UTS

SOFTWARE DESIGN DOCUMENT

31285 Mobile Application Development

SEONYOUNG MUN

Table of Contents

1 Introduction

1.1 Document Purpose

This document presents the design and specification of a new Android mobile application, 'Whose Treat (tentatively named)'. The aim of the document is capturing the design constraints and assumptions as well as demonstrating the detailed specifications of the app; both functional and non-functional requirements and specifications will be discussed and the corresponding diagrams will be presented for visual assistance.

1.2 Project Purpose and Application Vision

The new mobile app 'Whose Treat' is to assist individuals and groups selecting lunch menu or shops to visit easily and in an entertaining way. This app is targeting the young adults aged between 18 to 35 who want to navigate the information regarding the menu choice for their short break or lunch time easily and quickly without being disturbed by overwhelming information.

The target platform and device for the initial application is Android smartphones; the minimum SDK is API 15 which can cover 97.4% of all Android users. Fragments which take Android tablets into consideration could be implemented in the initial version of the app.

1.3 Project scope

Whose Treat will provide three main functionalities; 1) the random selection of the person who treats for the group and the shop in which the group or individuals can have meal or drinks, 2) the suggestions for the shop based on users' selection and 3) the collection of information regarding the user's preference and use history for the better user experience.

If the application implements the additional functionalities, it can be functioning as a revenue model by encouraging the local shop owners to promote their business on the application and allowing them to form a strategic alliance with the company or with other adjacent business owners. The details of the additional functionality that can be implemented if time and resources allow are discussed in Section 2.

Document Revision History

Revision History

Date	Version	Description of Change
04/05/2017	0.5	Skeleton of the document Diagrams are made
05/05/2017	0.8	Revision
07/05/2017	1.0	First Version Signed Off
09/05/2017	1.1	Use case changed
10/05/2017	1.2	Sequence diagram update
19/05/2017	1.6	Feature summary added
20/05/2017	1.7	Testing updated
21/05/2017	1.8	Revision
22/05/2017	2.0	The second Version Signed Off

Glossary

Term	Description / Meaning
APP	Application In this document, particularly, refer to 'Whose Treat'
API	Application programming interface
UI	User Interface

2 Functionality Overview

This section will describe the functionalities and features of the application. Section 2.1 will presents detailed diagrams for a better understanding of written description of the app in following sections.

2.1 Diagrams

2.1.1 Use Case Diagram

The use case diagram of Whose Treat is illustrated in Figure 1. The corresponding use case items and the description are shown in Table 1.

