
Software Requirements Specification

for

Clubhouse

Version <2.0.0>

Prepared by

Norman Low Wei Kit
Lee Swen Li
Anirwan Chowdhury

0317265
0317584
0801A66013

Group Name: Clubhouse

normanlowweikit@gmail.com
leeswenli@gmail.com
anirwanchowdhury@gmail.com

Instructor: Mr. Azween Abdullah

Course: BSc Computer Science

Lab Section: N/A

Teaching Assistant: N/A

Date: 15 APRIL 2015

Contents

CONTENTS	II
REVISIONS	III
1 INTRODUCTION.....	1
1.1 DOCUMENT PURPOSE	1
1.2 PRODUCT SCOPE.....	1
1.3 INTENDED AUDIENCE AND DOCUMENT OVERVIEW.....	1
1.4 DEFINITIONS, ACRONYMS AND ABBREVIATIONS.....	2
1.5 DOCUMENT CONVENTIONS.....	2
1.6 REFERENCES AND ACKNOWLEDGMENTS.....	2
2 OVERALL DESCRIPTION.....	3
2.1 PRODUCT PERSPECTIVE	3
2.2 PRODUCT FUNCTIONALITY	4
2.3 USERS AND CHARACTERISTICS	4
2.4 OPERATING ENVIRONMENT	4
2.5 DESIGN AND IMPLEMENTATION CONSTRAINTS.....	4
2.6 USER DOCUMENTATION	5
2.7 ASSUMPTIONS AND DEPENDENCIES	6
3 SPECIFIC REQUIREMENTS.....	7
3.1 EXTERNAL INTERFACE REQUIREMENTS.....	7
3.2 FUNCTIONAL REQUIREMENTS.....	12
3.3 BEHAVIOUR REQUIREMENTS	14
4 OTHER NON-FUNCTIONAL REQUIREMENTS	15
4.1 PERFORMANCE REQUIREMENTS.....	15
4.2 SAFETY AND SECURITY REQUIREMENTS.....	16
4.3 SOFTWARE QUALITY ATTRIBUTES.....	17
APPENDIX A – DATA DICTIONARY.....	19
APPENDIX B - GROUP LOG	20

Revisions

Version	Primary Author(s)	Description of Version	Date Completed
1.0.0	Anirwan Chowdhury Norman Low Wei Kit Lee Swen Li	First draft	08/04/2015
2.0.0	Anirwan Chowdhury Norman Low Wei Kit Lee Swen Li	Final Draft	15/04/2015

1 Introduction

The name given to this proposed project for a mobile application is Clubhouse. This mobile application's purpose is to provide a single platform where all clubs and societies can be listed out in one place, benefiting both the user (ease of finding out about clubs) and the clubs themselves (easy of exposure). This application does not only list out clubs however, but contains a myriad of other functions such as the listing of events (both upcoming and past), links to online registration of clubs, push notifications for members of clubs, and so on.

Following in this section will explain more about the purpose of this document as well as more detailed information about our product, Clubhouse.

1.1 Document Purpose

The purpose of this document is to give a detailed overview of the project, Clubhouse (scope, functionalities, constraints etc.), as well as to list out any requirements necessary for the development of this project.

As this proposed application is a new, self-contained product on its own and with no parent subsystem nor is it a member of an already existing product family, this document will not be talking about any other products and will only focus on Clubhouse.

1.2 Product Scope

The boundary of the mobile application covers clubs organizations that are established globally and in university, college, high school and elementary school level. The mobile app is a one stop platform that provides a location for all clubs to gain exposure. Its objective is to gather all community driven clubs for visibility to the public as well as to provide extra features to club members who use the application for clubs they have already joined such as events and news updates.

The main benefit this would have to general users or the club members (both potential and current) is that it makes searching for clubs significantly easier, since they can all be found in one place. Not only is it easy to find a club, our provided feature of allowing the club to link their registration methods to the application also makes it easier for a new member to register for the club.

For clubs however, the single largest benefit to be gained is the extra exposure. Having all the clubs listed in one place makes it easier for even the most "unknown" club to gain exposure and recognition. Aside from exposure, this application also provides many features that make it easy for a club to interact with its members such as by posting news and updates on events that will send a notification to any members currently signed up for that club.

