

National College of Ireland
BSc in Computing
2015/2016

Leon Daly
X12112381
Kingidas@gmail.com

Restaurant Hangout

Technical Report



Table of Contents

Executive Summary	4
1 Introduction	5
1.1 Background	6
1.2 Aims	6
1.3 Technologies	7
1.4 Structure	7
2 System.....	9
2.1 Requirements	9
3 Requirements Specification	9
3.1 Functional requirements	9
3.1.1 Use Case Diagram	9
3.1.2 Requirement 1 User should be able to find restaurant near him/her	10
3.1.3 Requirement 1 Get Information about Location.....	12
3.1.4 Requirement 1 Make a plan	13
3.1.5 Requirement 1 Share a plan after saving	15
3.1.6 Requirement 1 Delete a plan.....	16
3.1.7 Requirement 1 Share existing plan	18
3.2 Non-Functional Requirements	19
3.2.1 Performance/Response time requirement	19
3.2.2 Availability requirement	19
3.2.3 Recover requirement.....	19
3.2.4 Reliability requirement.....	20
3.2.5 Maintainability requirement	20
3.2.6 Portability requirement	20
3.2.7 Reusability requirement.....	20
4 Interface requirements	20
4.1 Design and Architecture.....	21
4.2 Implementation	23
4.3 Testing.....	23
4.4 Graphical User Interface (GUI) Layout.....	24

4.5	Customer testing.....	28
5	Conclusions	29
6	Further development or research.....	30
	References.....	31
	Appendix.....	32
6.1	Project Proposal	32
6.2	Project Analysis and Requirement Specifications Designs	32
6.3	Monthly Journals.....	32

Executive Summary

I will be developing an android application which would include Google Maps API, Facebook API, xml parsing libraries and Google Play Services. My main target of development would be android 5 (Lollipop). But my application would support devices from Android 4 (ICS) to android 6 (Marshmallow).

The application would find the current location of the person using the Internet or GPS and send the estimated coordinates to Google Maps API to detect the location and query for the nearest restaurant then it will fetch the details of that restaurant and use Facebook API to show friend's list and send the collected information to the device calendar application where it would save the calendar entry and would alert on the event time.

1 Introduction

The purpose of this document is to set out the requirements for the development of my Restaurant Hangout android application.

The intended customers are owners of restaurants, clubs and coffee shops and all of the creative people that always look for great places around them for food , Drinks or General meet ups. Our customers must have an android phone in order to use our application.

The objective of this project is to facilitate users to manage their personal plans and to share these plans with their friends on social websites. Restaurant Hangout is an idea to build an Android application to help users to find Locations of interest near them and enable them to plan events for their Meeting or just social get together.

I am here to develop a Restaurant Hangout Application which is able to help users to create and share their plans with friends on social websites. When the user creates an event and sets its day, time and name, it displays the list of nearby locations by sorting the nearest one on top of the list. On choosing the location, all the available details of the food, bar or coffee shop including its contact number, deals, review, and location is displayed.

Once selected, the user can login to his/her Facebook account and publish a status on Facebook, Google plus or send an SMS. When the plan is checked in, it is moved to the completed plan list. However, if the plan is deleted, it is moved to the incomplete plan list.

1.1 Background

Users always appreciate an idea which helps them in their routine life and makes things easier for them i.e. managing budgets, locating places, communicating with many people in same time and so on. People look for recreation especially on weekends after a busy routine week.

Usually people like to go for lunch or dinner on weekends and finding the best and nearest place always seems hectic for them; they call their friends to seek Restaurant suggestions or they even Google it sometimes.

Some people make plans before the weekend and sometimes due to immense work load, they might forget about their plans and commitments with their family members. There are also social types of people whom have always wanted to share their events with their friends i.e. where they are going on the weekends; they might tell via Facebook but that seemed a little hectic to share something by using another platform.

Social websites like Facebook, Instagram, Twitter and many more sites have plummeted isolation of individuals by bringing them closer by proving different features.

The Growing trend of social websites and the problem of planning hangouts with social service convinced me to develop such an application to help users to plan events for hangouts in restaurants with auto reminding services and also helps them to share their hangouts activities with social websites.

1.2 Aims

The objective of project is to facilitate users to manage their personal hangout plans .Restaurant Hangout is an idea to build an Android application to help users to find restaurants near them and enable them to plan events for their hangout.

Restaurant Hangout is also able to help users to share their plans with friends on social websites. When the user creates an event and set its day, time and name, it displays the list of restaurants by sorting the nearest one on top of the list. On choosing the restaurant, all the available details of the restaurant including its contact number, deals, review, and location is displayed.

Once the restaurant is selected, the user can login to his/her Facebook account and publish a status on Facebook, Google plus or send SMS. When the plan is checked in, it is moved to the completed plan list. However, if the plan is deleted, it is moved to the incomplete plan list.

1.3 Technologies

The use of Eclipse and Its SDK and Emulator Will be used to build the Application while Implementing the Java Language. The use of the Android SDK will be used Where Android 5 and 6 for lollipop and marshmallow. The Facebook and Goggle Maps Android were used. XLM was needed in many areas and the Jason parser libraries. When my back end database used is SQLite,

1.4 Structure

In the next chapters of the Document we will look at the main requirements for the project step by step and how the Android application will handle its methods and the flow of the application. Then we will go into the non-functional requirements.

