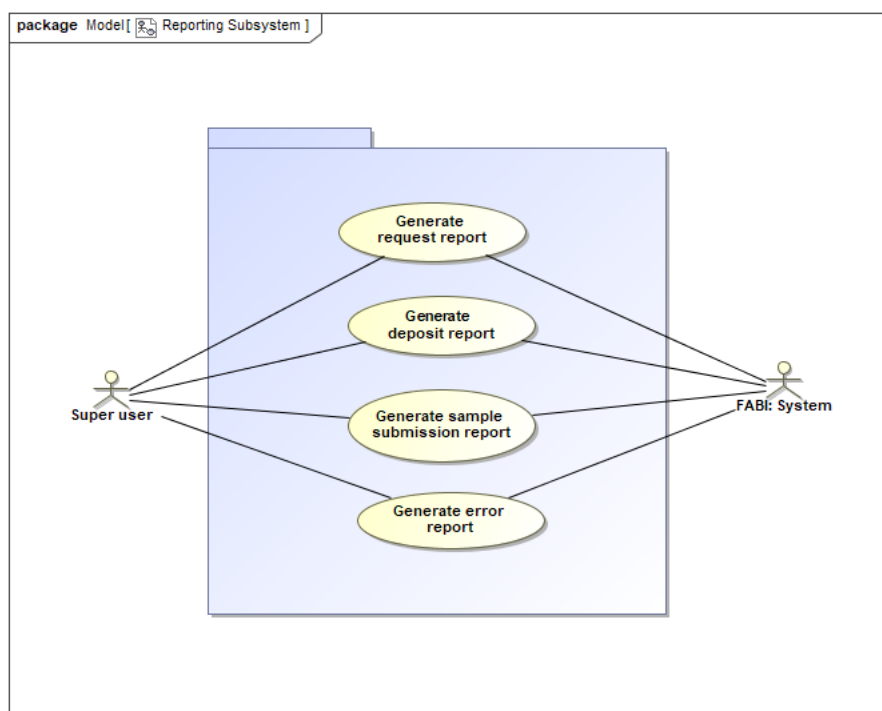


Figure 16: Reporting Subsystem Use Case Diagram

### 6.13.2 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.

**6.13.3 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 13: Reporting Subsystem Traceability Matrix

## 7 Quality Requirements

### 7.1 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.

### 7.2 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.

### 7.3 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.

### 7.4 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.

### 7.5 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.

### 7.6 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.



## 7.7 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.

## 7.8 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).  $MTD = RTO + WRT$

## 7.9 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.

## 7.10 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.

## 7.11 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.

## 7.12 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.

## 8 Technologies

- Node.js for the API and local Web App server since it is primarily used for non-blocking, event-driven services.
- Google Firestore which is a free, fast and fully managed NoSQL document database.
- Google Cloud hosting platform which is free and has high uptime, reliability and control over data.
- Angular CLI for Web App development. Angular uses a component-based architecture which provides reusability and maintainability and is unit-test friendly.
- Apache Cordova for porting to Android and iOS for compatibility, code efficiency and cost reduction.
- Travis CI for integration testing.
- Jasmine as a behavior-driven development framework for JavaScript testing.
- Karma as the task runner for Angular unit testing since it is highly configurable and easily debugged.



	Staff Handler Subsystem	Authentication Subsystem	Diagnostic Clinic Handler Subsystem	Central Data Access Point Subsystem	Culture Collection Handler Subsystem	Organization Handler Subsystem	Organization Subsystem	Organization Member Subsystem	Porting Subsystem	Sample Processing Subsystem	Notification Subsystem	Logging Subsystem	Reporting Subsystem
Culture Collection Form Handler Subsystem													
R1.					X								
R2.					X								
R3.					X								
R4.					X								
R5.					X								
R6.					X								
R7.					X								
Organization Handler Subsystem													
R1.						X							
R2.						X							
R3.						X							
R4.						X							
Organization Subsystem													
R1.							X						
R2.							X						
R3.							X						
R4.							X						
R5.							X						
R6.							X						
R7.							X						
Organization Member Subsystem													
R1.								X					
R2.								X					
R3.								X					
Porting Subsystem													
R1.									X				
R2.									X				
R3.									X				

	Staff Handler Subsystem	Authentication Subsystem	Diagnostic Clinic Handler Subsystem	Central Data Access Point Subsystem	Culture Collection Handler Subsystem	Organization Handler Subsystem	Organization Subsystem	Organization Member Subsystem	Porting Subsystem	Sample Processing Subsystem	Notification Subsystem	Logging Subsystem	Reporting Subsystem
Sample Processing Subsystem													
R1.										X			
R2.										X			
R3.										X			
R4.										X			
R5.										X			
R6.										X			
Notification Subsystem													
R1.											X		
R2.											X		
R3.											X		
R4.											X		
R5.											X		
R6.											X		
R7.											X		
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 14: System Traceability Matrix