1.3 Intended Audience and Document Overview

- Client
- Project Manager
- Lead Developer

1.4 Definitions, Acronyms and Abbreviations

API – Application Programming Interface
PRD – Product Requirements Document
SRM – System Requirements Manifest

1.5 Document Conventions

This document uses formatting requirements of Arial font size 11 or 12 throughout the document for text. Italics for comments. Document text should be single spaced and maintain the 1” margins.

1.6 References and Acknowledgments

[1] A. APIs, 'Android 3.0 APIs | Android Developers', Developer.android.com, 2015. [Online]. Available: <http://developer.android.com/about/versions/android-3.0.html>. [Accessed: 08- Apr- 2015].

[2] Developer.android.com, 'Building a Flexible UI | Android Developers', 2015. [Online]. Available: <http://developer.android.com/training/basics/fragments/fragment-ui.html>. [Accessed: 14- Apr- 2015].

[3] Developer.android.com, 'Layouts | Android Developers', 2015. [Online]. Available: <http://developer.android.com/guide/topics/ui/declaring-layout.html>. [Accessed: 14- Apr- 2015].

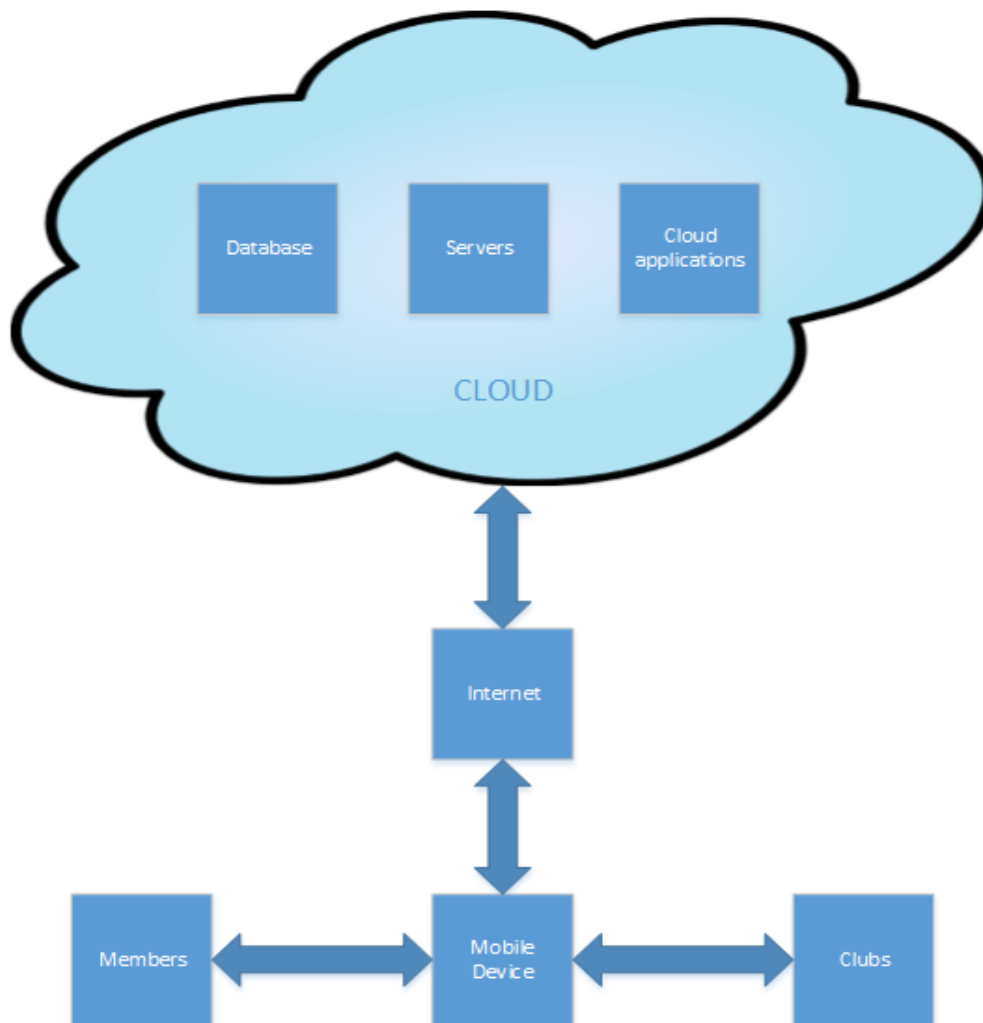
[4] SitePoint, 'Version Control Software in 2014: What are Your Options?', 2014. [Online]. Available: <http://www.sitepoint.com/version-control-software-2014-what-options/>. [Accessed: 08- Apr- 2015]

2 Overall Description

2.1 Product Perspective

As stated earlier in this document, this product is a new, self-contained product with no relation to any other existing products or product families currently out there. As of now there have been no existing products of similar scope.