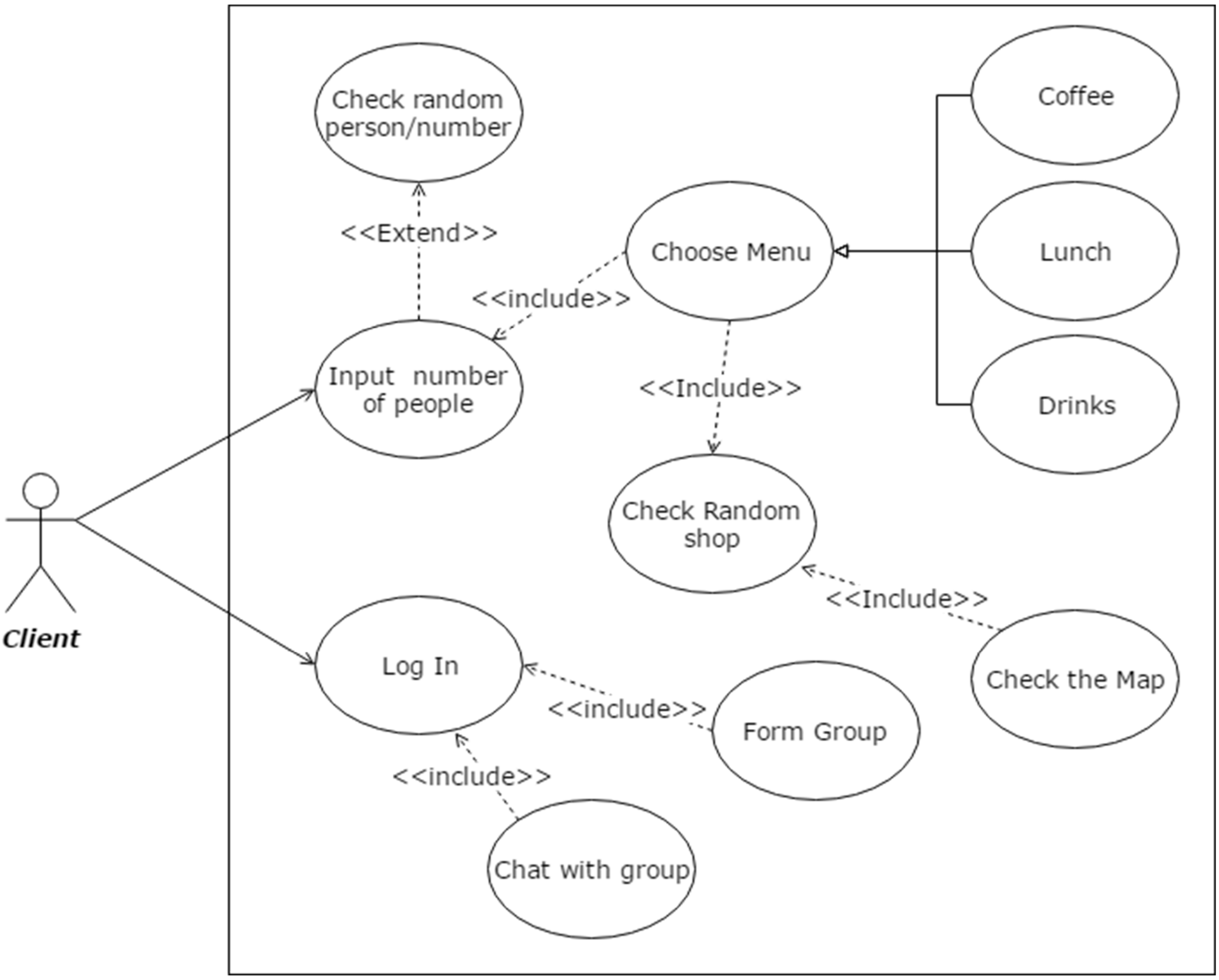


Figure 1 Use Case Diagram of the Application

Item	Description
Input number of people	The user provides the number of people for the polling. If there is no input, the app will assume that there is only one person
Check random person/number	Extending class of 'Input number of people' use case. The user can check the random result of person/number who need to pay.
Choose menu	Selecting the menu category for narrowing the search. This use case includes 'Choose Coffee', 'Choose Lunch', 'Choose Drinks' use cases.
Check random shop	Include use case of 'Choose menu. The user can check the randomly selected shop within 200M radius. If there is no search result available, the system will ask the user whether they want to expand the search to 200 more meters.
Check the map	Include use case of 'Check random shop. By clicking the 'Check the Map button, user can navigate the map (Google map)
Log in	Logging into the system for the better user experience. The user can log in either Google ID, email or phone number. This use case will include 'Chat with group', 'Form group' use cases.

Table 1 Use case description of Whose Treat

2.1.2 Sequence Diagram

Figure 2 Shows the sequence diagram of Whose Treat application. The login function is not included here as it would be excluded from the first release.

