Software Requirements Specification

for

<ForceQuit>

Version 1.1 approved

Prepared by <Zoe Lim>

<Trinity>

<11-04-21>

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
Zoe Lim	03-04-21	Draft	1.0
Zoe Lim	11-04-21	Completed Documentation	1.1

1.Introduction

1.1.Purpose

This document presents a detailed description of ForceQuit – an android application. It shall define all system boundaries, interfaces, and communications with or to external APIs and applications. Including diagrams and reasoning behind each design.

1.2.Intended Audience and Reading Suggestions

This document is intended for developers, project managers, users & testers. The “users” in this context refers to smokers who want to quit smoking and, who would potentially use the app to find a nearby quit center. A preferred sequence of going through the SRS, would be to follow the contents listed on the content page like a book. Perhaps any developers or project managers would be more interested in section 2 (Overall Description) onwards.

1.3.Product Scope

“ForceQuit” is an android application designed to assist not only in finding quit centers but also providing useful information to users. With Singapore aging population in mind, we planned to design a simple yet informative application. The information provided includes a quit center’s address, description, telephone number and also an all in one search function (postal code, quit center’s name or even a phone number look up) displayed on an interactive map.

1.4.References

The user interface design follows Ben Schneiderman's eight golden rules available here:

https://faculty.washington.edu/jtenenbg/courses/360/f04/sessions/schneidermanGoldenRules.html

2.Overall Description

2.1.Product Perspective

"ForceQuit" is meant to complement and support each other, between Google Maps and the Gov.sg Dataset. The government dataset provides many informative and useful data, yet it lacks proper visualization. Google map being a strong standalone API is capable of visualizing data onto a map. We can unify the strengths of each API and create a far more useful and superior application with some additional features to provide users with an enhanced user experience.

2.2.Product Functions

Major Functions:

- Search for a quit center.
- Viewing details of a selected quit center.
- Login with an Email account.
- Count down the days since the user stop smoking.

Minor Functions

- Able to edit profile details such as profile picture.
- Change password.

2.3.User Classes and Characteristics

Novice users – This class of users will most likely use the application to find out where quit centers are located at on the map.

Intermittent users – This class of users will be likely to use the application not only to find out where quit centers are located at on the map. They will also find out detailed information about quit centers such as their description and telephone number.

Advanced users – This class of users are generally tech-savvy and know the application very well. Will be using the search function and navigational features very comfortably. They also use the 'counter' to track their progress of smoking cessation.

Developers – A developer's job is to consistently monitor and maintain the application. This class of users are usually well-versed and proficient in Google APIs, Android, Flutter and Dart.

Our application will have to be well designed to satisfy the needs of these users.

2.4 Operating Environment

The application will be running on phones with at least Android 6.0 Marshmallow (API level 23) and above. The Minimum Software Development Kit (SDK) Version is 23 and Targeted Software Development Kit (SDK) Version is 28. The network connection requires a Wireless Network Interface Card (WNIC) with cellular network. Application will parse the dataset from gov.sg into Google Firebase for database management. After which the application will make a request to pull data from Firebase into the application for plotting of map markers on Google Maps. Also Google Places API will only be called only and only if user tap on the detailed information. This is to reduce the amount of API calls and also implemented as a pull observer pattern. User details and profile will be handled by Google Cloud Firestore. Lastly User login and authentication will be handled by Google Firebase Authentication.

2.5.Design and Implementation Constraints

Due to the heavy reliance on Google APIs. Especially Login access control. We have a few constraints for the authentication of valid email addresses. Basic authentication of checking if an email is a valid or non-valid email are provided. But sending an email verification to newly registered users for verifications requires additional costs. Which are not reflected within our budget. We will implement "true" authentication when our budgets are increased.

2.6.User Documentation

Application is designed in an open and play fashion. No tutorial will be needed, it's simple as it is.

2.7.Assumptions and Dependencies

Major assumption here is that the users will always be connected to the internet. If users are not connected to the internet, the application shall find and use the old cached data as its data set.

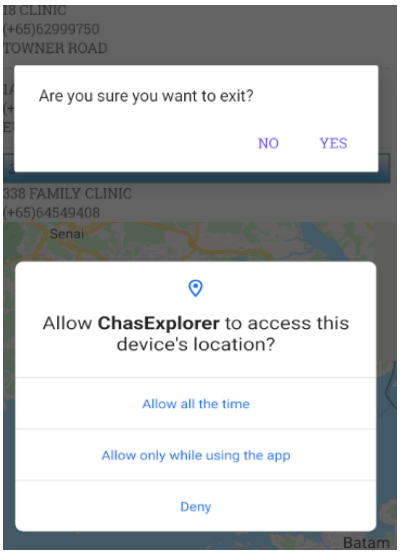
3.External Interface Requirements

3.1.User Interfaces

Shneiderman's eight golden rules of UI design were applied to the design of the application.

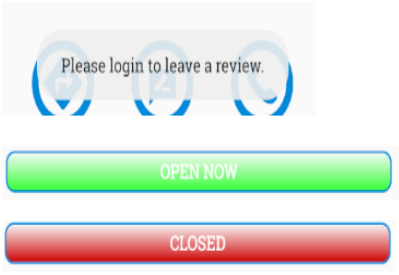
3.1.1.Loading Progress Bar

Loading bars are implemented into the applications where longer loading time are applicable. So as to prevent users from staring onto a blank screen thinking the application has "hanged".