Though this product is intended to be used globally for any clubs and societies, both public and private, the project proposed in this document is only going to be developed for Taylor's University as of now.



Above is a simple data-flow diagram to depict the data flow for this proposed project.

2.2 Product Functionality

Clubs	Users
<ul style="list-style-type: none">• Show club listing	<ul style="list-style-type: none">• Receive push notifications from club
<ul style="list-style-type: none">• Post news & updates	<ul style="list-style-type: none">• View all listed clubs according to region
<ul style="list-style-type: none">• Create events/meetings	<ul style="list-style-type: none">• Register for membership
<ul style="list-style-type: none">• View merchandise list	<ul style="list-style-type: none">• Synchronise calendar from app to 3rd party application
<ul style="list-style-type: none">• Customize registration link	<ul style="list-style-type: none">• One account for club application
<ul style="list-style-type: none">• View Membership list & details (hierarchy)	<ul style="list-style-type: none">• View news archive
<ul style="list-style-type: none">• View news archive	

2.3 Users and Characteristics

Members: Both potential members and current members. This project's target customers will almost solely consist of university students. These users will be interested in the search functions and also of the notifications for club updates.

Clubs: Will consist of whoever is put in charge of managing the application by the club itself, which we will refer to as the club administrator. These users will be interested in what features the application has to offer a club such as ability to post updates and news, view member lists, etc.

2.4 Operating Environment

The Clubhouse mobile application will require at least **Android API 3.0** (Honeycomb) platform for end users in order for it to be functional as intended on the smartphone. The Android 3.0 API is assigned an integer identifier of 11 that is stored within the system. The API determines whether the application is compatible with the system prior to installing the application. The targeted framework (android:targetSdkVersion) is Android API Level 19 in which it was developed and tested on. The application is designed to run on Android 3.0 and higher, in which it would not be able to install on earlier versions of the platform.

2.5 Design and Implementation Constraints

2.5.1 Timing Constraints

Ideally the project would require at least 6 months for the program to include all functions stated in the requirements.

2.5.2 Database Constraints

The project team requiring capital and/or the expertise to build and maintain an intended proprietary cloud based server and as such outsourcing to a 3rd party cloud provider is an option.

2.5.3 Security Constraints

Java security flaws which contributes to the susceptibility of this program.

2.5.4 Technology Constraints

The Clubhouse mobile application will require at least **Android API 3.0** (Honeycomb) platform for end users in order for it to be functional as intended on the smartphone. The Android 3.0 API is assigned an integer identifier of 11 that is stored within the system. The API determines whether the application is compatible with the system prior to installing the application. The targeted framework (android:targetSdkVersion) is Android API Level 19 in which it was developed and tested on. The application is designed to run on Android 3.0 and higher, in which it would not be able to install on earlier versions of the platform.

2.5.5 Skill of Programmers

Programmer proficiency in Java, JSON, Android, Swift and Windows Phone. The project is intended to be multi-platform however due to estimated lack of manpower and expertise in the initial stage of the project, the decision was made to switch to an Android-only application.

2.5.6 Resource/Tools Constraints

The Swift language is required to develop for an Apple device. The lack of specific resource – a Macintosh device – necessary to compile this language temporarily eliminates the option of developing on the iOS platform.

2.6 User Documentation

Upon entering the Clubhouse application for the first time, users will be prompted to go through a step-by-step tutorial on how to use the application first. This tutorial will demonstrate all the basic functions of Clubhouse.

Also, there will be a build in FAQ page as well as a “contact us” page for any further enquiries. The “contact us” option may also be used to report any errors or bugs in the application encountered.

2.7 Assumptions and Dependencies

- The content of the mobile application depends on participation of clubs.
- The use of this application assumes the user has a working internet/data connection.
- **Internet Connectivity** - The use of this application assumes the user has no internet connectivity at one particular point of time:

The club newsfeed listing functionality is highly dependent and requires the client's API for the mobile application to pull JSON data format provided and storing the data in a formatted display. As the data is needed to display to the user, it is a necessity for this function to operate. In the event that there is no internet connectivity, the Clubhouse application would retrieve the data pre-fetched stored locally in **SQLite Database**. The system is designed to retrieve the latest data whenever internet connectivity is available. During offline mode, users are unable to register as a new member and will not be able to receive notifications from club.