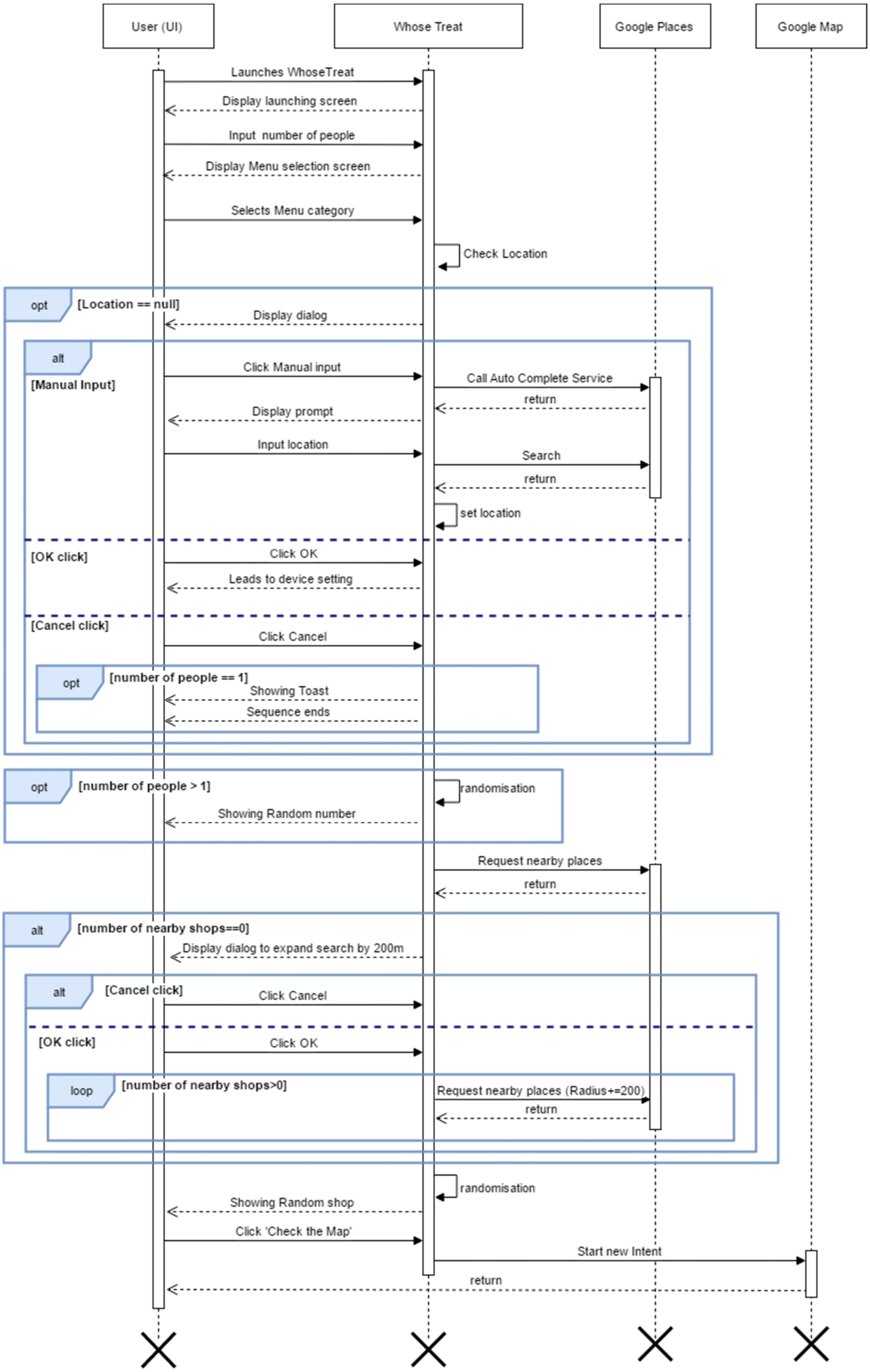


Figure 2 Sequence Diagram of Whose Treat

2.2 Feature Summary

As the application is not fully developed yet, some of the features are yet implemented and could be excluded for the first release depending on the time frame.

2.2.1 Implemented Features

The features that are definitely included and already implemented are as follows.

- Random selection of the number/person who will pay for the group
- Random selection and suggestion of the shops based on the user's choice and location
- Map intent that is showing the location of the selected shop
- Storing username by SharedPreference

Random selection of the number/person feature is operated by user's input. In the launching activity, the user will be asked the number of people. If the user clicks the button without any input or passes '1', the app will assume that there is only one person, hence, will not return a random number selection screen. Otherwise, the app will show the random selection result for the person after the user selects the menu category. At the moment, the names are hard-coded, therefore, the random number is also returned. However, once the authentication is fully implemented, the user can form a group and store the names of the members. Authentication is under development and is for Firebase service.

Random selection and suggestion of the shops feature is working based on the user's menu selection and location. If user's current location is unknown, the app will prompt a dialogue asking whether the user wants to turn on the GPS on the device or manually provide their location. If user clicks 'OK' button of the dialogue, the app will open the setting screen of user's device allowing the user to turn on the GPS. In the case of 'Manual Input' option, Google Places API Autocomplete service intent will be called and displayed in Overlay mode and allow user to put their place accurately. The user can click 'Cancel' as well so that he/she can just randomise person to buy without place suggestions. This will check one more logic that is the number of people; if there is only one person in the poll, it will toast a message saying "It is your own treat ☺", and if in another case, will show the randomised result for a person treating the group.

In addition, the random place feature will return the shops within a 200-meter radius of the user's current location by default. However, if there are no available shops within that range, the app will ask the user whether he/she wants to expand the search to 200 more meters. The app will keep on asking that until it gets at least one available shop.

Map intent feature will be launched when the user clicks “Check the Map” button at the randomly selected shop result screen. It will show the pinned place that has been suggested randomly in the previous screen.

2.2.2 Features Under Development

The features that would be implemented in the first release are as follows.

- Authentication by Google account, Email or Phone number
- Chatting feature between users
- Counting the number of treats and summing up the amount of payment made

Authentication is for using Firebase to store user information and group information. At this moment, authentication by Google account is implemented. However, if time is insufficient, the feature will be excluded from the first release and will be added to the next version.

The features that will be included in the next release of the app are as follows.

