Software Requirements Specification

for

<ChasExplorer>

Version 1.1 approved

Prepared by <Samuel>

<theboys>

<11-11-19>

Table of Contents

Table of Contents ii

Revision History iii

1. Introduction 1

1.1 Purpose 1

1.2 Intended Audience and Reading Suggestions 1

1.3 Product Scope 1

1.4 References 1

2. Overall Description 2-3

2.1 Product Perspective 2

2.2 Product Functions 2

2.3 User Classes and Characteristics 2

2.4 Operating Environment 3

2.5 Design and Implementation Constraints 3

2.6 User Documentation 3

2.7 Assumptions and Dependencies 3

3. External Interface Requirements 4-7

3.1 User Interfaces 4

3.1.1 Loading Progress Bar 4

3.1.2 Confirmation Dialogs & Permission Dialogs 4

3.1.3 Informative Toasts & Dynamic Colours 4

3.1.4 Action Bar 5

3.1.5 Card View Layout 5

3.2 Golden Rules Of UI Design 5

3.2.1 Strike for Consistency 5

3.2.2 Cater to Universal Usability 5

3.2.3 Offer Informative Feedback 6

3.2.4 Design Dialog to Yield Closure 6

3.2.5 Permit Easy Reversal of Actions 6

3.2.6 Support Internal Locus of Control 6

3.2.7 Reduce Short Term Memory 6

3.2.8 Error Prevention 6

3.3 Software Interfaces 7

3.4 Communications Interfaces 7

4. System Features 7-11

4.1 Login / Register 7

4.2 Search Clinic 8

4.3 View Clinic Information 9

4.4 Rate & Review Clinic 9

4.5 Call Clinic's Tel. No 10

5. Other Nonfunctional Requirements 11-12

5.1 Performance Requirements 11

5.2 Software Quality Attributes 12

Appendix A: Data Dictionary 12

Appendix B: Analysis Models 13-29

B-1: Use Case 13

B-2: Use Case Descriptions 13-19

B-3: Boundary Class Diagram 19

B-4: Entity Class Diagram 20

B-5: Sequence Diagrams 21-24

B-6: Dialog Map 25

B-7: System Architecture 26

B-8: Black Box Testing 27

B-9: White Box Testing 28-29

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
| Samuel Tan | 11-11-19 | Draft | 1.0 |
| Samuel Tan | 11-11-19 | Completed Documentation | 1.1 |

# Introduction

## Purpose

This document presents a detailed description of ChasExplorer – 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.

## Intended Audience and Reading Suggestions

This document is intended for developers, project managers, users & testers. The “users” in this context refers to anyone, who would potentially use the app to find a nearby Chas certified clinic. 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.

## Product Scope

“ChasExplorer” is an android application designed to assist not only in finding Chas certified clinics 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 clinic’s opening hours, reviews by other users, direct call with a button and also an all in one search function (postal code, clinic’s name or even a phone number look up) displayed on an interactive map.

## 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>

# Overall Description

## Product Perspective

“ChasExplorer” 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, however it lacks detailed information. With the combination of these 2 APIs. We are able to 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.

## Product Functions

Major Functions:

* Search for Chas certified clinic.
* Viewing details of a selected clinic.
* Submit rating & review for the selected clinic.
* Call a clinic Tel. no through the application.
* Login with a Google account / Email account.

Minor Functions

* Able to change user display picture.
* Get directions via Google Map to the selected clinic.
* View all nearby clinics radius of 2.4km
* Change password.

## User Classes and Characteristics

Novice users – This class of users will most likely use the application to find out where clinics are located at on the map, and also probably call the clinic to check for more information / bookings.

Intermittent users – This class of users will be likely to use the application not only to find out where clinics are located at on the map. They will also find out detailed information about clinics such as their opening hours, and reviews.

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. These users are also the main contributor of reviews and ratings.

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, XML and Java.

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.

## 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. Firebase Authentication API allow developers to compile multiple Social Medias login and a local login into a single page. 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.

## User Documentation

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

## 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.

# External Interface Requirements

## User Interfaces

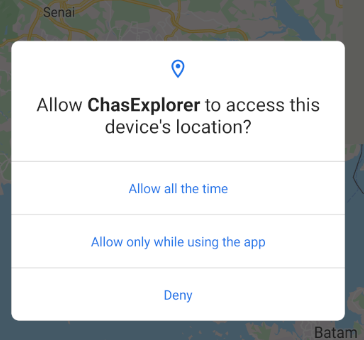
Shneiderman’s eight golden rules of UI design were applied to the design of the application.

### 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”.

### 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.



### 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.

### Action Bar

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

### 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.

## Golden Rules of UI designs (Shneiderman).

Explanations are listed below.

### 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.

### 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.

### 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.

### 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.

### 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.

### 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.

### 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.

### 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.

## 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 clinic data. (Real time Database)
* Google Firebase Authentication API – To handle the entire login / register module.
* Google Places API – To get the opening hours of each clinics.
* Google Cloud FireStore – To store user details + photos of users.

## 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.

# System Features

## Login / Register

### Description & Priority

Users who intend to submit ratings & reviews need to be logged in. To do so they can pick to login with either that Google email or any other existing email. This feature is of medium priority as not everyone will want to rate and review a clinic.

### Stimulus / Response Sequences

* 1. User tap on the profile icon.
* 2. Application will check if user is already logged in.
* 3. If user is logged in it shall redirect user to login page.
  + Else user will be asked to login.
* 4. User can pick either Google email or email for login.
  + If user pick Google email, it shall check if user’s phone is attached to any Google account.
    - If found, user can choose the list of detected Google accounts found.
    - Else user can manually key in their Google email to login.
  + Else user would have chosen email login, user shall key in their email.
    - If email is already registered before it will prompt user for password
    - Else it will mean that this is a new user and register this user by asking user to fill in more information.

### Functional Requirements

1. The application must allow the user to login with their Google account.
   1. Upon logging in, the application must have a sign out option.

