**Software requirement specification**

**for**

**LocAdoc**

*a location based document locking application*

Version 0.1

11th August 2016

**Prepared by: Abhi Jay Krishnan**

**Kim Hyeocheol**

**Rivaldo Erawan**

**Durrah Afshan**

# Table of Contents

[Table of Contents i](#_Toc443216140)

[Revision History iv](#_Toc443216141)

[1. Introduction 1](#_Toc443216142)

[1.1 Purpose 1](#_Toc443216143)

[1.2 Document Conventions 1](#_Toc443216144)

[1.3 Intended Audience and Reading Suggestions 1](#_Toc443216145)

[1.4 Project Scope 2](#_Toc443216146)

[1.5 References 2](#_Toc443216147)

[2. Overall Description 3](#_Toc443216148)

[2.1 Product Perspective 3](#_Toc443216149)

[2.2 Product Features 3](#_Toc443216150)

[2.3 User Classes and Characteristics 4](#_Toc443216151)

[2.4 Operating Environment 4](#_Toc443216152)

[2.5 Design and Implementation Constraints 4](#_Toc443216153)

[2.6 User Documentation 4](#_Toc443216154)

[2.7 Assumptions and Dependencies 4](#_Toc443216155)

[3. System Features 5](#_Toc443216156)

[3.1 Log in::REQ1 5](#_Toc443216157)

[3.2 Member Menu::REQ2 5](#_Toc443216158)

[3.3 Search Facility::REQ3 5](#_Toc443216159)

[3.4 Normal Search::REQ4 6](#_Toc443216160)

[3.5 Advanced Search::REQ5 6](#_Toc443216161)

[3.6 Sort Result::REQ6 6](#_Toc443216162)

[3.7 Booking Facility::REQ7 7](#_Toc443216163)

[3.8 Cancel Booking::REQ8 7](#_Toc443216164)

[3.9 Edit Profile::REQ9 7](#_Toc443216165)

[3.10 Request Membership Ranking Upgrade::REQ10 8](#_Toc443216166)

[3.11 Cancel Request::REQ11 8](#_Toc443216167)

[3.12 View Notification::REQ12 9](#_Toc443216168)

[3.13 View Bookings::REQ13 9](#_Toc443216169)

[3.14 Manager Menu::REQ14 9](#_Toc443216170)

[3.15 Create Facilities::REQ15 10](#_Toc443216171)

[3.16 Delete Facilities::REQ16 10](#_Toc443216172)

[3.17 Update Facilities::REQ17 10](#_Toc443216173)

[3.18 Set Charging Rates::REQ18 11](#_Toc443216174)

[3.19 Change Other Details::REQ19 11](#_Toc443216175)

[3.20 Set Membership Fees::REQ20 11](#_Toc443216176)

[3.21 Set Membership Priorities::REQ21 12](#_Toc443216177)

[3.22 Set Membership Access Rights::REQ22 12](#_Toc443216178)

[3.23 Set Membership Discount::REQ23 13](#_Toc443216179)

[3.24 Facility Usage Report::REQ24 13](#_Toc443216180)

[3.25 Rental Behavior Report::REQ25 14](#_Toc443216181)

[3.26 Discount Rates Report::REQ26 14](#_Toc443216182)

[3.27 Update Manager Details::REQ27 14](#_Toc443216183)

[4. External Interface Requirements 16](#_Toc443216184)

[4.1 User Interfaces 16](#_Toc443216185)

[4.2 Hardware Interfaces 16](#_Toc443216186)

[4.3 Software Interfaces 16](#_Toc443216187)

[4.4 Communications Interfaces 16](#_Toc443216188)

[5. Other Nonfunctional Requirements 17](#_Toc443216189)

[5.1 Performance Requirements 17](#_Toc443216190)

[5.2 Safety Requirements 17](#_Toc443216191)

[5.3 Security Requirements 17](#_Toc443216192)

[5.4 Software Quality Attributes 17](#_Toc443216193)

[Appendix A: Glossary 18](#_Toc443216194)

[Appendix B: Analysis Models 20](#_Toc443216195)

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# 1. Introduction

# 1.1 Purpose

The purpose of this Software Requirements Specification (SRS) document is to specify the requirements and to give detailed description of the functionalities of the locAdoc a location based document locking application. This document will cover each features of the system, software requirements, and the targeted audience.