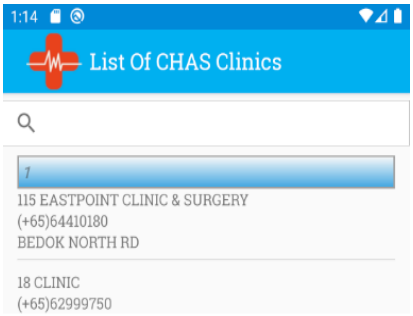
3.1.2.Confirmation Dialogs & Permission Dialogs

The Dialogs in ChasExplorer follows this simple design where title and message of the dialogs are changed accordingly to the situation.



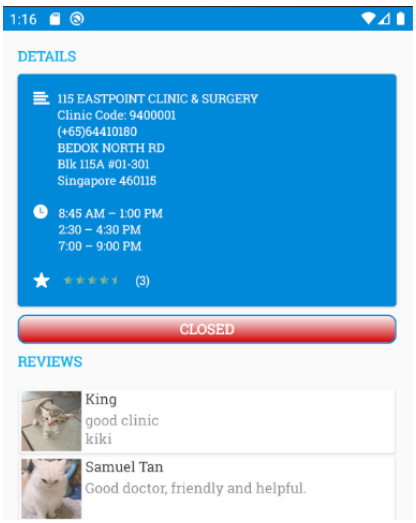
3.1.3.Informative Toasts & Dynamic Colours

The informative toasts ChasExplorer follows a simple design, they appear at the bottom of the screen with the appropriate message. Also Clinic's Opening hours are obtained in real time with the informative bar updating its colour based on the status.



3.1.4.Action Bar

The action bar follows the main application theme colour schema and stay consistent on all screens.



3.1.5.Card View Layout

ChasExplorer made use of Card View layout as a primary container of information. This promotes modular design and allows for more content, to be added to each individual pages in the near future by stacking more card views.

3.2. Golden Rules of UI designs (Shneiderman).

Explanations are listed below.

3.2.1.Strive for Consistency

Similar sequences of actions are all labelled clearly and properly. ChasExplorer uses a light colour schema consistently across all features. It looks neat and clean without any alteration to its design.

3.2.2.Cater to Universal Usability

ChasExplorer has simple features, doesn't throw too much information at user at once. This reduces the steep on the learning curve, enabling most users to operate the application with confidence and ease.

3.2.3.Offer Informative Feedback

As reflected on (3.1 UI) ChasExplorer informs users on various task and actions performed with informative feedback. Having these feedbacks allow users to be aware of the current status at all time.

3.2.4.Design Dialog to Yield Closure

Whenever a risky not so easily reversible action is performed. The user will be presented with additional dialogs to confirm their actions. This allows user to think twice, before committing any actions & also ease down on their anxiety level by reassuring them. Also being presented with a sequence of dialogs provide users additional confidence in operating the application.

3.2.5.Permit Easy Reversal of Actions

ChasExplorer supports the usage of the phone hardware button where users can go back to their previous page with ease and quick succession.

3.2.6.Support Internal Locus of Control

One of the key features of ChasExplorer is that users are not forced to login to use the application. Users are only required to login only and only if they want to leave a rating & review. Furthermore logged in users will be able to customize their display pictures and edit any ratings they have submitted. Giving them the feeling of control over every actions performed.

3.2.7.Reduce Short Term Memory

ChasExplorer is built with the intention of Singapore's Aging population in mind. Every features or functions in the applications are limited to 3 steps, and very limited clustering of data on the screen to prevent information overload. Reducing the need for users to remember or the need to be distracted by unnecessary information.

3.2.8.Error Prevention

The UI of ChasExplorer is designed to be robust and user friendly. Reiterating my previous points the application is simple, being simple it avoids big logic flaws which could lead to serious errors. Also all the networking details are masked behind the UI with the additional of multiple try catch's in code to handle those errors graciously.

3.3.Software Interfaces

The following APIs are used by the application:

- Singapore Government Dataset – To fetch Chas certified clinic details.
- Google Map API – Displaying Map & Markers.
- Google Firebase API – To store quit center data. (Real time Database)
- Google Firebase Authentication API – To handle the entire login / register module.

3.4.Communications Interfaces

For live dynamic way of getting the clinics opening hours. We used Google Places API. As such HTTPS protocol will be used to communicate with all APIs. However majority of the calls will be leading to Google Places API.

4.System Features

4.1.Login / Register

4.1.1.Description & Priority

Users who intend to use the application need to be logged in. To do so, they need to register with their email on this application. The feature is of medium priority as not everyone will want to register.

4.1.2.Stimulus / Response Sequences

1. To register, click on the 'register' button on the top right hand corner.
2. Enter email and password and click 'register'.
3. The user will be directed to the application's dashboard if the email and password provided are valid.
4. To log in, the user has to click on the 'login' button on the top right hand corner.
5. Enter email and password and click 'login'.
6. The user will be directed to the application's dashboard if the email and password provided are valid.

4.1.3.Functional Requirements

1. The application must allow the user to login with their registered account.
   1.1. If this is the user's first time using the application,

1.1.1. The application will display a ‘register’ option to the user.
1.1.2. The application will ask for the user’s ‘registering’ details:
1.1.2.1. User’s email
1.1.2.2. User’s password

1.2. If this is not the user’s first time using the application,
1.2.1 The application will display a ‘log-in’ option to the user.
1.2.2 The application will ask for the user’s ‘log-in’ details:
1.2.2.1. User’s email
1.2.2.2. User’s password

1.3. Upon logging in, the application must have a sign out option.

4.2.Search Quit Centre

4.2.1.Description & Priority

Users can search for quit centers by using the provided search bar to filter the amount of quit centers. This is a high priority feature as filtering and finding the quit centers you are finding out of hundreds of centers is important.