- Displaying results of all available shops
- Web bulletin board system that allows the user and local shop owners can interact with each other which mainly about promotion information.
- Location-based notification service alarming users about instant deals or other information.
- Social media service which allows users to arrange instant meeting based on the location or preferences.

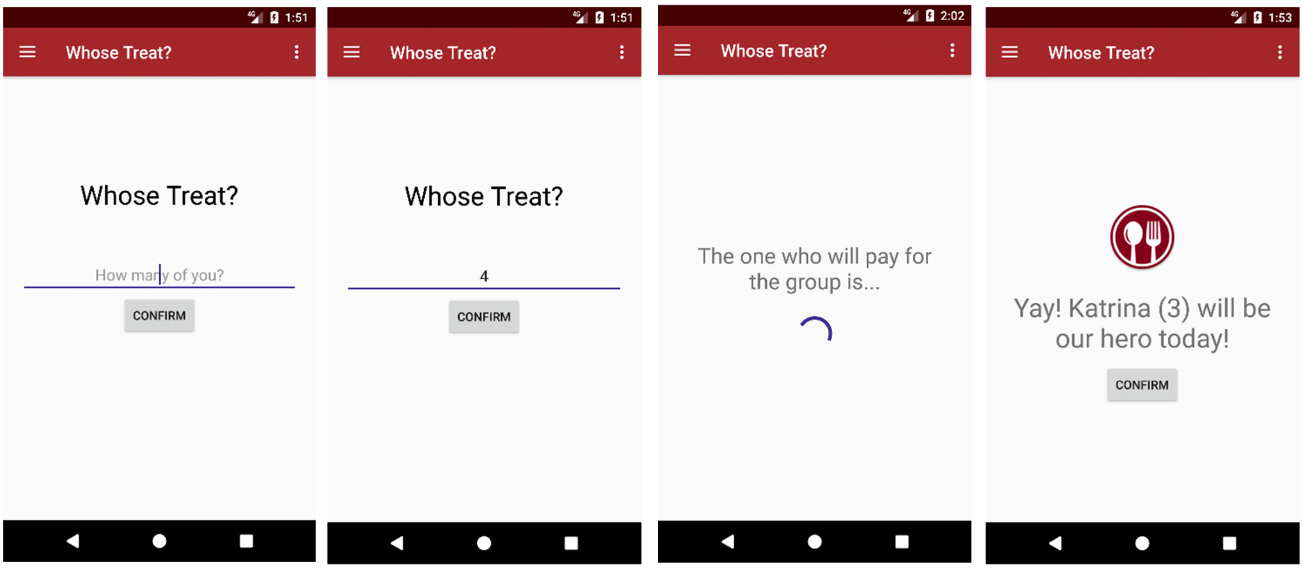
3 Non-Functional Requirements and Features

3.1 User Interface

User interface (UI) design will be conformed to the material design specification which aims for devising visual language that can convey meaningful messages by good and intentional design.

The UI design will be applied in a user-friendly and intuitive manner; most of the screens will be comprised of a few buttons and selections. Also, too many colours use will be avoided preventing users overwhelmed by the complicated display.

Figure 3 shows the flow of random selection of person/number; the user can enter the number of people or just click confirm button if the randomisation of person is not needed.