## 1.2 Document Conventions

Main Section Titles

* Font : Calibri Light
* Face : Bold
* Size : 16

Sub Section Titles

* Font : Calibri Light
* Face : Bold
* Size : 14

Other Text Explanations

* Font : Arial
* Face : Normal
* Size : 12

## 1.3 Intended Audience and Reading Suggestions

This document is intended for all individuals taking part in the locAdoc application development, such as developers, project managers, users, supervisor and testers.

Readers who are interested in the overview of the project should concentrate on Part 1 which will give an introduction of the project. Part 2 of the document describes a brief overview of the project in each aspect.

Readers who want to understand the project as a whole should focus on Part 3 which provides the features of the project in detail. Part 4 will give the visualization of the project as well as the hardware, software, and communication requirements of the project.

Readers interested in the nonfunctional requirements of the project should read Part 5, which gives information about the performance, safety, security, and other attributes.

## 1.4 Project Scope

This project aim to provide user a way to store confidential documents in mobile devices and access it only in the area he/she find it is safe. By including two factor protections, one being password (what the user knows) and second being the location (where the user is currently), we will be able to provide a better solution compared to the applications currently in the market (based on market survey).

These are few ways a document in a mobile device may be compromised: -

* The documents stored in mobile device may end up in the wrong hands if the device itself is stolen.
* The user may lend the device to someone who intern may wish to gain access to these documents.
* The documents may be accessed remotely by penetrating device through network.

Our solution aim to provide a secure vault for document storage so the it does not get into wrong hands even if the device is compromised. The solution also provide a secure backup cloud storage with double layer encryption one by the app itself and one by Amazon server.

## 1.5 References

SRS Template by Karl E. Wiegers

http://users.encs.concordia.ca/~eshihab/teaching/slides/srs\_template\_sep14.pdf

# 2. Overall Description

## 2.1 Product Perspective

LocAdoc (CCFBS) is a new, self-contained application running on android platform written in Java. It is intended in providing the user a safe place to store his pdf files and view it at an area he feels secured. The system uses a client cloud architecture where the client is the application running on a mobile device and it makes use of cloud services provided by amazon web services.

## 2.2 Product Features

The features of this system are:

***PDF Viewer***

A PDF viewer for user to view his documents in the vault. The pdf viewer will only be accessible after the user has been authenticated and if the user is within the radius of the location stored in the database. The pdf viewer will close when the user moves out of this zone. The file that the user wishes to see will be the only one that will be decrypted. The rest will remain as cipher text even when the user is in authorised area. This pdf viewer will help the user to be more productive by having the ability to access sensitive document while moving within the secure location.

***Deleting files***

The user has the option to delete the files that are not needed and these files will also be deleted from the backup.

***Setting preferred locational radius***

Once the user adds a new file he can set the radius he wishes with small radius being more secure and larger radius being more convenient. The files will be grouped based on the location and the user can choose the area if there is an overlap.

***Less clustered interface***

The user will be only able to view the files that was saved to a location making file accessing, pleasant and less tedious.

***Import files***

The user will be able to import a new file from the local file directory and secure it through encryption. The original file can be deleted to prevent adversary from viewing it.

***Secure cloud storage***

The data will be safely stored in the central database with additional layer of encryption by the cloud infrastructure provider.

These are the key features that will be included in the application. Further enhancements such as support for more file types will be added if these basic requirements are met.

## 2.3 User Classes and Characteristics

**Physical Actors:**

* **Mobile users:** The user who uses the system and make use of the services provided by the application.

**System Actors:**

* **DynamoDB:** A NoSQL database service provided by Amazon web services(AWS) and this service will be used to create a central application database.
* **AWS Cognito:** A user password authentication service provided by Amazon and will be used to authenticate user based on a central user pool.
* **AWS S3:** A central file storage facility provided by the AWS will be used to backup user data.

## 2.4 Operating Environment

This system only operates in Android Operating System.

## 2.5 Design and Implementation Constraints

The main constraint of this program is that the interface will be console based, therefore it might be less user friendly. It only works on Linux environment and the available language is English, so it cannot be used by some users, especially those who do not understand the English language and not familiar with the Linux environment.

## 2.6 User Documentation

The user can use the help menu in the system to understand the interfaces more.

## 2.7 Assumptions and Dependencies

* Each member must have a UserID and password.
* The system may run online (via the Internet).
* Facilities and Members are fully observed by the Club Manager.

3. System Features

All these features are detailed descriptions that are stated on 2.2.

## 3.1 Log in::REQ1

3.1.1 Description and Priority

To allow club members or club managers to login.