3 Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

Android Multi-pane Layouts

Android devices come in many different form factors and screen sizes hence the layout is designed to adjust to varying screen sizes and orientation.

Android Fragments

Fragment provides for alternative layouts based on screen size and orientation.





Figure: The same activity housing two fragments on different device – tablet and handset

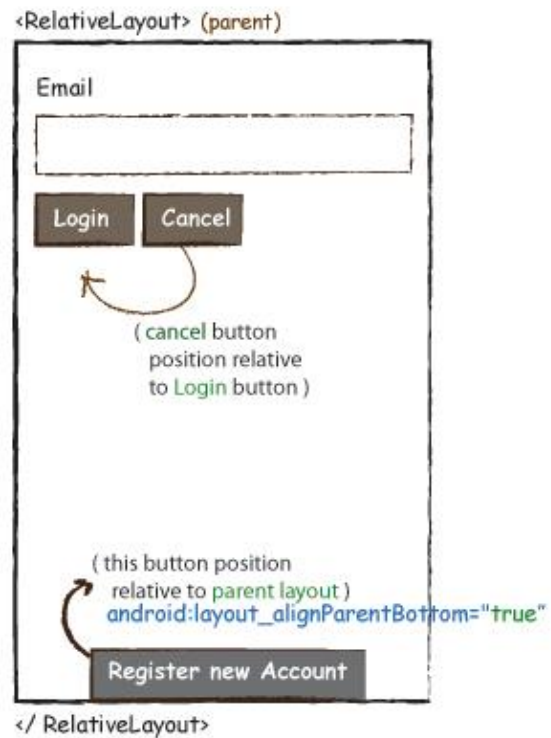
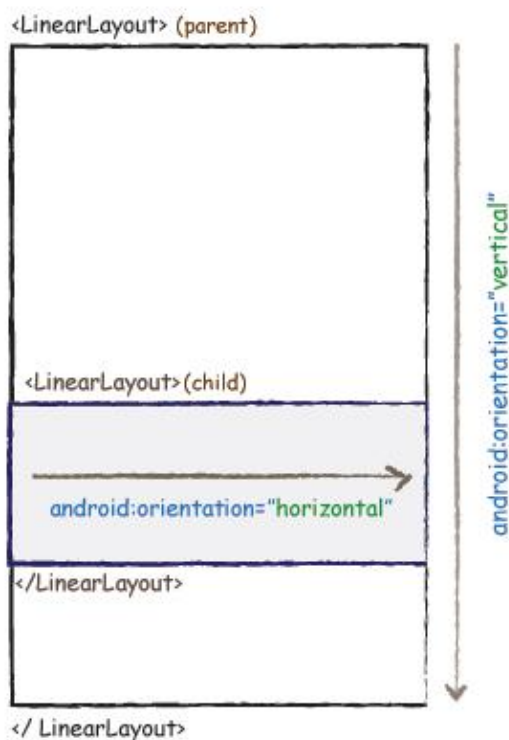
On a handset device, the screen is only capable of housing/fitting a fragment due to limits of the screen layout constraint therefore it can only fit one fragment at a time. The user will not be able to view Fragment B while in Fragment A until the user navigates to Fragment B. However tablets with a larger screen than the handset is able to fit two fragments side by side. The only exception that two can fragments is feasible to appear is when the handset is in a landscape orientation mode.

Android Layouts

The visual structure for a user interface (UI) is defined with a layout by declaring the UI elements in **XML**. The project will use 2 out of the 4 widely used Android layouts.

- FrameLayout
- LinearLayout
- RelativeLayout
- GridLayout

Linear Layout	Relative Layout
 <ul style="list-style-type: none"> • Horizontal or vertical row display • A scrollbar is created if the size constraints of the screen cannot accommodate the length of the application window. 	 <ul style="list-style-type: none"> • Every element arranges itself relatively to other sibling (children) elements of parent element. • Useful when adding views/button next to each other



Android Graphical User Interface (GUI)

This is the UI for the smartphone/handset.