## Search Clinic

### Description & Priority

Users can search for clinics by using the provided search bar to filter the amount of clinics.

This is a high priority feature as filtering and finding the clinic you are finding out of thousands of clinic is important.

### Stimulus / Response Sequences

* 1. User tap on the search bar.
* 2. User type in their query.
  + If user search on the list view page the list view will refresh dynamically real time to show the results.
  + Else this would mean user would search at the map page. In order to execute the search at the map page user would have to click submit.
    - After submission it will first search for clinic names, postal code and tel.no.
      * Results are displayed as map markers.

### Functional Requirements

1. The application must allow users to search the location of specific clinics on the map by
   1. The Name of the clinic.
   2. The Postal code of the clinic.
   3. The Telephone number of the clinic.
2. The application must enable the user to view a list of clinics without the map.

## View Clinic Information

### Description & Priority

Users can view clinic’s detailed information by accessing it through the list view of clinics or by the map markers informative markers. This is the highest priority feature with the combination of search, these two are the main core function of the application.

### Stimulus / Response Sequences

* 1a. User tap on the clinic list view icon. It will bring user to a page full of clinics
* 2a. User tap on any clinic listed there.
* 1b. User tap on any map markers populated on the map. A pop out box will appear
* 2b. User tap on the marker’s informative box.
* 3. User will be redirected to the specific page with the selected clinic full details.

### Functional Requirements

1. The application must display more detailed information of a clinic when the user taps into any of the map markers shown on the map.
   1. The application must display the clinic’s full name.
   2. The application must display the clinic’s full address.
   3. The application must display the clinic’s rating scores, the number of total ratings and reviews.
   4. The application must display the clinic’s telephone number.
   5. The application must display the clinic’s operating hours.
      1. When the user taps on the operating hours
         1. The application must display if the clinic is currently open according to system time.
         2. The application must display more details of the daily operating hours of the clinic.

## Rate & Review Clinic

### Description & Priority

Users can rate & review a clinic based on their experiences there to let others know what to look out for. Or if they have any recommendation or comments they can do so.

This feature is at a low priority as only advanced users will be using this features mainly.

### Stimulus / Response Sequences

* 1. User can access the submit review page by going into the clinic details page.
* 2. User click on the review icon at the bottom.
  + Application will check if user is logged in, if not user will be prompted to login.
* 3. User will then give a rating & review.
* 4. User clicks submit.
  + Application will check if review and ratings are valid. i.e. not blank
* 5. A toast will be displayed to inform user of the status of submission.

### Functional Requirements

4.3.1 When user taps on ratings, the application must ask user to login prior to giving a rating or leaving a review.

4.3.1.1 If user is already logged in,

4.3.1.1.1 User must be able to leave a rating out of five stars based on their   
experience in the clinic.

4.3.1.1.2 User must leave a compulsory review alongside the rating.

4.3.1.1.3 The application must accumulate the total ratings and calculate the weighted average from total users.

4.3.1.2 If user is not logged in, the application must provide an informative message to the user asking them to login.

## Call Clinic’s Tel. No

### Description & Priority

Because the gov.sg API was so informative we have the clinic’s tel. no as well. Users can actually call the clinic via the application. This function is as of medium priority as not every user will call the clinics unless they have some important queries to ask.

### Stimulus / Response Sequences

* 1. User can access the call clinic page by going into the clinic details page.
* 2. User click on the phone icon at the bottom.
  + Application will ask user for permission to access their phone call function.
    - If granted call will commence.
    - Else it will inform user permission was not granted and return back to the clinic details page.

### Functional Requirements

4.4.2 When the user taps on the telephone number icon, the application must seek for user’s permission before accessing the user’s phone call function.

4.4.2.1 If permission is given, the application must be able to call the clinic.

4.4.2.2 If permission is denied, the application must show an informative message telling the user to grant permission it.

# Other Nonfunctional Requirements

## Performance Requirements

1. The application must be compatible with phones running Android 6.0 and above.
2. The application must fetch clinic data details within 500ms when a user taps into any clinic.
   1. If the data fetching fails, the application must retry up to 3 times before displaying an informative error message.
3. The application must be functional without GPS
   1. User must be able to perform the functions without GPS
      1. Search for clinic (List view)
      2. Call clinic

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.

Why 500ms and why retry 3 times? – Mainly because the main bulk of the users using this application would most likely be aged 30 and above. 350ms is the average response time of an elderly based off a research done in 2016. We figured setting 500ms would be an ideal number. Retrying 3 times is just to ensure something is really wrong with the server and not just any random bug.

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.

## Software Quality Attributes

**Usability**

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

Justifications:

Why operable with ether 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

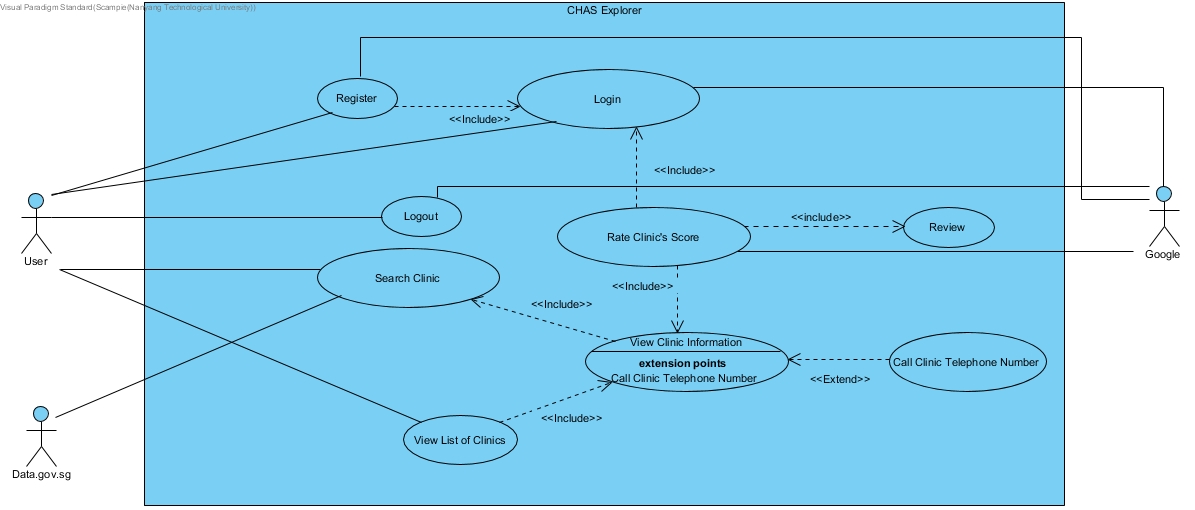
A set of terms matching with a description of the term in the context of the application.