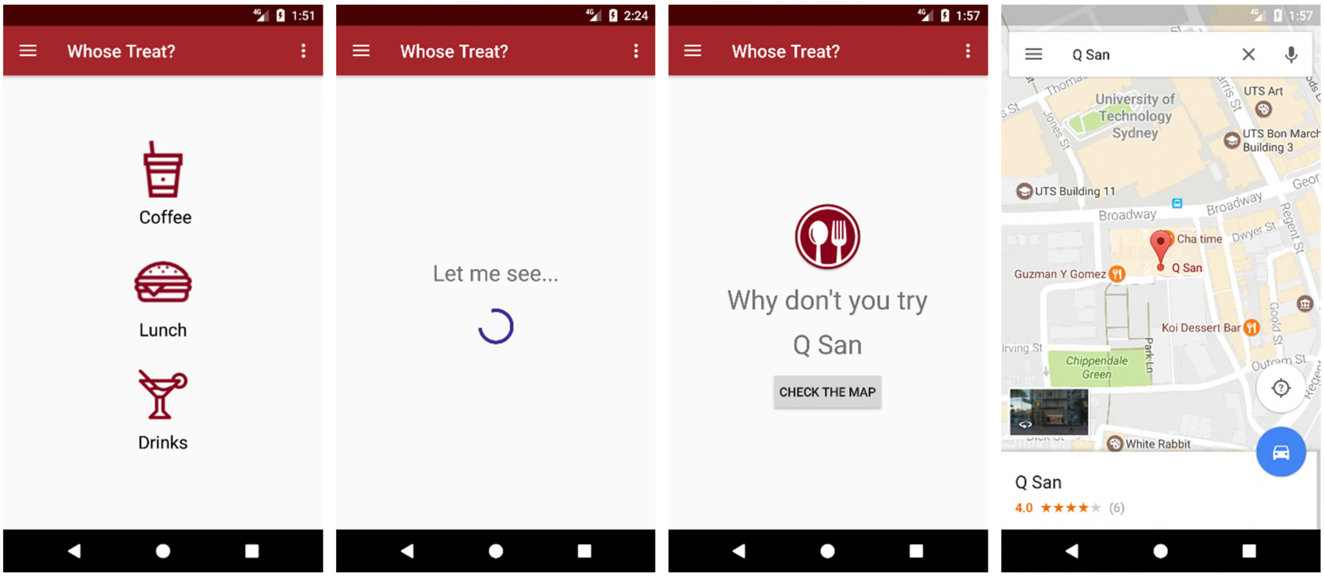


The one who will pay for the group is...

Yay! Katrina (3) will be our hero today!

Figure 3 Random Selection of Person Flow

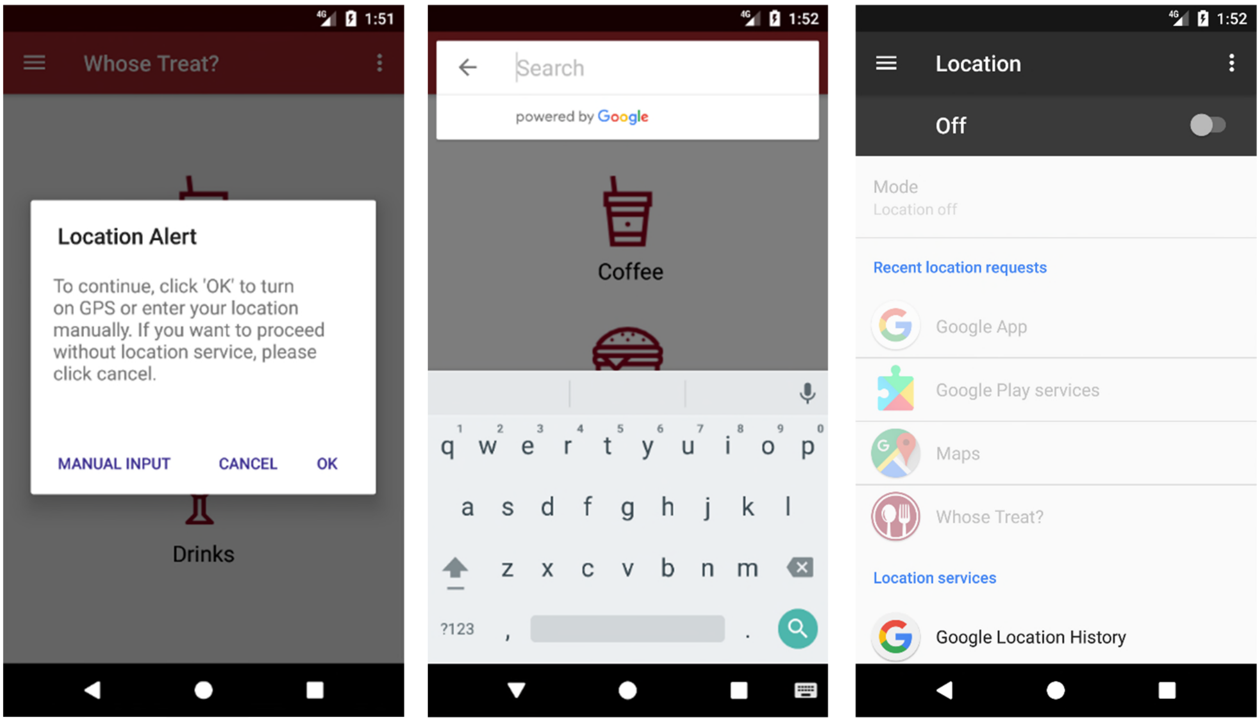
Figure 4 represents the Random Shop selection flow. If there is no current location found, the app will prompt a dialogue allowing the user to select options (Figure 5). Depending on the options user selected, the app will show relevant screens or proceed to the random number fragment. Figure 6 illustrates the 'Cancel' flow; the user can proceed to the random number fragment if the number of people is more than 1, however, cannot see the confirm button which can lead he/she to the random shop suggestion fragment. If the number of people is '1', the user will see the Toast message.



Let me see...