3.1.2 Hardware Interfaces

Android System

The intended supported device types requires an Android 3.0 (API level 11) and later which will run on approximately 90.4% of the devices that are active on the Google Play Store. A lower API level allows for more devices to be targeted with the drawbacks of fewer features available. This software application will support tablets and smartphones.

Android 3.0 and earlier has four standard hardware buttons.



- **Back** button – navigates user to the previous step/stack
- **Menu** button – show menu options
- **Search** button
- **Home** button – returns user to the home screen

Android 3.0 and higher features a virtual navigation bar with **Home**, **Back** and **Recent** button (options). This application will use this feature.



- **Up**, **Search** and **Menu** button is relocated to the Action Bar.

The application will need to implement these buttons programmatically.

3.1.3 Software Interfaces

1. Minimum software requirement is Android 3.0 (Honeycomb) and above.
2. Clubhouse will be constantly connected (via Internet) to the chosen cloud service database to fetch data (club listings, member lists, etc.). In the event that there is no data or Internet connection, Clubhouse will use pre-fetched data stored locally in a SQLite database.
3. Clubhouse will communicate with 3rd party calendar applications (e.g. Google Calendar, iCal etc.) in order to synchronise dates of events and meetings.
4. Clubhouse will communicate with an external form system (e.g. Google Forms) in order to display registration forms.

DATABASE

To fetch data from the database, Clubhouse goes through an Ubuntu based server to pull data such as club listings and member lists from the chosen cloud service database.

The data is then received by the mobile phone in JSON, a lightweight data-interchange format, which will then be read and tokenized into legible data for the user by the application itself.

INPUT

This software will only receive input from one source, the user of the application. Once the user has given its input (e.g. browse filters), Clubhouse then filters through the data it has received via communication with the external cloud service database.

OUTPUT

The output to be seen would be the club listings and also individual Club profile pages as well as push notifications for club meetings and events.

3.1.4 Communications Interfaces

Communication function used in Clubhouse mobile application includes electronic forms and network server communications protocols. Electronic form is used when the users register their clubs or register as a member, they use the form to fill up their personal information. For the network server communications protocols, Secure Socket Layer (SSL) protocol is used to secure the data transfer between the mobile application and the servers. The user's information and transactions will be remained confidential by establishing an encrypted link between the server and the mobile application. Other communication standards such as FTP or HTTP protocols are used to transmit the interfaces in the mobile application such as displaying the users' profile and the clubs' profile.

For communication security and encryption issues, it will be taken care by the mobile vendors such as Google's Play Store and Apple's App Store.

3.2 Functional Requirements

3.2.1 User level facility

The Clubhouse mobile app must be able to facilitate the users' need by fulfilling the requirements such as posting news and updates of the clubs on the club's page. The admin of the clubs can create events or meetings and inform the members of the clubs. The merchandise list can be displayed for each club or society. A graphical user interface (GUI) must be able to display all the clubs and societies, their profiles and the events held.

3.2.2 System properties

The Clubhouse mobile app stores the users' data such as members' information, contact number, student ID, passwords, and other users' data in the cloud storage. Storage limitation will not be a problem to the development of the mobile app. The Clubhouse mobile app also has the function of synchronising the calendar to third party application.

3.2.3 Specific Algorithms

The Clubhouse mobile app must have the function of calculating the amount that the users need to pay for registration fees. The total number of members in each club or society depends on the package that the club or society sign up. The system will calculate the amount that the club or society pay to have more members in their club.

3.2.4 Specific Constraints

Every users need to sign up for an account before they can use the Clubhouse mobile app. The members who are interested in joining a club or society must fill in a registration form at a custom registration link. The admins must approve or decline the request of the users who are interested in joining their club or society. Only approved users can view the events and meetings, receive notifications about any updates, and search for the clubs and societies according to the region.

3.2.5 Legal Constraints

The Clubhouse mobile app must conform to the Governing Laws of the Malaysian Government. The Term of Service of Clubhouse is subjected to the terms and conditions governed in the Governing Law for its users. The system must ensure that all the personal information of the users are kept secure.