4.2.2.Stimulus / Response Sequences

<suhana>

4.2.3.Functional Requirements

2. The application must allow users to search the location of specific Quit Centres on the map by:
   2.1. The Postal code of the Quit Centre.
   2.2. The Telephone number of the Quit Centre.
   2.3. The Name of the Quit Centre.
   2.4. Users dragging the map manually with 1 finger to pan the map around.
3. The application must display more detailed information of a Quit Centre when the user taps into any of the map markers shown on the map.
   3.1. The application must display the Quit Centre’s full name.
   3.2. The application must display the Quit Centre’s full address.

3.3 The application must display the Quit Centre's telephone number.

4.3.View Counter Information

4.3.1.Description & Priority

Users can view their quitting progress by viewing their counter information that records the number of days since they quit smoking. This is the highest priority feature with the combination of search, these two are the main core function of the application.

4.3.2.Stimulus / Response Sequences

1. User tap on the 'dashboard' page on the bottom navigation bar to navigate to the dashboard.
2. User tap on the 'play' button to start the timer.
3. User views the counter information.
4. User can tap on the 'reset' button if he breaks his record of not smoking.

4.3.3.Functional Requirements

4. The application must display a counter in the dashboard page which shows the number of days the user has not smoked.

4.1 The application must display and allow a reset button to reset the number of days the user has not smoked, to zero.

4.4.Edit Profile Details

4.4.1.Description & Priority

Users can edit their profile detail if need be.
This feature is at a low priority as not all users will be using this feature mainly.

4.4.2.Stimulus / Response Sequences

1. User can tap on profile page in the bottom navigation bar to navigation to their profile details.
2. User can edit their name, profile picture and their quitting date.

4.4.3.Functional Requirements

5. The application must display the user's profile page.

5.1 The application must allow the edit of the user's profile details:
5.1.1.1 User's name
5.1.1.2 The date the user quits smoking
5.1.1.3 User's profile picture

5.2 The application must allow the user to save their profile details.

5.Other Nonfunctional Requirements

5.1.Performance Requirements

1. The application must be compatible with phones running Android 6.0 and above.
2. The application must be functional without GPS
   2.1. User must be able to perform the functions without GPS
      2.1.1.Search for clinic (List view)

Justifications:
Why android 6.0? – Released in Oct 2015, it's almost 5 years we figured it would be a good choice to set this as the base version of android we will support.

Functional without GPS why? – We wanted an application that wasn't solely designed based on the Google Map API, and would work even without location. It is bad design and honestly pointless if the application would stop working without GPS.

5.2.Software Quality Attributes

Usability

3. The application must be operable with either hands.
   3.1. User must be able perform the following functions single handily.
      3.1.1.Search for clinic (Map, List view)
      3.1.2.Pan the map

Justifications:
Why operable with either hands? Once again this application is catered to age 30+ they might not have the dexterity to execute complex gestures. So as a way of making sure the application was so simple and easy to use. We are benchmarking the usage of the application by being operable using either hands, dominant or non-dominant.

Appendix A: Data Dictionary

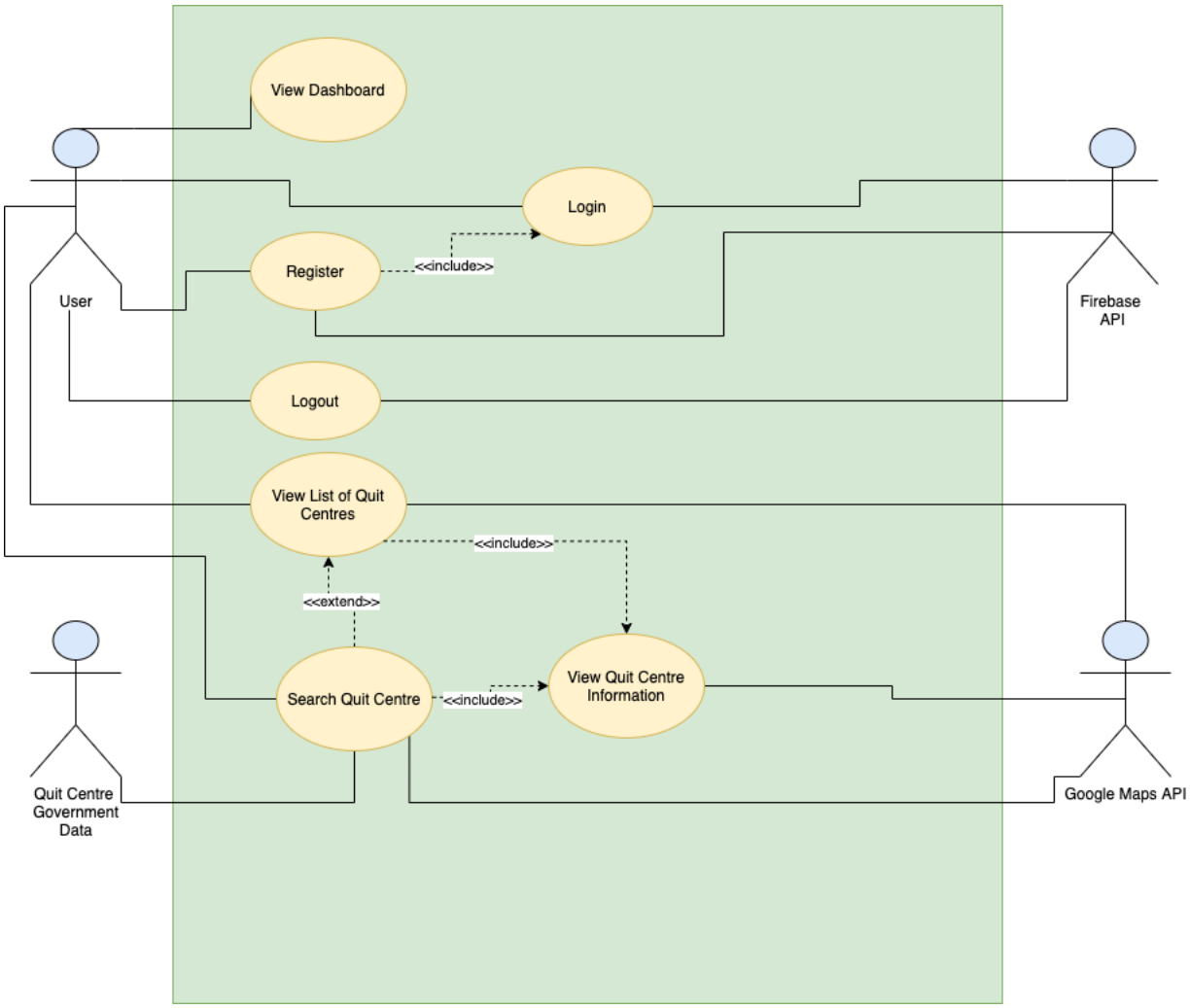
A set of terms matching with a description of the term in the context of the application.