Why don't you try Q San

Figure 4 Random Selection of shops Flow

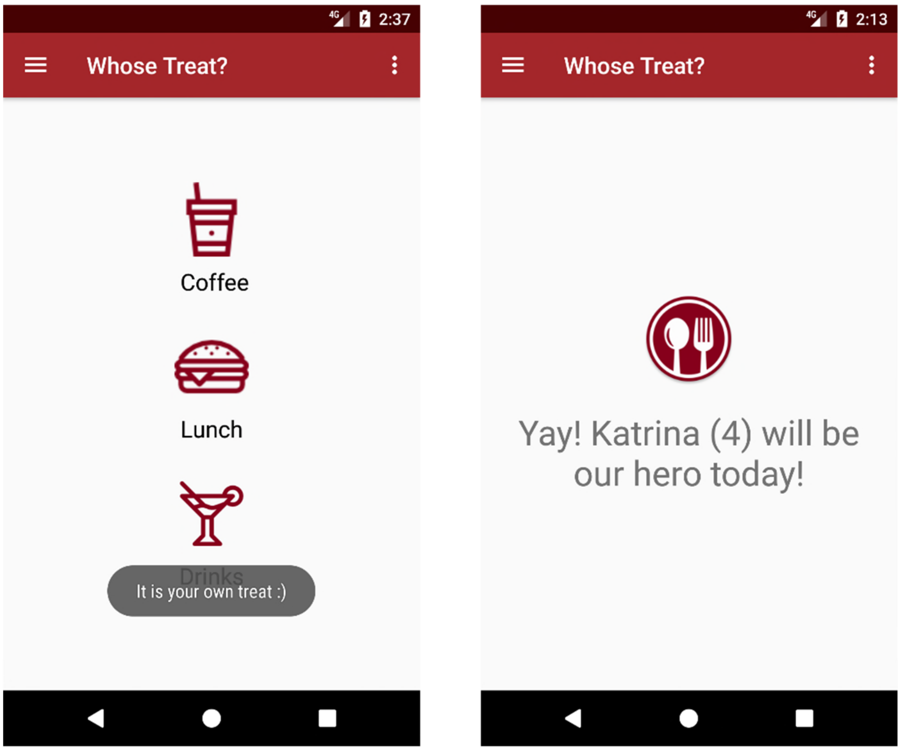


Location Alert

To continue, click 'OK' to turn on GPS or enter your location manually. If you want to proceed without location service, please click cancel.

MANUAL INPUT CANCEL OK

Figure 5 Unknown Location Alert (left): onManualCity (middle) and onOkayClick (right)