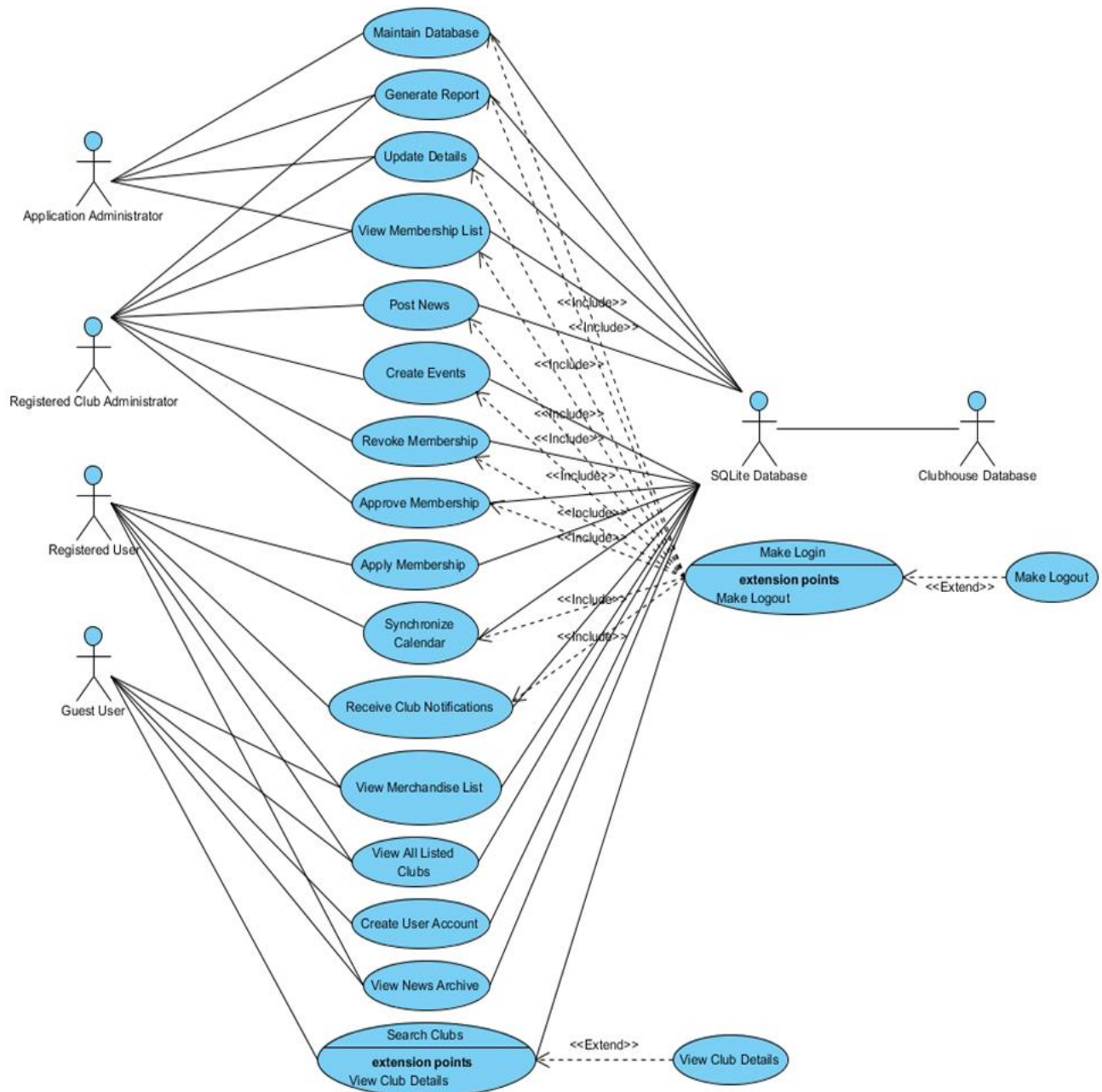
3.2.6 Functions

Functions	Descriptions
Clubs	
Show club listing	Clubhouse mobile app lists all the clubs and societies which have registered using the app. It categorises them into schools, universities and public to make it easier for the users to find the clubs they are interested in.
Post news & updates	Clubs and societies can post news and updates on their clubs' profiles to share with the club's members. Members can comment of the news and updates.
Create events/meetings	The admins for each club or society can create their own events and meetings with the details of time and location. The members will not miss the chance of participating in the events held by clubs and societies.
View merchandise list	The merchandise of clubs and societies are shown on the clubs' profile. The members can buy it online by contacting the admins of the clubs.
Custom registration link	Users who are interested in joining a club or society can register to sign up to be a member of the club. They need to fill up a registration form, which is created by the individual club or society.
View Membership list & details (hierarchy)	Only the clubs' admins and the developers can view the membership list and details of each member to keep the details of the users secure.
View news archive	News archive can be viewed by the members and admins of the club or society.
Users	
Receive push notifications from club	Users can get notifications from the club to know about the news and event taking place in a club.
View all listed clubs according to region	All the clubs are categorised according to regions. This can make it easier for the users to search for the clubs they are interested in.
Ease of membership registration	The process of membership registration is easy for all the users. The registration will be auto filled with the information from the users' account. The users can choose to link their Clubhouse account with social network such as Facebook and Twitter.
Synchronise calendar from app to 3 rd party application	Users can synchronise their calendar from the app to 3 rd party application to keep up with the latest events and meetings held by the clubs.

3.3 Behaviour Requirements

3.3.1 Use Case View

Depicted below is a simple use-case illustrating all users and functionalities.



4 Other Non-functional Requirements

4.1 Performance Requirements

4.1.1 Response time

The delays for the system must be as less as possible. For processes such as registering, opening windows, popping error messages and saving settings, the delay should be below 2 seconds. In displaying results from searching the databases, there should not be any delays and the results should be shown within 2 seconds.

4.1.2 Transaction rates/Throughput

The system must process 97 requests within 11.3s.

4.1.3 Workloads

The system RAM usage should be less than 50%, the other 50% will be left for background processes. Overloading of processes will slow down the system, which will lead to a poor performance.

4.1.4 Available storage space

The system must have sufficient memory capacity to carry out all the processes. Insufficient storage space will cause the mobile app to shut down abruptly.

4.1.5 Reliability

There must be less than 1 hour downtime per year and the system components are up-to-date. The user information must be stored at the correct location. The probability of failure of system must be kept at 0.1 to ensure that the mobile application is reliable.