3.1.2 Stimulus/Response Sequences

Step 1: System prints out login screen.

Step 2: Actor fills in the account ID and password.

Step 3: System validates the data.

Step 4: System processes the data.

Alternate Step 4: If the data is invalid, an error message is displayed and the actor is sent back to the form to reenter the login data correctly.

Step 5: System then shows the actor to the actor’s menu screen if the login is successful.

* + 1. Functional Requirements

Connected to the server

## 3.2 Member Menu::REQ2

3.2.1 Description and Priority

To access the menu screen for club members to choose the subsystems.

3.2.2 Stimulus/Response Sequences

Step 1: System prints out the menu screen for members.

Step 2: Club Member chooses the subsystem from the list of subsystems in the menu.

Step 3: System processes the choice.

Alternate step 3: If the choice is invalid an error message is printed and the user is redirected to the main menu.

Step 4: System prints the subsystem menu based on the subsystem the Club Member chooses.

* + 1. Functional Requirements

Logged in as club member and connected to the server

## 3.3 **Search Facility::REQ3**

3.3.1 Description and Priority

To allow club members to view available online facilities for booking or club’s services to request.

3.3.2 Stimulus/Response Sequences

Step 1: System prints out the Search Facility Subsystem menu.

Step 2: Club Member either does a normal search or an advanced search to search for available online facilities for booking or club’s services to request.

Alternate step 2: If the Club Member makes an invalid choice an error message is printed and redirected to menu. If he choose to quit he is redirected to the main menu

3.3.3 Functional Requirements

Logged in as club member and connected to the server

## 3.4 **Normal Search::REQ4**

3.4.1 Description and Priority

To allow club members to do a normal search on available online facilities for booking or club’s services to request.

3.4.2 Stimulus/Response Sequences

Step 1: System prints the search form.

Step 2: Club Member fills in data to be searched.

Step 3: System validates data.

Step 4: System processes data.

Alternate Step 4: If the data is invalid, an error message is displayed and the Club Member is sent back to the form to reenter the data correctly.

Step 5: System prints out result of the search.

Step 6: Club Member can choose to go to booking facility system or choose to quit in which case he will be redirected to search menu.

3.4.3 Functional Requirements

Logged in as club member and connected to the server

## 3.5 **Advanced Search::REQ5**

3.5.1 Description and Priority

To allow club members to do an advanced search on available online facilities for booking or club’s services to request.

3.5.2 Stimulus/Response Sequences

Step 1: System prints the search form.

Step 2: Club Member fills in data to be searched which can be done by various criteria such as by facilities, charging rates, availability and so on.

Step 3: System validates data.

Step 4: System processes data.

Alternate Step 4: If the data is invalid, an error message is displayed and the Club Member is sent back to the form to reenter the data correctly.

Step 5: System prints out result of the search.

Step 6: Club Member can choose to go to the booking facility system or choose to quit in which case he will be redirected to search menu.

3.5.3 Functional Requirements

Logged in as club member and connected to the server

## **3.6** **Sort Result::REQ6**

3.6.1 Description and Priority

To allow club members to sort the search results.

3.6.2 Stimulus/Response Sequences

Step 1: System prints out the search results.

Step 2: Club Member sorts the search results by various criteria such as by name, by booking time and so on.

3.6.3 Functional Requirements

Logged in as club member and connected to the server

## **3.7** **Booking Facility::REQ7**

3.7.1 Description and Priority

To allow club members to sort the search results.

3.7.2 Stimulus/Response Sequences

Step 1: System prints out the booking facility form.

Step 2: Club Member enters data for facilities to be booked which are the name of the facility and the booking time and date.

Step 3: System validates the data.

Step 4: System processes the data.

Alternate Step 4: If the data is invalid, an error message is displayed and the Club Member is sent back to the form to reenter the data correctly.

Step 5: System updates the database.

Step 6: System prints success message that the facility is successfully booked.

Step 7: The Club Member is redirected to the search menu.

3.7.3 Functional Requirements

Logged in as club member and connected to the server

## **3.8** **Cancel Booking::REQ8**

3.8.1 Description and Priority

To allow club members to cancel their bookings.

3.8.2 Stimulus/Response Sequences

Step 1: System prints all the booking made by the user

Step 2: Club Member chooses booking to be cancelled.

Step 2: System process the cancellation.

Step 3: System updates the database.

Step 4: System prints a successful message showing that the booking is successfully cancelled.

Step 5: The Club Member is redirected to the search menu.