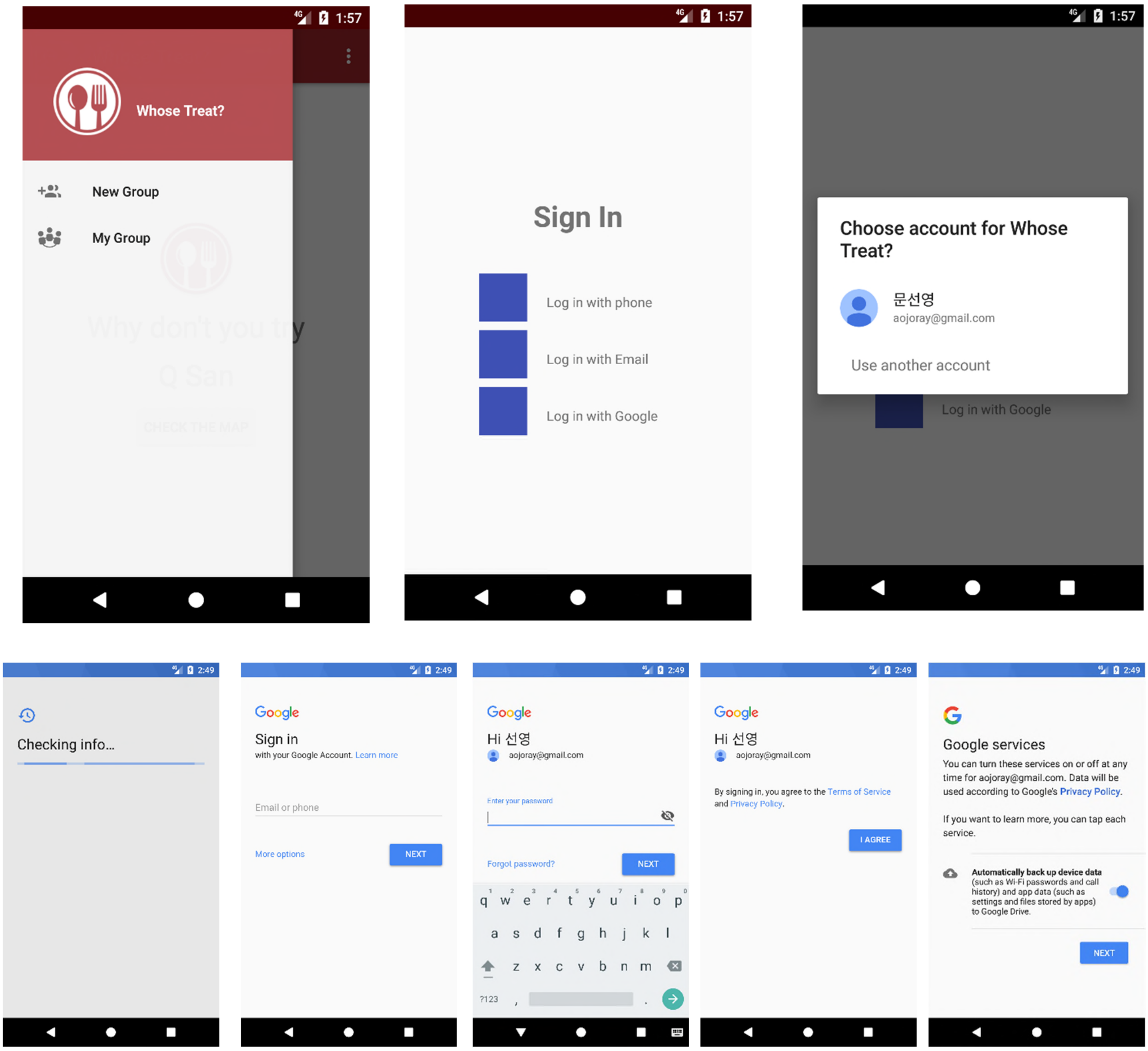


It is your own treat :)

Yay! Katrina (4) will be our hero today!

Figure 6 Unknown Location onCancelClick event

One of other features that are under on-going developments is authentication for group chat and organisation (Figure 7). This feature would probably be excluded in the first release. Currently, authentication with Google account has been implemented, however, no user interface afterwards.



Choose account for Whose Treat?

문선영
aojoray@gmail.com

with your Google Account. Learn more

Hi 선영
aojoray@gmail.com

Enter your password

By signing in, you agree to the Terms of Service and Privacy Policy.

You can turn these services on or off at any time for aojoray@gmail.com. Data will be used according to Google's Privacy Policy.

If you want to learn more, you can tap each service.

Automatically back up device data (such as Wi-Fi passwords and call history) and app data (such as settings and files stored by apps) to Google Drive.

Figure 7 Authentication with Google Account

3.3 Used Application Programming Interfaces

The APIs, libraries and services used for the app are as follows:

- Google Places API: to retrieve information regarding nearby places as well as getting users' location
- Google Paly Services (auth, maps, places and location): to track user's location
- Gson: for parsing JSON result returned from Google Places API
- Firebase and Firebase UI: for user authentication and storing information (not fully implemented yet)
- MyFirebaseMessagingService: for allowing users to chat each other (not fully implemented yet)

4 Application Structure

4.1 Packages and Activities

The package structure of the app and the description of the activities, fragments and other included in each package are shown in Table 2.

Package Name	Contents
Activities	This package contains the activities of the app • MainActivity: launching activity displays the user input screen • PreferenceActivity: activity for preference changes • SigninActivity: for authentication to Firebase (would be excluded) • ChatActivity: for chatting between users (would be excluded)
ActivityFragments	This package contains the fragments of the app. First three classes are forming a main flow of the app, and the last three is for Firebase feature which could be excluded from the first release. • MenuSelectionFragment: showing user the categories of menu • RandNumFragment: showing user the random result of number/person • WhereToGoFragment: showing user the random result of shop suggestion • MyGroupFragment: for browsing formed group (would be excluded) • NewGroupFragment: for adding new group (would be excluded) • ChatFragment: for displaying chatting room UI (would be excluded)
Dialog	This package contains the fragments for dialogues. • LocDialogFragmenFragmentt: dialogue for the situation in which the user's location is unknown. • NewSearchDialogFragment: dialogue for asking user whether she/he wants to expand search area when there is no result available.
JavaHelperClasses	This package contains helper classes for activities and fragments. • JsonResultToJava: parsing JSON result to Java object. This class will be used for presenting the list of available shops in a later version of the app as well. • Person: representing an individual user of the app. It would be used only temporarily before implementing the Firebase feature.

Table 2 The Packages of WhoseTreat?