4.1.6 Resources required

Keep the resources required to the minimum. Resources such as processor, disk space, memory and network should be used wisely to enhance the performance of the system.

4.2 Safety and Security Requirements

4.2.1 Safety Requirements:

1. The system should stop operating if the security attack is detected in the system to protect the data in the database.
2. The damage towards the system should be minimized by having firewalls and other protection measures to keep the mobile app safe.
3. Risk assessment and safety analysis are done every months to ensure that the mobile app is protected against risks and hazards.
4. An analysis of the potential hazard and risk is prepared for future reference on the causes and ways to mitigate the risks.
5. Hazard analysis and fault trees are used to analyze the safety requirements of the system. This can reduce the probability of the occurrence of hazards.

4.2.2 Security Requirements:

1. The user information is confidential and unauthorized attempts are resisted. Unauthorized users will not get the user information such as passwords and profile details. Only system's data administrator has the right to modify and access the system information.
2. All communications between the system's data server and clients' system must be encrypted to avoid being hacked by malicious virus. The encryption level must be 256-bit Advanced Encryption Standard (AES), which is also known as End-to-End encryption. This will provide extra security for the mobile app and make it more difficult to be hacked. The data will be encrypted using AES and decrypted at the destination. The keys will be more difficult to be derived by hackers by using this encryption level.
3. The mobile application Clubhouse should obtain Level 1 of the Federal Information Processing Standard (FIPS) Publication 140-2, which is the lowest level with limited requirements and loosely secured compared to other levels. Level 1 certification is the highest possible for a software-only product as levels above level 1 require hardware elements for physical tamper resistance.

4.3 Software Quality Attributes

4.3.1 Availability

The Clubhouse mobile application should achieve 98% uptime. This can be achieved by eliminating the failure in the system. If the mobile application is under maintenance or system information fails to be retrieved from databases, the issues must be resolved within a short time and the mobile application will be restarted for the users. The recovery time for the system must be as short as possible because a buggy mobile application is not an ideal application for the users.

4.3.2 Modifiability

Clubhouse mobile application will have a high modifiability so that the changes and updates are done in a shorter time. New features can be added anytime and unnecessary features can be deleted from the mobile application easily. The information in database and the settings must remain unchanged when installing the latest version of the mobile application. The effort and cost for modifying the mobile application determines the modifiability of the mobile application. The changes for the mobile application should cost less and less effort is needed to modify or update the mobile application.