**General terms**

|  |  |
| --- | --- |
| **Term** | **Description** |
| Map marker | An indication of where the clinic is, on the map. |
| Rating | A measure of users’ satisfaction on a particular area ranging from 1 to 5 stars where 1 depicts very unsatisfactory and 5 depicts very satisfactory. |
| Review | Comments from users describing how satisfied they are with a certain clinic. |
| User | A person who uses the application to look for clinics. |
| Search | Search for clinics based on user’s given input. |
| Km | A unit of length in the metric system: Kilometer. |
| Ms | A thousandth of a second. |

Appendix B: Analysis Models

B-1 (Use Case)



B-2 (Use Case Descriptions)

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | UC001 | | |
| Use Case Name: | Register | | |
| Created By: | Jing Xiang | Last Updated By: | Samuel |
| Date Created: | 03-09-19 | Date Last Updated: | 26-10-19 |

|  |  |
| --- | --- |
| Actor: | User, Google |
| Description: | Register a new user account (local and google) |
| 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: | Low |
| Frequency of Use: | Low |
| Flow of Events: | Local Register:   1. The user inputs new user id and password 2. The user selects “CREATE ACCOUNT” 3. The application validates the user details 4. The user account is created 5. The application displays a successful message 6. Use case ends   Google Register:   1. The user click the google button for registration 2. The user inputs google email address and password 3. The user selects “LINK ACCOUNT” 4. The application validates the user details 5. The user account is granted permission 6. The application displays a successful message 7. 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)   If user inputs empty fields   1. The application displays an error message 2. The application returns to step 2) |
| Exceptions: | 1. If Google or Local Server is down. 2. 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: | Jing Xiang | Last Updated By: | Samuel |
| Date Created: | 03-09-19 | Date Last Updated: | 26-10-19 |

|  |  |
| --- | --- |
| Actor: | User, Google |
| Description: | Perform login procedure to allow user to rate specific CHAS clinic and give a review |
| Preconditions: | User must be a registered user with the application |
| Postconditions: | 1) User is successfully logged in and moved to the Home page  2) Home page displays search bar and map view of clinic details |
| Priority: | Medium |
| Frequency of Use: | Medium |
| Flow of Events: | 1. The user selects “Login” 2. The user inputs email address and password 3. The user clicks “Ok” 4. The application validates the user details 5. The application moves to the profile page 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)   If user inputs empty fields   1. The application displays an error message 2. The application returns to step 2) |
| Exceptions: | EX1: If user is not registered   1. The user starts from step 1) of “Register” use case (UC001) |
| 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: | Jing Xiang | Last Updated By: | Jing Xiang |
| Date Created: | 03-09-19 | Date Last Updated: | 10-09-19 |

|  |  |
| --- | --- |
| 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 in to 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: | Rate Clinic’s Score | | |
| Created By: | Jing Xiang | Last Updated By: | Samuel |
| Date Created: | 03-09-19 | Date Last Updated: | 26-10-19 |

|  |  |
| --- | --- |
| Actor: | User, Google |
| Description: | The use case allows user to rate specific CHAS clinic |
| Preconditions: | The user is login and on a CHAS clinic page after searching |
| Postconditions: | Give a one to five star rating |
| Priority: | Medium |
| Frequency of Use: | Medium |
| Flow of Events: | 1. The user taps on “rating” 2. The user selects the amount of star rating to give 3. The user must write a review 4. User clicks submit 5. Review is saved 6. Use case ends |
| Alternative Flows: | If user is not logon   1. Application will provide an informative message to the user asking them to login 2. User login 3. The application returns to step 2) |
| Exceptions: | 1. If User has already rated and reviewed the clinic, rating it again will remove the old rating given by the user. Replacing it with the new one. |
| Includes: | Login |
| Special Requirements: | - |
| Assumptions: | All Servers are up and running with no connection problems. |
| Notes and Issues: | - |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | UC005 | | |
| Use Case Name: | Search Clinic | | |
| Created By: | Jing Xiang | Last Updated By: | Samuel |
| Date Created: | 03-09-19 | Date Last Updated: | 26-10-19 |

|  |  |
| --- | --- |
| Actor: | User |
| Description: | The use case allows the user to search for a CHAS clinic with postal code, telephone number, name, type or distance |
| Preconditions: | The user has downloaded and started the application |
| Postconditions: | Display CHAS clinic if match |
| Priority: | High |
| Frequency of Use: | High |
| Flow of Events: | Map Search :   1. The user selects the search bar 2. The user inputs the details 3. The user press search 4. Map will plot markers based on search results 5. Tapping markers will forward users to Clinic Details (UC008) 6. Use case ends   Clinic List Search :   1. The user selects the search bar 2. The user inputs the details 3. The user press search 4. Clinic List will filter the existing list with the search query 5. The List will be updated with searched results. 6. Tapping on any results will forward users to Clinic Details (UC008) 7. Use case ends |
| Alternative Flows: | No matching clinic   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 clinics 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 clinics 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: | UC007 | | |
| Use Case Name: | View List of Clinics | | |
| Created By: | Jing Xiang | Last Updated By: | Samuel |
| Date Created: | 03-09-19 | Date Last Updated: | 26-10-19 |

|  |  |
| --- | --- |
| Actor: | User, Google |
| Description: | The use case allows the user to view the names of all the list of clinics before deciding what action to take next |
| Preconditions: | The user has downloaded and started the application |
| Postconditions: | The application displays view list of clinics page |
| Priority: | High |
| Frequency of Use: | High |
| Flow of Events: | 1. The user selects “List” 2. The application displays view list of clinics page 3. Use case ends |
| Alternative Flows: | - |
| Exceptions: | - |
| Includes: | View Clinic Information |
| Special Requirements: | - |
| Assumptions: | All Servers are up and running with no connection problems. |
| Notes and Issues: | - |