General terms

Term	**Description**
Map marker	An indication of where the Quit Centre is, on the map.
User	A person who uses the application to look for Quit Centre.
Search	Search for Quit Centres based on user's given input.
Km	A unit of length in the metric system: Kilometer.
Ms	A thousandth of a second.
Counter	A timer displaying days:hours:minutes:seconds of the user being smoke-free, to capture their progress.

Appendix B: Analysis Models

B-1 (Use Case)



B-2 (Use Case Descriptions)

Use Case ID:	UC001		
Use Case Name:	Register		
Created By:	Zoe Lim	Last Updated By:	Zoe Lim
Date Created:	02-02-21	Date Last Updated:	01-04-21

Actor:	User
Description:	Register a new user account (local)
Preconditions:	The user has downloaded and started the application
Postconditions:	1) User account is created in the application 2) The application displays login page
Priority:	Medium
Frequency of Use:	Medium
Flow of Events:	Local Register: 1) The user selects “Register” 2) The user inputs new email and password 3) The application validates the user details 4) The user account is created 5) The application displays a successful message 6) Use case ends
Alternative Flows:	If user inputs invalid email address/password or an existed email address 1) The application displays an error message 2) The application returns to step 2 in the flow of events If user inputs empty fields 1) The application displays an error message 2) The application returns to step 2 in the flow of events
Exceptions:	1) If Local Server is down, display an informative message to try again later.
Includes:	-
Special Requirements:	-
Assumptions:	All Servers are up and running with no connection problems.
Notes and Issues:	-

Use Case ID:	UC002		
Use Case Name:	Login		
Created By:	Zoe Lim	Last Updated By:	Zoe Lim
Date Created:	03-02-21	Date Last Updated:	01-04-21

Actor:	User
Description:	Perform login procedure to allow user to view his dashboard
Preconditions:	User must be a registered user with the application
Postconditions:	1) User is successfully logged in and moved to the Dashboard 2) Dashboard displays counter and smoking statistics
Priority:	Medium
Frequency of Use:	Medium
Flow of Events:	1) The user selects "LOGIN" 2) The user inputs email address and password 3) The application validates the user details 4) The application moves to the dashboard 5) Use case ends
Alternative Flows:	If user inputs invalid email address/password or an existed email address 1) The application displays an error message 2) The application returns to step 2 in the flow of events If user inputs empty fields 1) The application displays an error message 2) The application returns to step 2 in the flow of events
Exceptions:	EX1: If user is not registered 1) The user starts from step 1) of "Register" use case
Includes:	Register
Special Requirements:	-
Assumptions:	All Servers are up and running with no connection problems.
Notes and Issues:	-

Use Case ID:	UC003		
Use Case Name:	Logout		
Created By:	Zoe Lim	Last Updated By:	Zoe Lim
Date Created:	03-02-21	Date Last Updated:	01-04-21

Actor:	User
Description:	The use case allows the user to log out of the account at any point of time
Preconditions:	User has successfully logged into the application
Postconditions:	User has successfully logged out of the application
Priority:	Low
Frequency of Use:	Low
Flow of Events:	1) The user selects “LOG OUT” 2) The user has logged out of the application 3) Use case ends
Alternative Flows:	-
Exceptions:	-
Includes:	-
Special Requirements:	-
Assumptions:	User is already logged in.
Notes and Issues:	-

Use Case ID:	UC004		
Use Case Name:	Search Quit Centre		
Created By:	Suhana Gupta	Last Updated By:	Suhana Gupta
Date Created:	03-02-21	Date Last Updated:	01-04-21

Actor:	User
Description:	The use case allows the user to search for a quit centre with postal code, telephone number, name
Preconditions:	The user has downloaded and started the application
Postconditions:	Display quit centre if match
Priority:	High
Frequency of Use:	High

Flow of Events:	1) The user selects the search bar 2) The user inputs the details 3) The user press search 4) The application compares the information 5) The application displays the matching quit centre 6) Use case ends
Alternative Flows:	No matching quit centre 1) The application displays not found 2) The user inputs the details again The user has successfully input search terms that corresponds to the list of quit centres matched 1) The user selects the search bar 2) The user inputs the details 3) The user press search 4) The application compares the information 5) The application displays the matching quit centres 6) Use case ends
Exceptions:	-
Includes:	-
Special Requirements:	-
Assumptions:	All Servers are up and running with no connection problems.
Notes and Issues:	-