4.3.3 Portability

The Clubhouse mobile application will support all Android platform. However, it will require at least Android API 3.0 (Honeycomb) platform. The language used for Clubhouse mobile application will be Java which will be programmed on the Android Studio software (along with any necessary plug-ins). In the future, Clubhouse will support iOS platform as well.

4.3.4 Readability

The code for the mobile application must be readable and not too complex. The requirements documentation of the project such as SRS, SDS, Software Process Documents, Requirement Specification and others should be clear and easily understood by the project team. The code and documents are both crucial to the maintainability of the project, which will affect the overall software quality.

4.3.5 Reusability

The Clubhouse mobile application must be systematic in order to have a high reusability. For Clubhouse, the project team uses the existing Android libraries in the code of the mobile application. The quality of a reused project will be improved and definitely better in terms of productivity and stability of system. The architecture of Clubhouse can be reused over and over again with minimal effort if the mobile application has a high reusability.

4.3.6 Robustness

The Clubhouse mobile application must be robust in way that make reading, changing, updating and managing the system easier. The mobile application must have the ability to handle the unexpected failures in the system. The estimated loss of data in case of connection lost during any operation should be less than 0.01%.

4.3.7 Scalability

The Clubhouse will be able to deal with large amounts of data as the number of user increases significantly over time. The project team has the sufficient resources such as CPU and memory size on the mobile application to handle the issue of increasing workloads and requirements. Clubhouse project team ensures that the increased workloads will not affect the process time for each requests and the system performance will not be degraded.

4.3.8 Testability

Clubhouse mobile application has high testability as the system can detect and solve the system failure easily. The architecture design of Clubhouse should be simple to make testing process easier. The testing process covers the code and requirements of the mobile application. The testing process must be done at a regular basis to make sure there is no presence of defects in the mobile application.

Appendix A – Data Dictionary

Android Constants - SQLiteDatabase

SQLite does not use Boolean storage class hence values are stored as integers 0 (false) and 1(true). **TEXT**, **REAL** and **INTEGER** are the available formats that the application can choose to store dates and times and is freely convertible using its built-in date and time functions.

Tables

Club			
Column Name	Data Type	Null	Description
CLUB_ID	Int	No	Primary key
COLUMN_CLUB_NAME	Text	No	Club name
COLUMN_ABBREVIATION	Text	No	Club initials
COLUMN_TYPE	Text	No	Type of club – technology, leadership
COLUMN_DATE_REGISTERED	Int	No	Date of club app registration
COLUMN_DATE_CLOSED	Int	No	Date of club app closure
COLUMN_EMAIL	Text	No	Club email for communication
COLUMN_WEBSITE	Text	Yes	The external club URL website
COLUMN_SHORT_DESC	Text	No	Item list view description
COLUMN_DESC	Text	Yes	Club content description

User			
Column Name	Data Type	Null	Description
USER_ID	Int	No	Primary key
COLUMN_FIRST_NAME	Text	No	User first name
COLUMN_LAST_NAME	Text	No	User last name
COLUMN_SURNAME	Text	No	User surname/family name
COLUMN_USER_TYPE	Text	No	Determines the user level – Club Administrator/End User
COLUMN_STATUS	Int	No	Active/Inactive
COLUMN_EMAIL	Text	No	User email
COLUMN_PASSWORD	Text	No	User password
COLUMN_PHONE	Int	No	User phone
COLUMN_COUNTRY	Text	No	User country (global user)
COLUMN_NATIONALITY	Text	No	User nationality

Membership			
Column Name	Data Type	Null	Description
CLUB_ID	Int	No	Foreign key
USER_ID	Int	No	Foreign key
COLUMN_DATE_JOINED	Int	No	Date of membership registration
COLUMN_DATE_CLOSED	Int	Yes	Date of membership closure

Appendix B - Group Log

- 1st Meeting: Monday, 6th April 2015, 12pm – 2pm / 8pm – 12am
Discussed what the main product was going to be, finally decided on clubhouse.
Briefly discussed product scope and functionalities.
Discussed what needed to be done. Made sure we were not leaving anything out.
- 2nd Meeting: Tuesday 7th April 2015, 1pm – 6pm
Delegation of work for the upcoming SRS draft.
Discussion of certain parts of the SRS.
- 3rd Meeting: Friday 10th April, 2pm – 4pm
Discussion of website
Discussion of remaining SRS
- 4th Meeting: Monday 12pm – 2pm
Discussion of all aspects of the SRS