3.8.3 Functional Requirements

Logged in as club member and connected to the server

## **3.9** **Edit Profile::REQ9**

3.9.1 Description and Priority

To allow club members to update their particulars.

3.9.2 Stimulus/Response Sequences

Step 1: System prints out particular form.

Step 2: Club Member fills in the particulars to be updated.

Step 3: System validates the data.

Step 4: System processes the data.

Alternate Step 4: If the data is invalid, an error message is displayed and the Club Member is sent back to the form to reenter the data correctly.

Step 5: System updates the database.

Step 6: System prints a successful message showing that the Club Member’s particulars are successfully updated.

Step 7: The Club Member is redirected to main menu

3.9.3 Functional Requirements

Logged in as club member and connected to the server

## **3.10** **Request Membership Ranking Upgrade::REQ10**

3.10.1 Description and Priority

To allow club members to request for upgrade of their rankings.

3.10.2 Stimulus/Response Sequences

Step 1: Club Member chooses to request for membership ranking upgrade.

Step 2: System processes the request.

Step 3: System updates the database.

Step 4: System prints message that the Club Member’s request is successful.

Step 5: The Club member can choose to cancel his request in which case he will be redirected to cancel request else he can choose to go back to main menu.

3.10.3 Functional Requirements

Logged in as club member and connected to the server.

## **3.11** **Cancel Request::REQ11**

3.11.1 Description and Priority

To allow club members to cancel their request for upgrade of their rankings.

3.11.2 Stimulus/Response Sequences

Step 1: Club Member chooses to cancel his request for membership ranking upgrade.

Step 2: System processes the request.

Step 3: System updates the database.

Step 4: System prints message that the Club Member’s cancellation is successful.

Step 5: Club Member is redirected to the main menu.

3.11.3 Functional Requirements

Logged in as club member and connected to the server

## **3.12** **View Notification::REQ12**

3.12.1 Description and Priority

To allow club members to view their notifications whenever their booking starts or ends.

3.12.2 Stimulus/Response Sequences

Step 1: Club Member chooses to view his notifications for his bookings.

Step 2: System processes the choice.

Step 3: System prints a table showing the actor’s notifications for his bookings.

Step 4: Club Member is redirected to the main menu (UC-ID 1) once he is done.

3.12.3 Functional Requirements

Logged in as club member and connected to the server.

## 3.13 View Bookings::REQ13

3.13.1 Description and Priority

To allow club members to view their bookings for facilities.

3.13.2 Stimulus/Response Sequences

Step 1: Club Member chooses to view his bookings.

Step 2: System processes the choice.

Step 3: System prints a screen showing the actor’s bookings.

Step 4: Club Member is redirected to the main menu (UC-ID 1) once he is done.

3.13.3 Functional Requirements

Logged in as club member and connected to the server

## 3.14 Manager Menu::REQ14

3.14.1 Description and Priority

To access the menu screen for club managers to choose the subsystems.

3.14.2 Stimulus/Response Sequences

Step 1: System prints out the menu screen for club managers.

Step 2: Club Manager chooses the subsystem from the list of subsystems in the menu.

Step 3: System processes the choice.

Step 4: System prints the subsystem menu based on the subsystem the Club Manager chooses.

3.14.3 Functional Requirements

Logged in as club manager and connected to the server

## 3.15 Create Facilities::REQ15

3.15.1 Description and Priority

To allow the club manager to create facilities for the database.

3.15.2 Stimulus/Response Sequences

Step 1: System prints out the form.

Step 2: Club Manager fills in the facility details.

Step 3: System validates the data.

Step 4: System processes the data.

Alternate Step 4: If the data is invalid, an error message is displayed and the Club Manager is sent back to the form to reenter the data correctly.

Step 5: System updates the database.

Step 6: System prints out success message that the facility is successfully added.

Step 7: System redirects the Club Manager to main menu.

3.15.3 Functional Requirements

Logged in as club manager and connected to the server

## 3.16 Delete Facilities::REQ16

3.16.1 Description and Priority

To allow the club manager to delete facilities for the database.

3.16.2 Stimulus/Response Sequences

Step 1: Club Manager chooses to delete a facility from the database.

Step 2: System processes the choice.

Step 3: System updates the database.

Step 4: System prints successful message that the facility is successfully deleted.

Step 5: System redirects the Club Manager to main menu

3.16.3 Functional Requirements

Logged in as club manager and connected to the server

## 3.17 Update Facilities::REQ17

3.17.1 Description and Priority

To redirect the club manager to respective facility services.