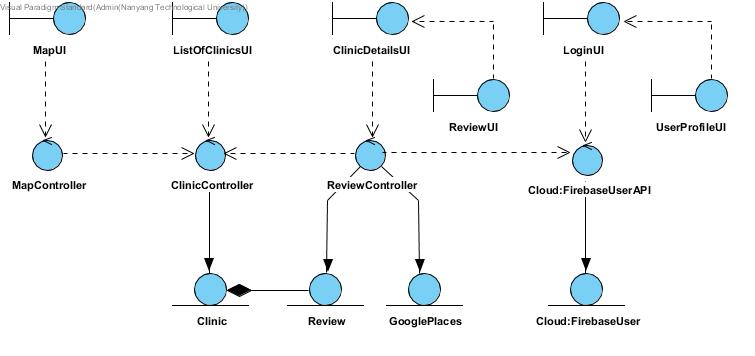
|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | UC008 | | |
| Use Case Name: | View Clinic Information | | |
| Created By: | Jing Xiang | Last Updated By: | Samuel |
| Date Created: | 03-09-19 | Date Last Updated: | 26-10-19 |

|  |  |
| --- | --- |
| Actor: | User, Google |
| Description: | The use case allows the user to view the information about a clinic |
| Preconditions: | The user has successfully searched for a clinic and click on it, or user click on one of the clinics in list of clinics |
| Postconditions: | The application displays a view clinic information page |
| Priority: | Medium |
| Frequency of Use: | Medium |
| Flow of Events: | 1. The user selects a specific clinic 2. The application displays a specific clinic page with the following information:  * Name * Address * Rating Score * Review * Telephone Number * Operating Hours * Type  1. Use case ends |
| Alternative Flows: | - |
| Exceptions: | - |
| Includes: | - |
| Special Requirements: | - |
| Assumptions: | All Servers are up and running with no connection problems. |
| Notes and Issues: | - |

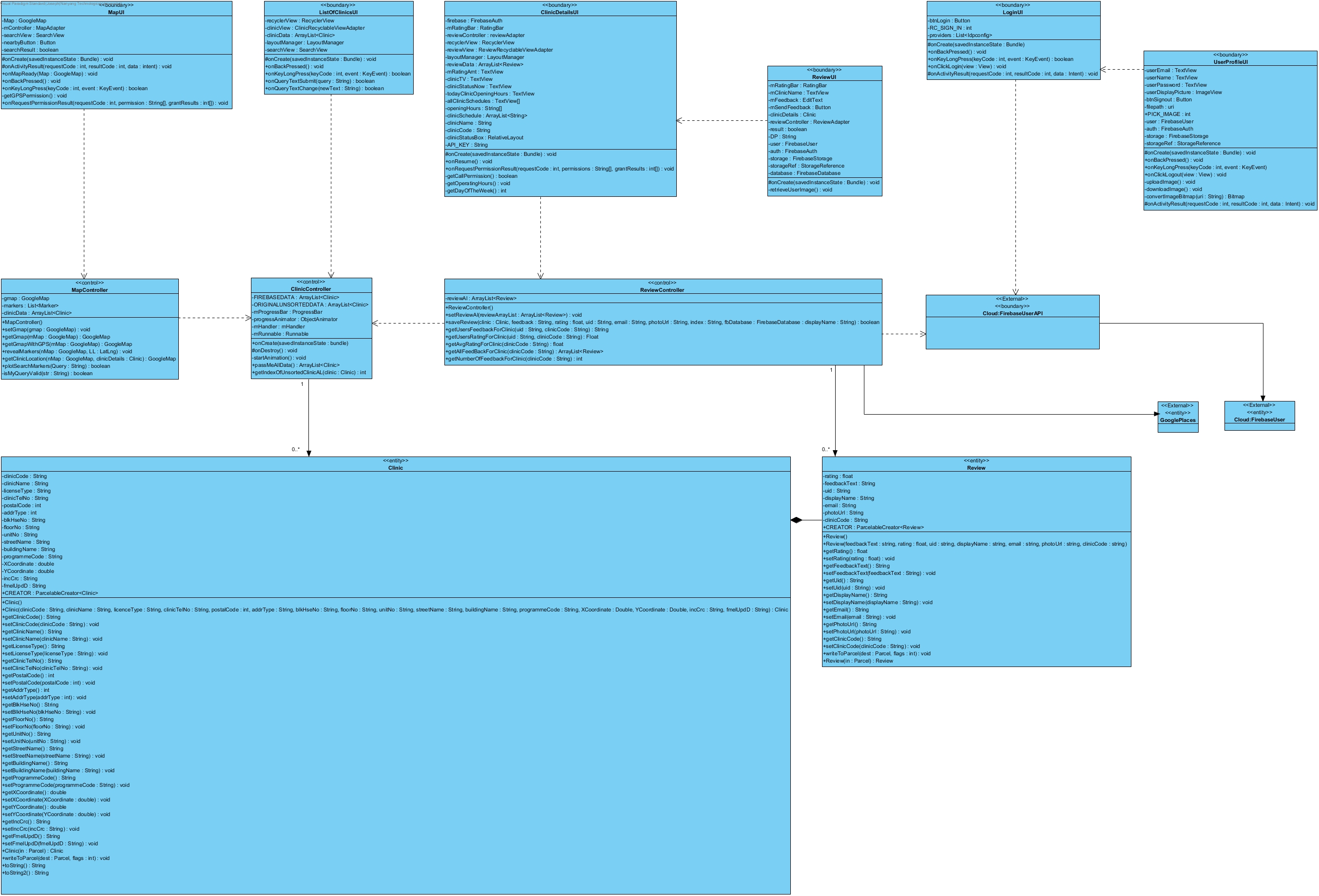
|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | UC009 | | |
| Use Case Name: | Call Clinic Telephone Number | | |
| Created By: | Jing Xiang | Last Updated By: | Samuel |
| Date Created: | 03-09-19 | Date Last Updated: | 26-10-19 |