Use Case ID:	UC005		
Use Case Name:	View List of Quit Centres		
Created By:	Suhana Gupta	Last Updated By:	Zoe Lim
Date Created:	03-02-21	Date Last Updated:	17-02-21

Actor:	User
Description:	The use case allows the user to view the names of all the list of quit centres before deciding what action to take next
Preconditions:	The user has downloaded and started the application
Postconditions:	The application displays view list of quit centres page
Priority:	High

Frequency of Use:	High
Flow of Events:	1) The user selects “List” 2) The application displays view list of quit centres page 3) Use case ends
Alternative Flows:	-
Exceptions:	-
Includes:	View Quit Centre Information
Special Requirements:	-
Assumptions:	All Servers are up and running with no connection problems.
Notes and Issues:	-

Use Case ID:	UC006		
Use Case Name:	View Quit Centre Information		
Created By:	Suhana Gupta	Last Updated By:	Zoe Lim
Date Created:	03-02-21	Date Last Updated:	17-02-21

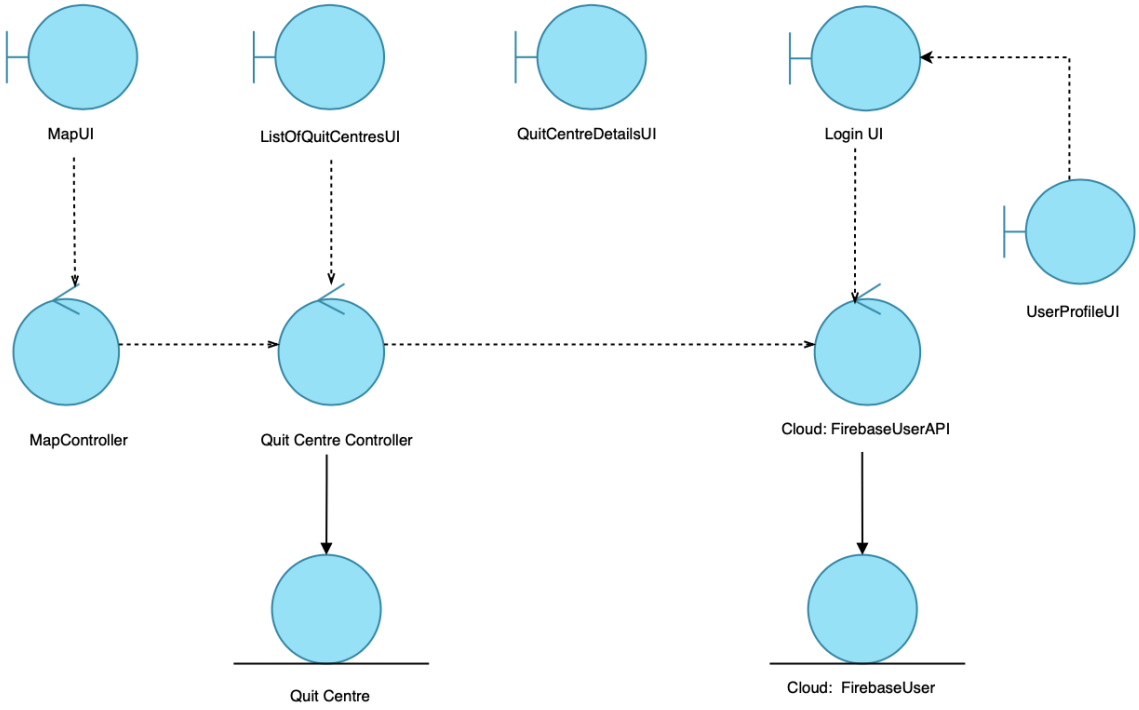
Actor:	User
Description:	The use case allows the user to view the information about a quit centre
Preconditions:	The user has successfully searched for a quit centre and click on it, or user click on one of the quit centre in the list
Postconditions:	The application displays a view quit centre information page
Priority:	Medium
Frequency of Use:	Medium
Flow of Events:	1) The user selects a specific quit centre 2) The application displays a specific quit centre page with the following information: • Name • Address • Telephone Number 3) Use case ends
Alternative Flows:	-

Exceptions:	-
Includes:	-
Special Requirements:	-
Assumptions:	All Servers are up and running with no connection problems.
Notes and Issues:	-

Use Case ID:	UC007		
Use Case Name:	View Dashboard		
Created By:	Parthan	Last Updated By:	Zoe Lim
Date Created:	03-02-21	Date Last Updated:	01-04-21