3.17.2 Stimulus/Response Sequences

Step 1: Club Manager chooses to update a facility from the database.

Step 2: System prints out menu for the actor to choose the subsystem.

Step 3: Club Manager makes his choice.

Step 4: System processes the choice.

Step 5: System redirects the user to respective sub system.

Alternate Step 5: If choice is invalid, an error message is displayed and the Club Manager is sent back to the update facilities menu.

3.17.3 Functional Requirements

Logged in as club manager and connected to the server

## 3.18 Set Charging Rates::REQ18

3.18.1 Description and Priority

To allow the club manager to set charging rates for a club’s facilities.

3.18.2 Stimulus/Response Sequences

Step 1: Club Manager fills in charging rates for a certain facility.

Step 2: System validates the data.

Step 3: System processes the data.

Alternate Step 3: If the data is invalid, an error message is displayed and the Club Manager is sent back to the form to reenter the data correctly.

Step 4: System updates the database.

Step 5: System prints success message and he is redirected to update facility menu

3.18.3 Functional Requirements

Logged in as club manager and connected to the server

## 3.19 Change Other Details::REQ19

3.19.1 Description and Priority

To allow the club manager to update facility details for the database.

3.19.2 Stimulus/Response Sequences

Step 1: System prints a menu with various field update option.

Step 2: Club Manager chooses the field he wants to update.

Step 3: System prints the form.

Alternate step 3: If the Club Manager chooses to quit he is redirected to update facility menu

Step 4: Club Manager fills in the data.

Step 5: System validates the data.

Step 6: System processes the data.

Alternate Step 6: If the data is invalid, an error message is displayed and the Club Manager is sent back to the form to reenter the data correctly.

Step 7: System updates the database.

Step 8: The Club Manager is redirected back to the update facility menu

3.19.3 Functional Requirements

Logged in as club manager and connected to the server

## 3.20 Set Membership Fees::REQ20

3.20.1 Description and Priority

To allow the club manager to set membership fees for the country club.

3.20.2 Stimulus/Response Sequences

Step 1: System prints all the rank types and fees

Step 2: Club Manager chooses the rank type.

Step 3: System prints out form.

Alternate Step 3: If invalid choice the system request reentry.

Step 4: Club Manager fills in data for membership fees for the country club.

Step 5: System validates the data.

Step 6: System processes the data.

Alternate Step 6: If the data is invalid, an error message is displayed and the Club Manager is sent back to the form to reenter the data correctly.

Step 7: System updates the database.

Step 8: System prints successful message that membership fees are successfully set.

Step 9: The Club Manager is redirected back to the main menu

3.20.3 Functional Requirements

Logged in as club manager and connected to the server

## 3.21 Set Membership Priorities::REQ21

3.21.1 Description and Priority

To allow the club manager to set membership priorities for the country club.

3.21.2 Stimulus/Response Sequences

Step 1: System prints all the rank types and priorities.

Step 2: Club Manager chooses the rank type.

Step 3: System prints out form.

Alternate Step 3: If invalid choice the system requests reentry.

Step 4: Club Manager fills in data for membership priorities for the country club.

Step 5: System validates the data.

Step 6: System processes the data.

Alternate Step 6: If the data is invalid, an error message is displayed and the Club Manager is sent back to the form to reenter the data correctly.

Step 7: System updates the database.

Step 8: System prints successful message that membership priorities are successfully set.

Step 9: The Club Manager is redirected back to the main menu

3.21.3 Functional Requirements

Logged in as club manager and connected to the server

## 3.22 Set Membership Access Rights::REQ22

3.22.1 Description and Priority

To allow the club manager to set membership access rights for the country club

3.22.2 Stimulus/Response Sequences

Step 1: System prints all the rank types and Access rights.

Step 2: Club Manager chooses the rank type.

Step 3: System prints out form.

Alternate Step 3: If invalid choice the system requests reentry.

Step 4: Club Manager fills in data for membership access rights for the country club.

Step 5: System validates the data.

Step 6: System processes the data.

Alternate Step 6: If the data is invalid, an error message is displayed and the Club Manager is sent back to the form to reenter the data correctly.

Step 7: System updates the database.

Step 8: System prints successful message that membership access rights are successfully set.

Step 9: The Club Manager is redirected back to the main menu.