|  |  |
| --- | --- |
| Actor: | User, Google |
| Description: | The use case allows the user to call the clinic |
| Preconditions: | The user has successfully entered the view clinic information page |
| Postconditions: | The user has successfully called the clinic with the Telephone Number provided |
| Priority: | Medium |
| Frequency of Use: | Low |
| Flow of Events: | 1. The user is viewing a specific clinic 2. The user clicks on the Telephone Number 3. The user will be prompted to make a call 4. Use case ends |
| Alternative Flows: | - |
| Exceptions: | EX1: First time users depending on android version might be prompted to choose to use which application to call the Telephone Number  OR  EX2: User cancels the Telephone Number   1. The application returns back to view clinic information page |
| Includes: | - |
| Special Requirements: | User must be using any device with a valid SIM Card with phone number. |
| Assumptions: | All Servers are up and running with no connection problems.  User are using a phone. |
| Notes and Issues: | - |

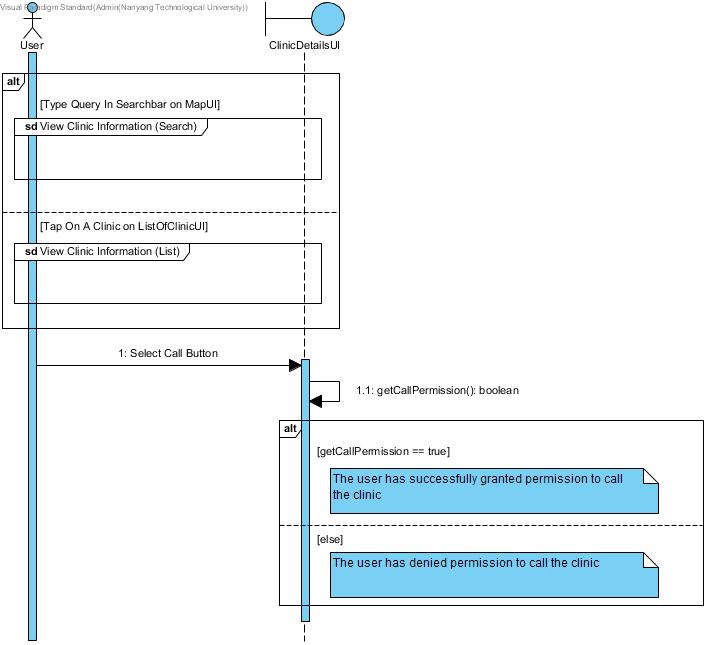
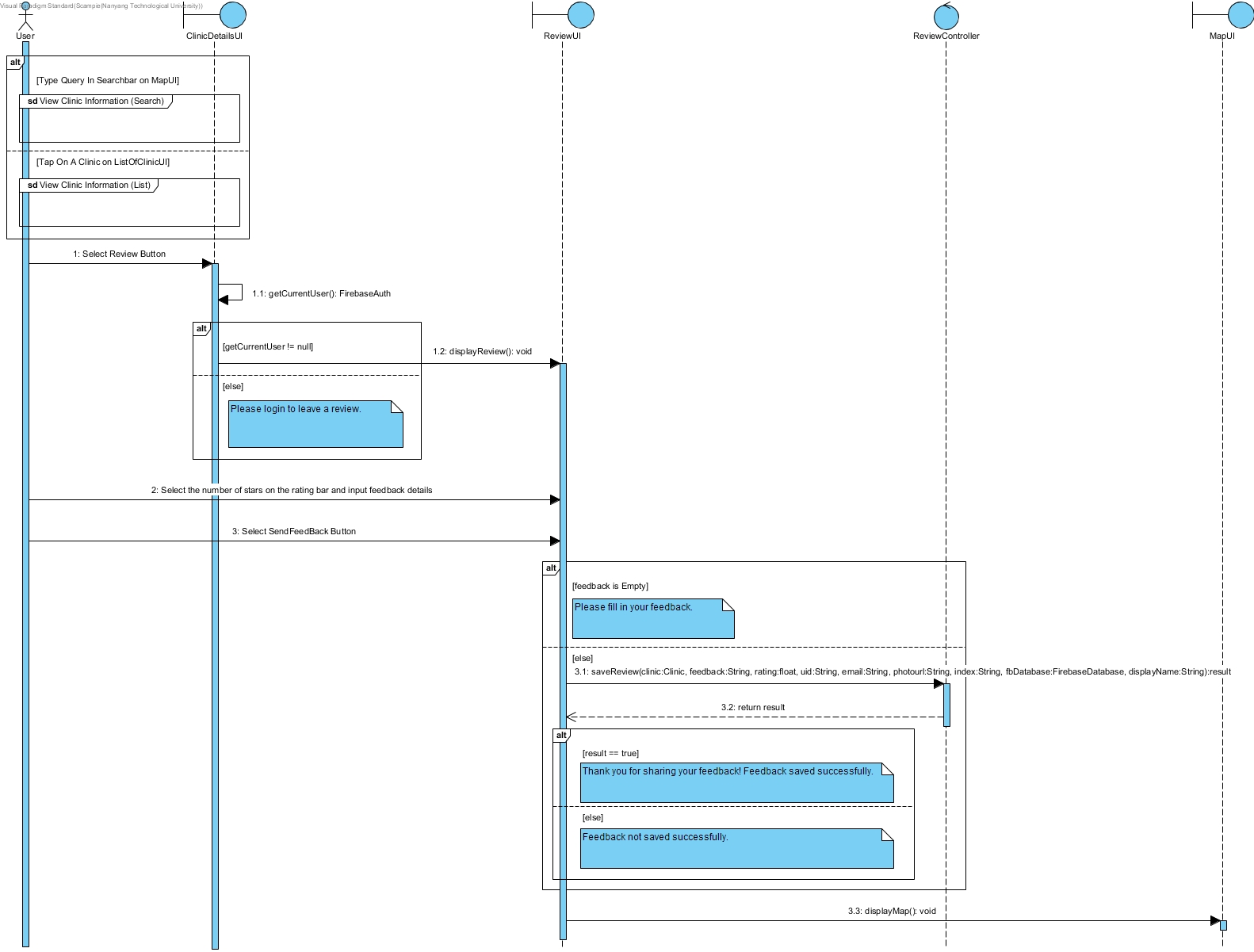
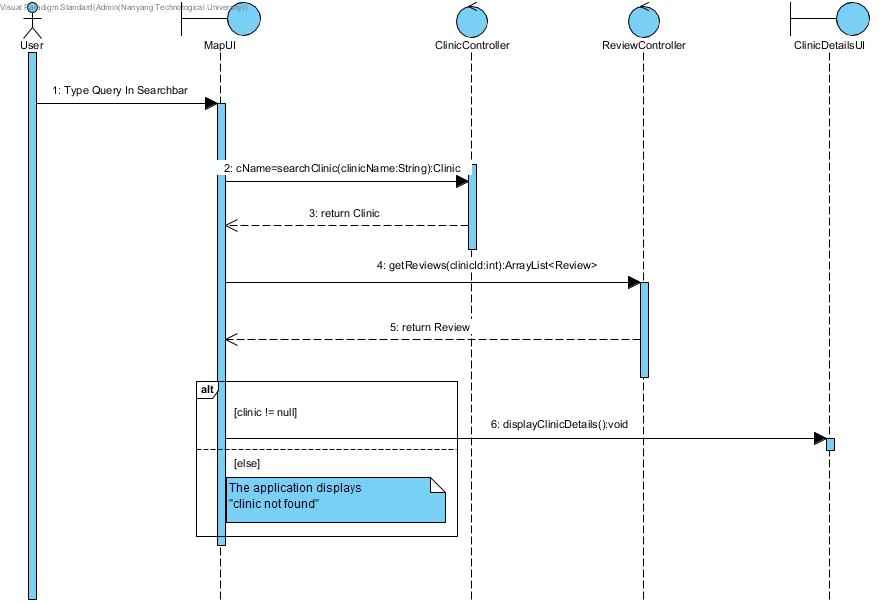
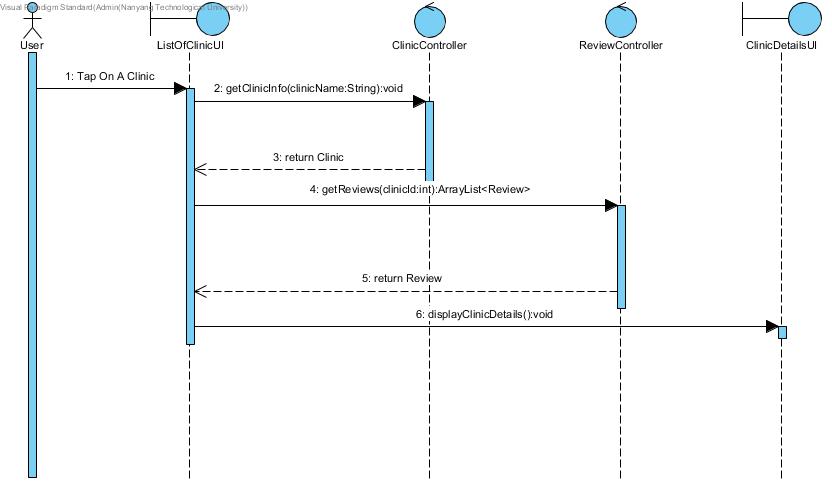
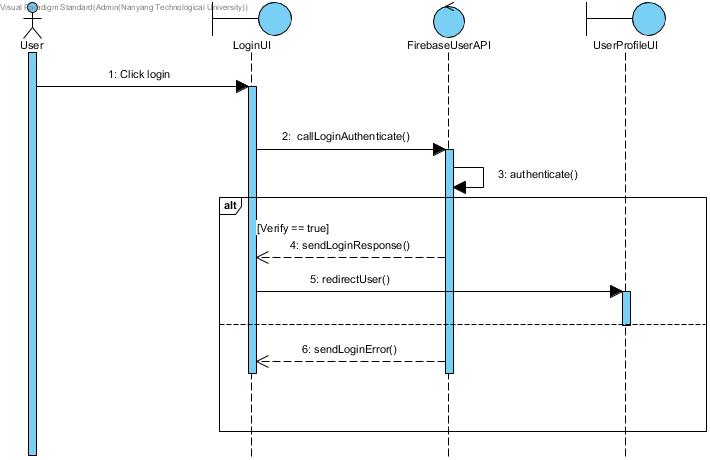
B-3 (Boundary Class Diagram)