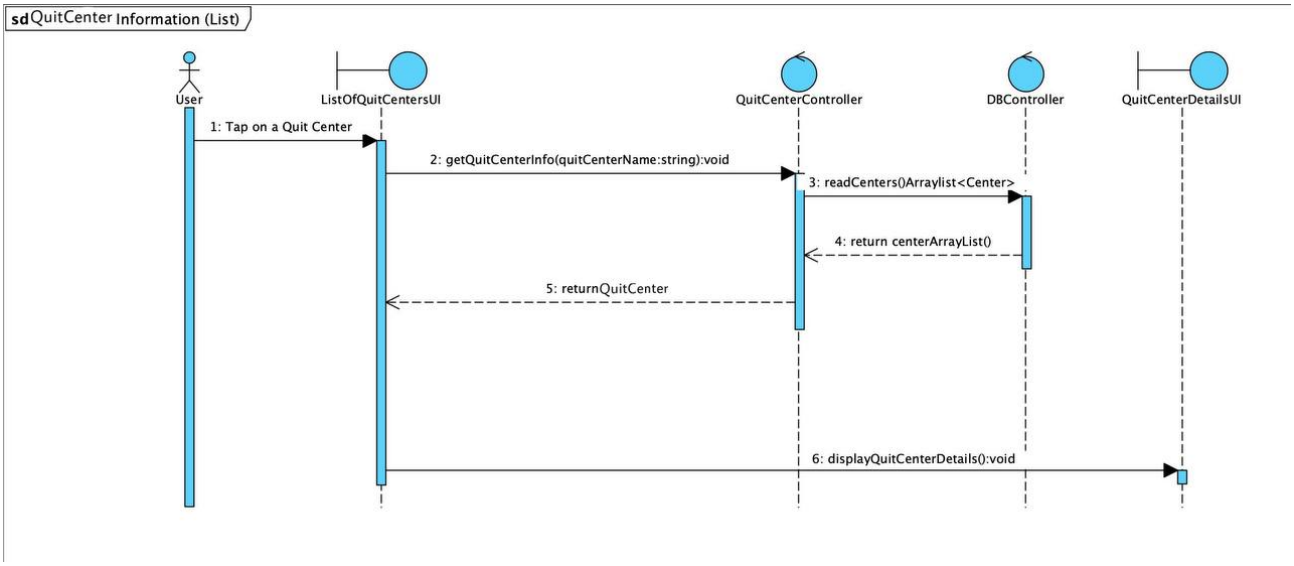
Actor:	User
Description:	The use case allows the user to view the information about his smoking cessation progress
Preconditions:	The user has successfully entered the dashboard page
Postconditions:	The application displays a view of the dashboard page
Priority:	Medium
Frequency of Use:	Medium
Flow of Events:	1) The user selects the dashboard page 2) The application displays the dashboard page with the following information: · Number of days,hours,minutes,seconds since the user quit smoking · Option to reset counter 3) Use case ends
Alternative Flows:	-
Exceptions:	-
Includes:	-
Special Requirements:	-
Assumptions:	All Servers are up and running with no connection problems.
Notes and Issues:	-

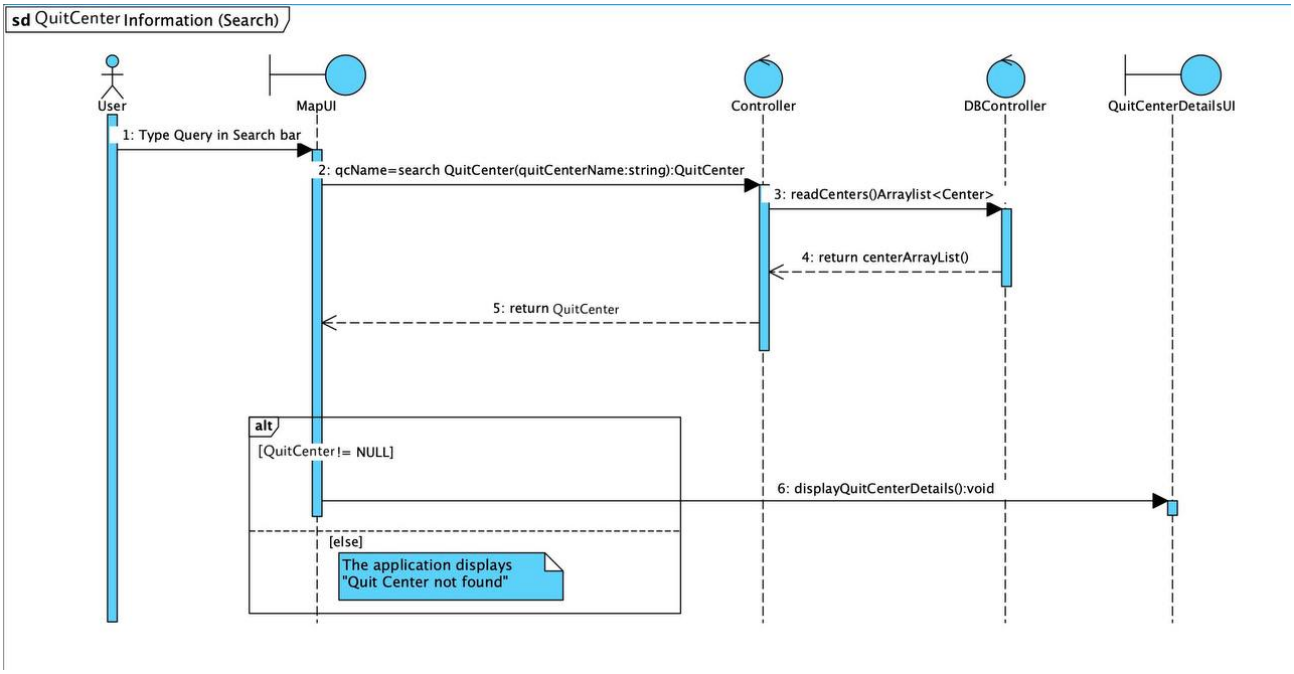
B-3 (Boundary Class Diagram)