3.22.3 Functional Requirements

Logged in as club manager and connected to the server

## 3.23 Set Membership Discount::REQ23

3.23.1 Description and Priority

To allow the club manager to set membership discounts for the country club.

3.23.2 Stimulus/Response Sequences

Step 1: System prints all the rank types and discounts.

Step 2: Club Manager chooses the rank type.

Step 3: System prints out form.

Alternate Step 3: If invalid choice the system requests reentry.

Step 4: Club Manager fills in data for membership discounts for the country club.

Step 5: System validates the data.

Step 6: System processes the data.

Alternate Step 6: If the data is invalid, an error message is displayed and the Club Manager is sent back to the form to reenter the data correctly.

Step 7: System updates the database.

Step 8: System prints successful message that membership discounts are successfully set.

Step 9: The Club Manager is redirected back to the main menu.

3.23.3 Functional Requirements

Logged in as club manager and connected to the server

## 3.24 Facility Usage Report::REQ24

3.24.1 Description and Priority

To allow the club manager to view the facility usage report for the country club.

3.24.2 Stimulus/Response Sequences

Step 1: Club Manager chooses to view the facility usage report.

Step 2: System processes the choice.

Step 3: System prints out the facility usage report.

Step 4: The Club Manager is redirected back to the main menu.

3.24.3 Functional Requirements

Logged in as club manager and connected to the server

## 3.25 Rental Behavior Report::REQ25

3.25.1 Description and Priority

To allow the club manager to view the rental behavior report for the country club.

3.25.2 Stimulus/Response Sequences

Step 1: Club Manager chooses to view the rental behavior report.

Step 2: System processes the choice.

Step 3: System prints out the rental behavior report.

Step 4: The Club Manager is redirected back to the main menu

3.25.3 Functional Requirements

Logged in as club manager and connected to the server

## 3.26 Discount Rates Report::REQ26

3.26.1 Description and Priority

To allow the club manager to view the discount rates report for the country club.

3.26.2 Stimulus/Response Sequences

Step 1: Club Manager chooses to view the discount rates report.

Step 2: System processes the choice.

Step 3: System prints out the discount rates report.

Step 4: The Club Manager is redirected back to the main menu

3.26.3 Functional Requirements

Logged in as club manager and connected to the server

## 3.27 Update Manager Details::REQ27

3.22.1 Description and Priority

To allow Club Manager to update his/her particulars.

3.22.2 Stimulus/Response Sequences

Step 1: System prints out particular form.

Step 2: Club Manager fills in the particulars to be updated.

Step 3: System validates the data.

Step 4: System processes the data.

Alternate Step 4: If the data is invalid, an error message is displayed and the Club Member is sent back to the form to reenter the data correctly.

Step 5: System updates the database.

Step 6: System prints a successful message showing that the Club Member’s particulars are successfully updated.

Step 7: The Club Manager is redirected to main menu

3.22.3 Functional Requirements

Logged in as club manager and connected to the server

# 4. External Interface Requirements

## 4.1 User Interfaces

* **Login Interface:** This is for registered members and managers to get into the system by inputting user ID and password
* **Manager Interface:** This is for the managers to access the authorized features given to the managers. For example: Facilities Management, Club Membership Rankings, Reporting System, etc.
* **Member Interface:** This is for the members to access the authorized features given to the members. For example: Search Portal, Member’s Profile, Booking Facilities, etc.
* **Facilities Management Interface:** This is for the managers to manage the facilities. For example to create a facility, delete existing facilities or update facilities.
* **Club Membership Rankings Interface:** This is for the managers to manage the registered members. For example to set membership fees, priorities, access rights and discount.
* **Reporting System:** This is for managers to view the report of the facilities usage, rental behavior or discount rate.
* **Search Portal:** This is for the club members to find which facilities are available
* **Member’s Profile Interface:** This is for the members to edit profile, upgrade ranking, view notifications and view bookings.

## 4.2 Hardware Interfaces

There will be a centralized server in a system running Linux server operating system. The client can be installed into multiple systems running Linux so that multiple users can access the server at the same time

## 4.3 Software Interfaces

The client system is connected with the database at server side to retrieve the data for members or managers.

## 4.4 Communications Interfaces

Setting up the server into server mode requires that there will be open ports for accepting connections from the clients. The connection between the client and the server uses Connection oriented communication, via TCP/IP—Transfer Control Protocol/Internet Protocol, implements reliable delivery of messages. Connection-oriented communication makes programming easier because the protocol includes mechanisms for detecting and handling errors and an acknowledgment mechanism between client and server.