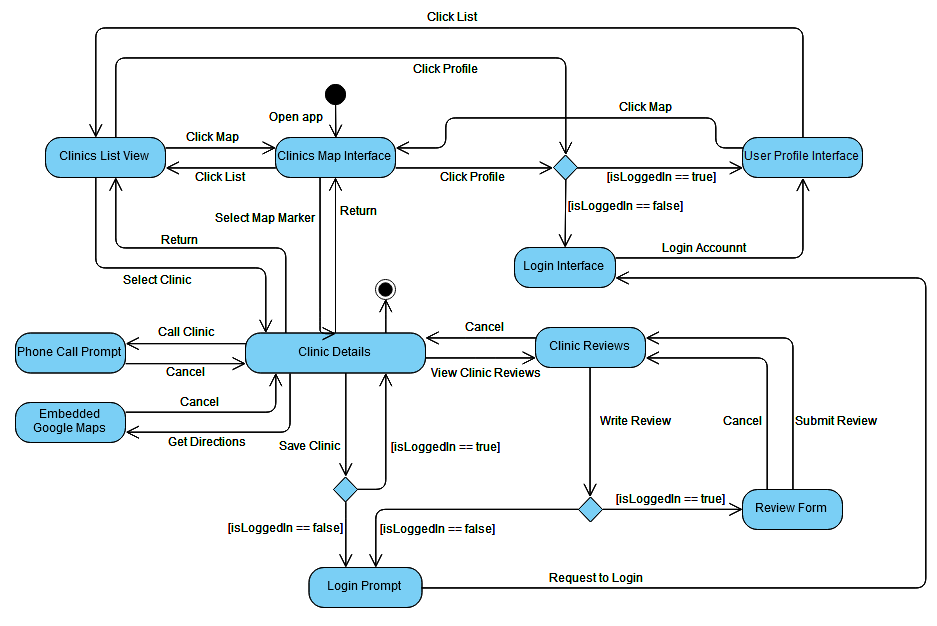
B-4 (Entity Class Diagram)



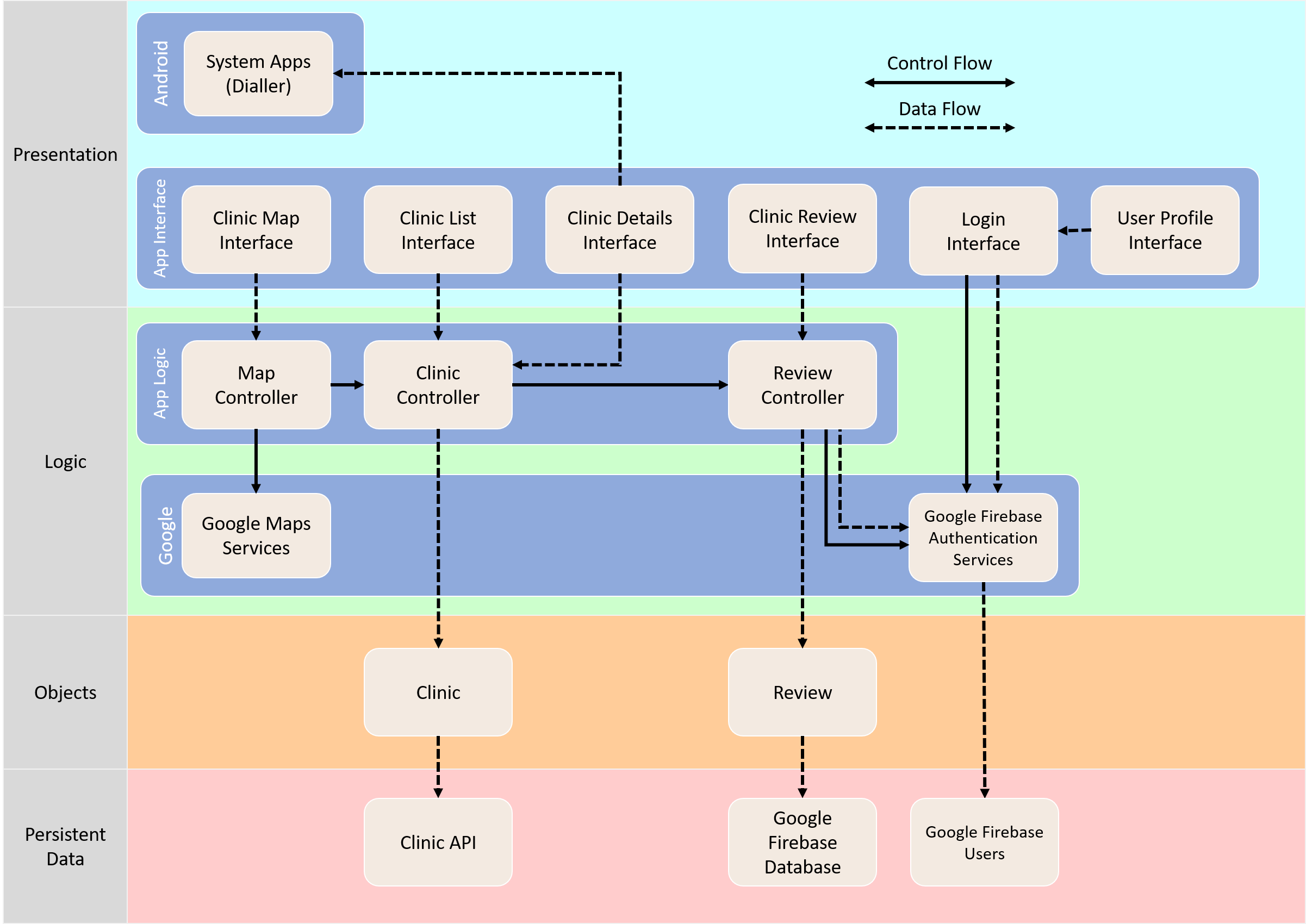
B-5 (Sequence Diagrams)



B-6 (Dialog Map)



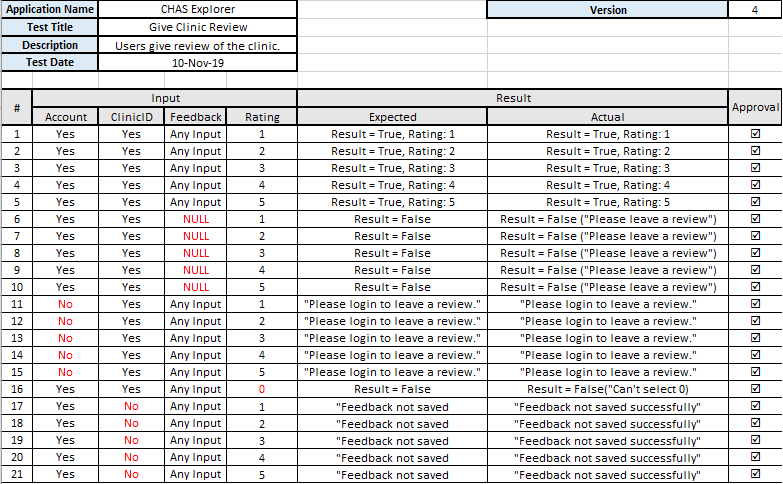
B-7 (System Architecture)