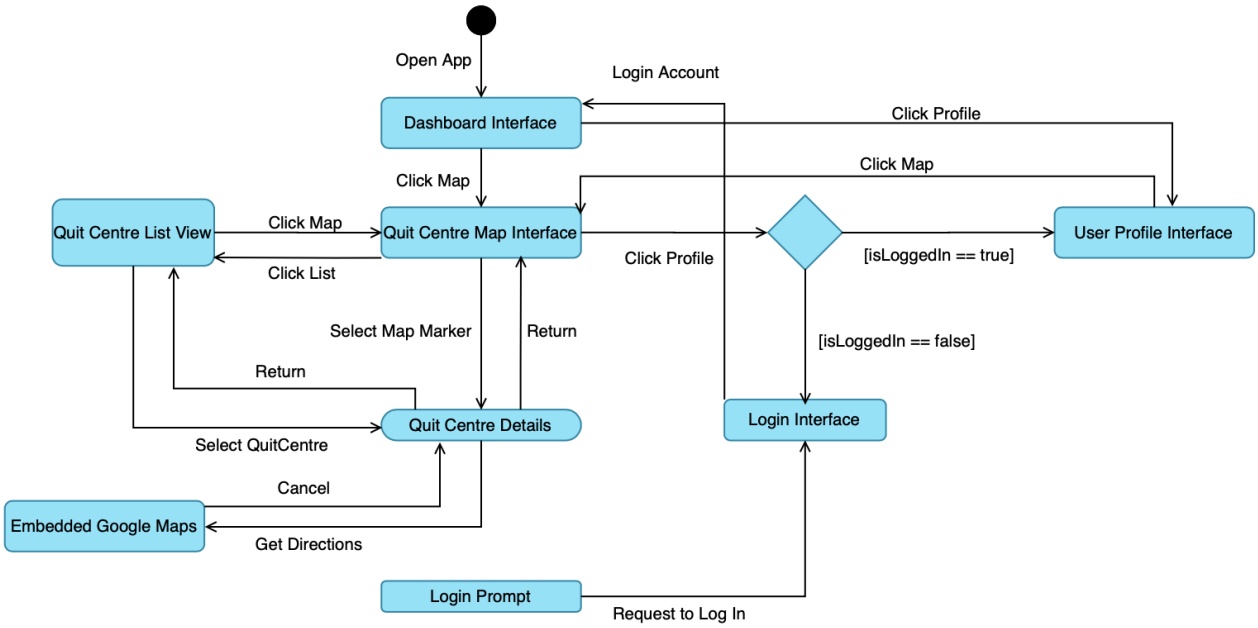
B-4 (Entity Class Diagram)

B-5 (Sequence Diagrams)

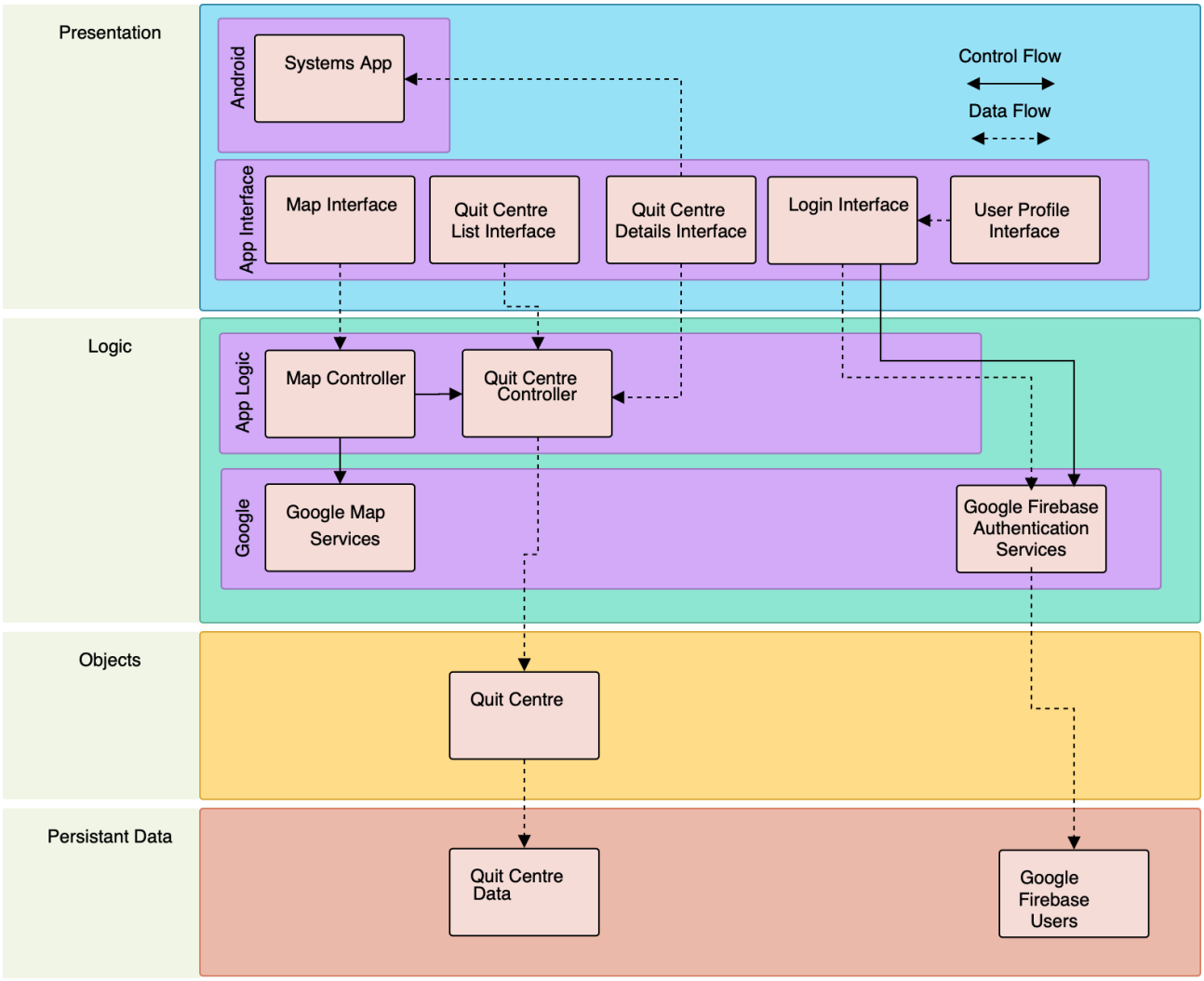




B-6 (Dialog Map)