# 5. Non-functional Requirements

## 5.1 Performance Requirements

A temporary SQLite database will be created to store the user credentials locally to costs (time) that is incurred through constant communication with the central database. This database will be dropped when the user logout. This will boost the performance as the encryption and decryption processes will be heavily depend on this data. The performance is key requirement as mobile devises are small and don’t have computational power of personal computers.

## 5.2 Safety Requirements

One of the key non-functional requirement is to keep the user data safe. Following are some of the identified Safety requirements: -

***Dealing with stolen device***

The user can migrate to a new device using the backup data and change password so that the data in the old phone can never be decrypted as the sever will never authenticate the user even if he knows the old password.

***Dealing with missing folder***

The user will be promoted to download the backup folder and will be able to access files as usual or the user can choose to create a new one.

***Migration***

The user may wish to move to a new device, and the app allows user to download the backup file from the cloud and continue using as usual. The user credentials are stored in a central database to ease migration.

***Backup***

The user can back up his data to cloud (AWS S3) to retrieve it later when the disaster strikes or for migration. The system will track the changes and only save the newly added file. The data stored in cloud is send over after encryption.

***Changing Password***

The user after logging into the app can change the password by providing the old password. The password will be saved to the central database. The password forms a part of the key used to secure files and tables in the database. Once the password is changed the database will be secured using new password when the user logout.

***Password recovery***

If the user forgets the password he will receive a password recovery code through his email or notification. Upon entering the code, he will be redirected to the changing password procedure.

***Protection against GPS spoofing***

The location will be checked regularly (every 10 sec) and will be compared with the previous location, if there is a significant difference in the coordinates (100m) the app will encrypt the files and logout.

## 5.3 Security Requirements

Since the main aspect of this application is to keep files secure, this requirement is to non-trivial. Files should be encrypted using secure cryptographic algorithm such as Advanced Encryption standard (AES256). The file should not have the same name as the original file after encryption, the original name will be stored in the database along with the new name. The password should be hashed using strong hash functions such as the SHA256. The central database will have an additional layer of encryption provided by the cloud service provider (Amazon Web Service).

## 5.4 Software Quality Attributes

**Availability:** Checking that the system always has something to function and always pop up error messages in case of component failures. In that case the error messages appear when something goes wrong to prevail availability problems.

**Usability:** Checking that the system is easy to handle and navigates in the most expected way with no delays. In that case the system program reacts accordingly and transverses quickly between its states.

**Functionality:** Checking that the system provides the right tools to perform task mentioned in the functional requirement section. Also testing these functionalities run smoothly and providing simple learnable interface.

# **Appendix A: Glossary**

**Actors**: These are the stakeholders of the system. It may be a person, a group of people or an external system that may either support or use the main system.

**Architecture**: It is the highest level structure of the software system. The whole system is developed based on the software architecture.

**Client**: These are the systems that interact with the user and transfer data to a central server.

**Club Member**: Members of the country club those who will be using the system to make bookings.

**Club Manager**: The manager of the club who uses the system to make changes to the background data of the system on which the booking depends on.

**C++**: It is a middle-level programming language developed by Bjarne Stroustrup starting in 1979 at Bell Labs.

**Database system**: This is the place where the data is stored. We are currently using a text file to store the data on server side but it can be upgraded to a full-fledged database system.

**Linux**: Is a Unix-like operating system developed in bell labs. It is free and open-source.

**Ranking**: Refers to the different membership rights in the club. Different rank members have different priorities, different access rights and different discount rates.

**Rational Unified process (RUP):** Quoted from <https://en.wikipedia.org/wiki/Rational_Unified_Process> “The Rational Unified Process (RUP) is an iterative software development process framework created by the Rational Software Corporation, a division of IBM since 2003.[1] RUP is not a single concrete prescriptive process, but rather an adaptable process framework, intended to be tailored by the development organizations and software project teams that will select the elements of the process that are appropriate for their needs. RUP is a specific implementation of the unified process.” RUP has following phases:-

* **Inception**: Collecting ideas.
* **Elaboration**: Documenting and designing system based on the collected ideas.
* **Construction**: Developing the design into a programme and testing the system.
* **Transition**: Deploying the system.

**Server**: They are the back end subsystem which contains the entity class and the data files. They manage the flow of data to the client. They also fulfill the requests from multiple clients at the same time.