5 Data Structures

The application is not using internal storage mechanism currently. However, if the Firebase based feature is implemented, user information will be stored accordingly. The assumed data dictionary for the Person object representing individual user and group entity are shown in Table 3

Person Entity

Attribute	Data Type	Description
Id	Integer	The primary key of the entity. Unique identifier of users / not null
username	Varchar 50	The name of user / not null
Count	Integer	The count of treat user made for the group
amount_paid	Double	The amount of money paid by user
group_id	Integer	The foreign key (the primary key of group entity)

Group Entity

Attribute	Data Type	Description
Id	Integer	The primary key of the entity. Unique identifier of group / not null
group_name	Varchar 50	The name of group
count	Integer	The count of gathering
amount_paid	Double	The amount of money paid by all members
member_id	Integer	The foreign key (the primary key of person entity)

Table 3 Data Dictionary for Person and Group entity

6 Testing

The app is tested on physical devices along with the virtual devices over the whole development process. The initial release version of the app should not fail to the tests on both circumstances. The functions implemented are tested during the testing. Table 4 and Table 5 show the result of testing on both physical and virtual device

Task	Errors	Result
Input number of people	No	Pass
Preference Change Listener	No	Pass
Check random person/number	No	Pass
Choose menu	No	Pass
Check random shop	No	Pass
Check the map	No	Pass
Open Settings	No	Pass
Manual Location	No	Pass
Cancel Handling	No	Pass
Log in with Google Account	No	Pass
Log in with Other Options	N/A	N/A

Table 4 Test Result on Physical Device: Samsung Galaxy Note 3 (Android 5.0 / API level 21)

Task	Errors	Result
Input number of people	No	Pass
Preference Change Listener	No	Pass
Check random person/number	No	Pass
Choose menu	No	Pass
Check random shop	No	Pass
Check the map	No	Pass
Open Settings	No	Pass
Manual Location	No	Pass
Cancel Handling	No	Pass
Log in with Google Account	No	Pass
Log in with Other Options	N/A	N/A

Table 5 Test Result on Virtual Emulator: Pixel (Android 7.1.1 / API level 25)

Task	Errors	Result
Input number of people	No	Pass
Preference Change Listener	No	Pass
Check random person/number	No	Pass
Choose menu	No	Pass
Check random shop	No	Pass
Check the map	No	Pass
Open Settings	No	Pass
Manual Location	No	Pass
Cancel Handling	No	Pass
Log in with Google Account	No	Pass
Log in with Other Options	N/A	N/A

Table 6 Test Result on Virtual Emulator: Galaxy Nexus (Android 6.0 / API level 23)

7 Conclusion

The development of WhoseTreat has been challenging yet interesting. The location based services are fully implemented and functioning well currently. However, the additional features such as showing the list of available places allowing users can browse themselves would be considered for a future version of the application.

The group chat feature based on Firebase is the work of priority as this function would probably be the most appealing feature for downloading the app by not an only single person in a group but by everyone.

The most difficult part of development was a time limitation. As a full-time student, having class every day this semester, it was quite challenging to organise time well. However, it was still an enjoyable experience that realising idea into an application with the enormous help of subject coordinator and mentor from Commonwealth Bank.

8 Reference

- Credit of Coffee Icon
  <div>Icons made by <a href="http://www.freepik.com" title="Freepik">Freepik</a> from <a href="http://www.flaticon.com" title="Flaticon">www.flaticon.com</a> is licensed by <a href="http://creativecommons.org/licenses/by/3.0/" title="Creative Commons BY 3.0" target="_blank">CC 3.0 BY</a></div>

- Credit of Lunch Icon
  <div>Icons made by <a href="http://www.freepik.com" title="Freepik">Freepik</a> from <a href="http://www.flaticon.com" title="Flaticon">www.flaticon.com</a> is licensed by <a href="http://creativecommons.org/licenses/by/3.0/" title="Creative Commons BY 3.0" target="_blank">CC 3.0 BY</a></div>

- Credit of Cocktail Icon
  <div>Icons made by <a href="http://www.freepik.com" title="Freepik">Freepik</a> from <a href="http://www.flaticon.com" title="Flaticon">www.flaticon.com</a> is licensed by <a href="http://creativecommons.org/licenses/by/3.0/" title="Creative Commons BY 3.0" target="_blank">CC 3.0 BY</a></div>

- Credit of Title Icon
  <div>Icons made by <a href="http://www.freepik.com" title="Freepik">Freepik</a> from <a href="http://www.flaticon.com" title="Flaticon">www.flaticon.com</a> is licensed by <a href="http://creativecommons.org/licenses/by/3.0/" title="Creative Commons BY 3.0" target="_blank">CC 3.0 BY</a></div>