B-7 (System Architecture)



B-8 (Black Box Testing)

Testing “Give Clinic Review” Control Class

Test Scenario Description	Test Case Description	Test Steps	Test Data	Preconditions	Expected Results	Post Conditions	Actual Results
Testing registration	Enter user Email and password	1. Press ‘Register’ button 2. Enter Email 3. Enter Password 4. Press register	Email: suhanagupta@hotmail.com Password: test1234	User should not have an exisiting account	Successful Registration	User should be Login page	Registered successfully
Testing Login system	Enter a valid Email and password	1. Enter a valid Email 2. Enter a Valid Password 3. Select Sign In	Email: suhanagupta@hotmail.com Password: test1234	User should be on Login Screen	Successful Login	User should be on Dashboard screen	Successfully logged in
Testing Login system	Enter a valid Email and invalid password	1. Enter a valid Email 2. Enter a invalid Password 3. Select Sign In	Email: suhanagupta@hotmail.com Password: test123	User should be on Login Screen	Wrong email or password message	User should be on Login Screen	Error message
Testing Login system	Enter a invalid Email and valid password	1. Enter a invalid Email 2. Enter a valid Password 3. Select Sign In	Email: suhana@hotmail.com Password: test1234	User should be on Login Screen	Wrong email or password message	User should be on Login Screen	Error message
Testing Login system	Enter a invalid Email and invalid password	1. Enter a invalid Email 2. Enter a invalid Password 3. Select Login	Email: suhana@hotmail.com Password: test123	User should be on Login Screen	Wrong email or password message	User should be on Login Screen	Error message
Test Password reset	Enter email address to be able to reset password	1. Select reset password option 2. Enter email address 3. Press send request	Email: suhanagupta@hotmail.com	User should have a registered account with entered email address	Password will be reset	User will be on Login screen	Password rest successfully
Testing access to Dashboard page	Accessing the Dashboard page	1. Press ‘Dashboard’ button	NIL	User should have already logged in.	The screen should switch to the dashboard page	User should be on Dashboard page	Accessed successfully
Testing dashboard functionality	Utilise the timer on the dashboard page	1. Press start button. 2. Press reset button	NIL	User should have already logged in	Upon pressing the start button the timer should start counting. Upon pressing rest it should go back to 0.	Timer will show 00:00	Timer functioned properly
Testing access to profile page	Accessing the profile page	1. Press ‘Profile’ button	NIL	User should have already logged in.	The screen should switch to profile page	User should be on Profile page	Accessed successfully
Testing edit profile functionality	Enter new username and password	1. Tap on existing email and password 2. Type new email and	Email: parthan97@gmail.com. Password: test12345	User should be on Profile page	The new email and password will be displayed	User on profile page showing new email and password	Edited successfully
Testing access to Map page	Accessing the Map page	1. Press Map button	NIL	User should have already logged in.	The screen should switch to Map page	User will be on Map page	Accessed successfully
Test Map functionality	See if quit centre details being displayed	1. Press on the pin displayed on map.	NIL	User should be on the Map page	Name of quit centre and contact details should pop up	User can observe contact info of a quit centre	Information is displayed
Test Map search	Filtering the quit centre displayed on map	1. Press search bar 2. Type info of quit centre	Guardian	User should be on Map page	List of quit centre that contain the word guardian	User will see a page of search results	Search results shown
Testing logout system	Loggin out of the app	1. Press the logout button	NIL	User should have already logged in.	Screen should switch to the login screen	User should be on the Login screen	Logged out successfully

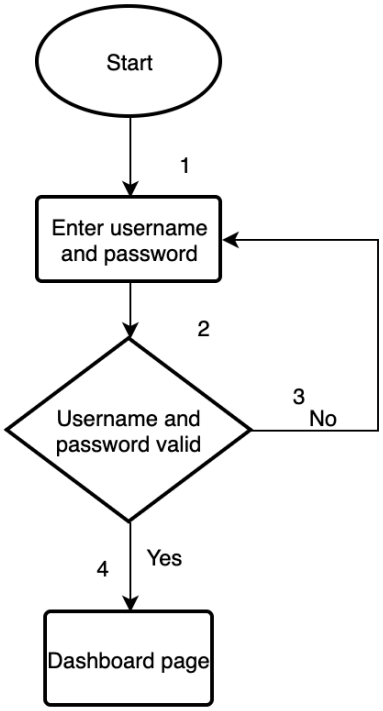
B-9 (White Box Testing)

Control Flow Test: Login

Test Cases:
a) User enters a valid username and password
b) user enters an invalid username or password

Execution Paths:
a) 1,2,4
b) 1,2,3,1,2,4

Cyclomatic Complexity:
[Conditions + 1] = [5 + 1] = **6**
[Edges - Nodes + 2] = [18 - 14 + 2] = **6**



Control Flow Test: Search Clinic on Search bar

Basis Paths:
a) 1,2,3,5
b) 1,2,3,4,6,2,3,5

Test Cases:
a) When there is a successful match for the list of quit centres
b) When there is an unsuccessful match for the list of quit centres