B-8 (Black Box Testing)

**Testing “Give Clinic Review” Control Class**

|  |  |  |
| --- | --- | --- |
| **Field** | **Valid Test Case** | **Invalid Test Case** |
| **Rating** | * Any Number from 1 to 5 | * 0 |
| **Feedback** | * Any input | * Null Feedback |
| **User** | * Already sign in account | * Haven’t sign in account |
| **Clinic** | * Have select this clinic | * Does not select any clinic |

****

B-9 (White Box Testing)

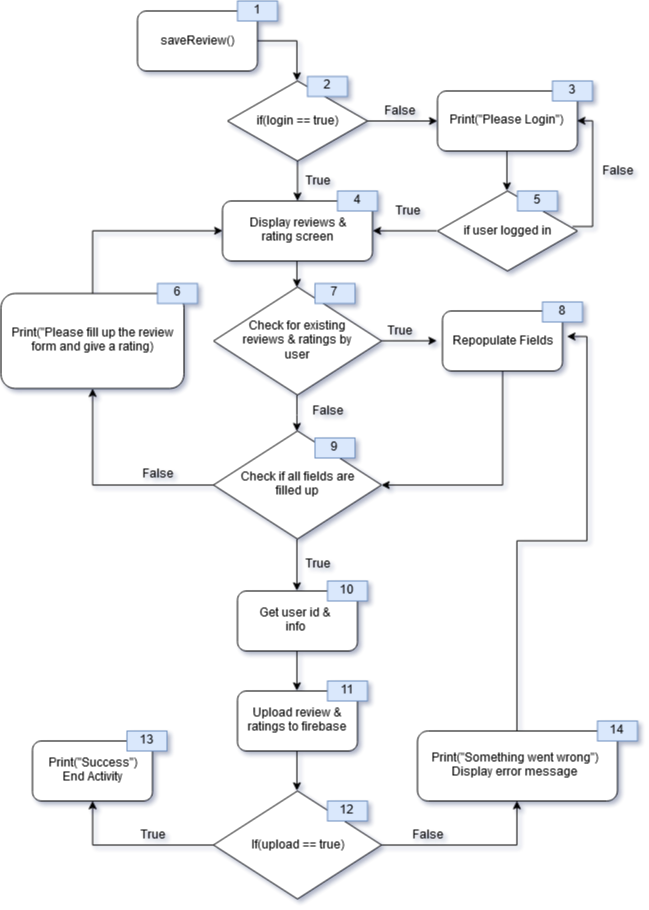
**Control Flow Test: Save Review**

Method: saveReview()

Parameters: Clinic, feedback, rating, uid, email, photo, firebaseconnection, displayname

Return: boolean

Purpose: upload and save the review into database

****

**Basis Paths:**a) 1,2,4,7,9,10,11,12,13

b) 1,2,3,5,4,7,9,10,11,12,13  
c) 1,2,4,7,8,9,10,11,12,13  
d) 1,2,4,7,9,5,4,7,9,10,11,12,13  
e) 1,2,4,7,9,10,11,12,14,8,9,10,11,12,13

**Test Cases:**a) User is logged in submits a new review without existing review found attached to user id.  
b) User is not logged in, logs in successfully and submits a new review.  
c) User is logged in and submit a review, but an existing review is found and fields are repopulated accordingly.  
d) User is logged in and submit a new review but user tries to submit an empty form.  
e) User is logged in and submit a new review, however for some reasons the server fails or form is incomplete and user is prompted to try again.  
  
**Execution Paths:**a) 1,2,4,7,9,10,11,12,13

b) 1,2,3,5,4,7,9,10,11,12,13  
c) 1,2,4,7,8,9,10,11,12,13  
d) 1,2,4,7,9,5,4,7,9,10,11,12,13  
e) 1,2,4,7,9,10,11,12,14,8,9,10,11,12,13

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

**Control Flow Test: Search Clinic on Map**

Method: searchClinicOnMap ()

Parameters: Query

Return: boolean

Purpose: Search Clinic by Clinic Name/Phone Number/Postal Code and show Clinic location on Map

**A close up of a map

Description automatically generatedBasis Paths:**a) 1,2,4,5,6,7,9,10,6,11,12

b) 1, 2, 3

c) 1,2,4,3

d) 1, 2, 4,5,6,7,10,6,11,3

e) 1, 2, 4,5,6,8,10,6,11,3

f) 1,2,4,5,6,8,9,10,6,11,12

**Test Cases:**a) User types “Peace” and press search, all the Clinics which contains the keyword “Peace” in their Clinic name will be plotted on the map.

b) User types nothing and just press search, “No Results Found!” will be printed.

c) User types “@#Peace@Clinic!” and press search, “No Results Found!” will be printed.  
d) User types “Harvard Clinic” and press search, no clinic containing the keyword in their Clinic Name is found after looping through the data set. “No Results Found” will be printed out.

e) User types “712345” and press search, no clinic containing the keyword in their Postal Code is found after looping through the data set. “No Results Found” will be printed out.

f) User types “732899” and press search, the Clinic with that Postal Code will be plotted on the map.

**Execution Paths:**a) 1,2,4,5,6,7,9,10\*,6,11,12

b) 1,2,3

c) 1,2,4,3

d) 1,2,4,5,6,7,10\*,6,11,3

e) 1,2,4,5,6,8,10\*,6,11,3

f) 1,2,4,5,6,8,9,10\*,6,11,12

\***While loop (Number 6) will be repeated 1063 times since there are 1063 Clinics in the given dataset.**

**Cyclomatic Complexity:**[Conditions + 1] = [6 + 1] = **7**   
[Edges - Nodes + 2] = [17 – 12 + 2] = **7**