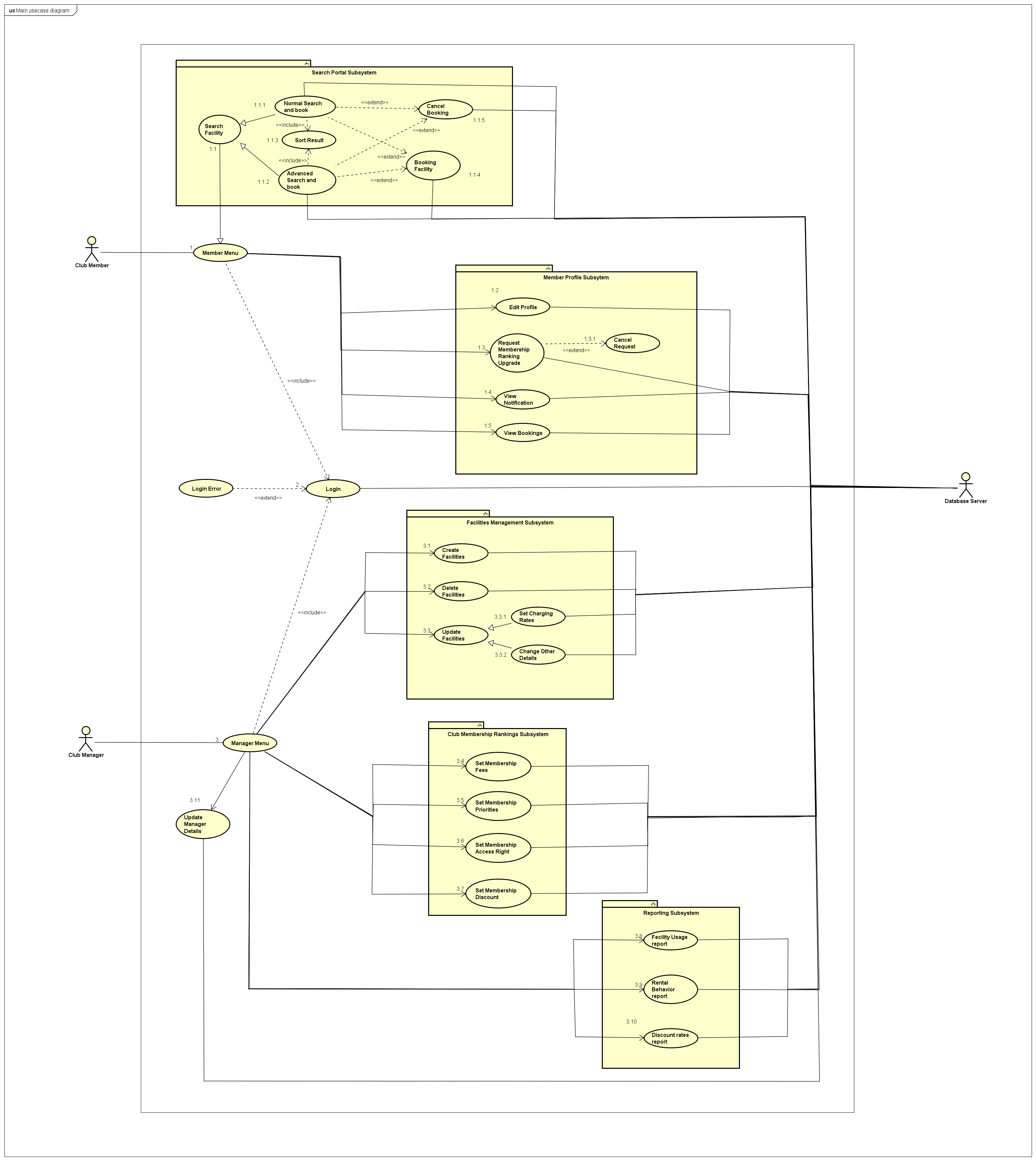
**System**: A system is a set of procedures according to which a problem is solved. A system has various subsystems which perform sub tasks to achieve a common goal.

**Subsystem**: A subsystem is a system containing various use cases which perform sub tasks to achieve common goal. Subsystem fulfill a major functionality of the entire system.

**TCP/IP—Transfer Control Protocol/Internet Protocol**:  Internet protocol suite is the computer networking model and a set of communications protocols used on the Internet and similar computer networks. TCP/IP provides end-to-end connectivity specifying how data should be packetized, addressed, transmitted, routed and received at the destination.

**Use-case**: It is the key functionality of the system.

# Appendix B: Analysis Models



Use Case diagram