Next is the Architecture and the main design followed by how we plan to test the application. Next there is a detailed describe of a graphical user interface with diagrams to give everything a look at how the application itself looks.

We will go through the documents conclusions and Evaluation next. Towards the end of the document there are many supporting documents that were produced along the process while developing the entire application and documentation.

2 System

2.1 Requirements

3 Requirements Specification

All user requirements are verified and user will be able to create plan, search restaurant, and make entry into the calendar in order to make a plan, social connectivity to share plan and check-ins with tagging friends.

User will be able to use application with maximum training of one hour.

And the user can get any run time error until the application becomes more mature. After fixing all runtime errors and reviewing bug or crash report in time or at a later revision it will run better.

3.1 Functional requirements

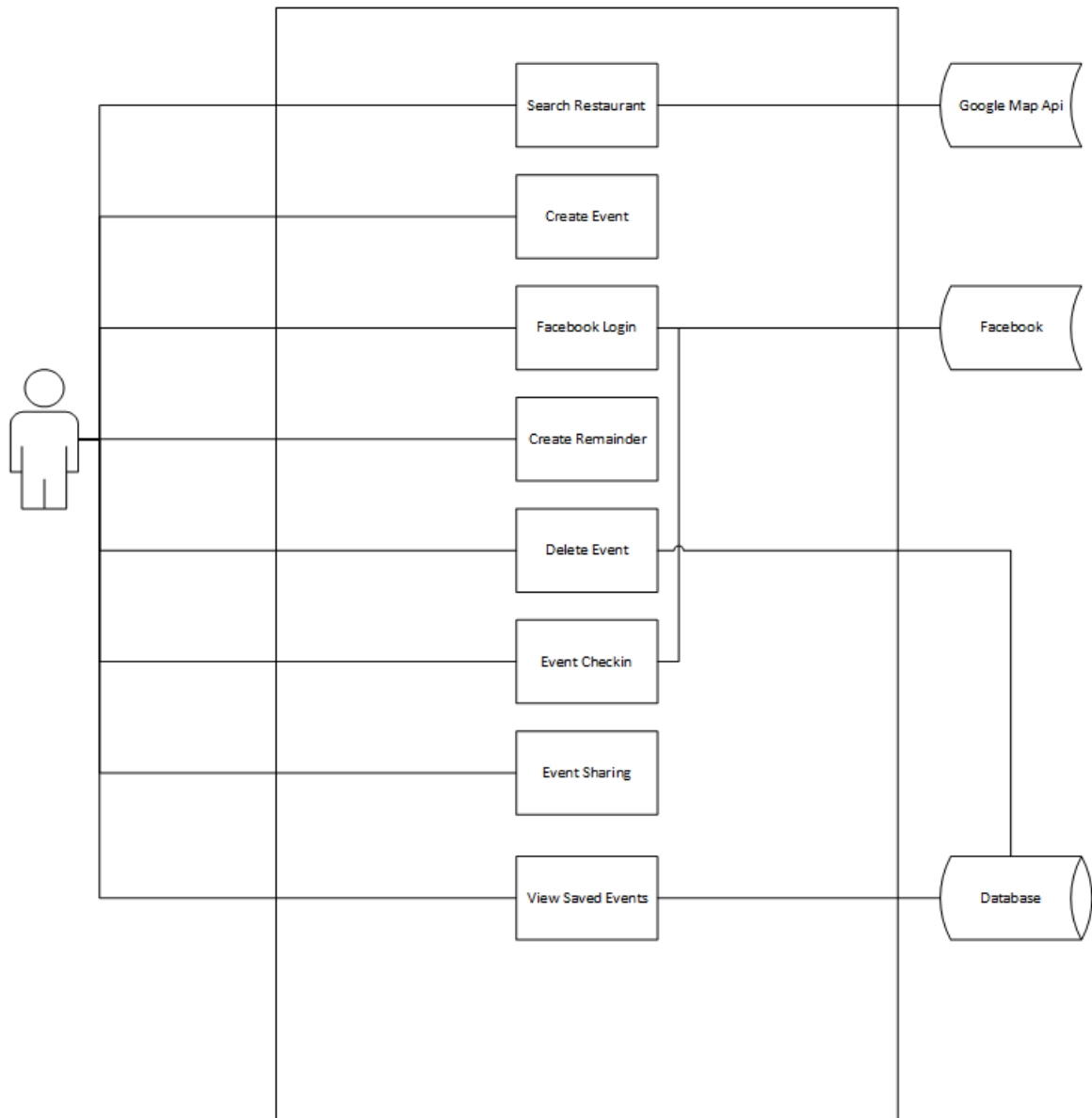
Functional requirements are mentioned below:

1. User should be able to plan a Meeting
2. User should be able to search Location near him/her
3. User should be able to make a calendar entry of the plan
4. User should be able to share plan with friends on social networking websites
5. User should be able to check-in
6. User should be able to tag his friends while post a plan
7. Interface should be user friendly

3.1.1 Use Case Diagram

The Use Case Diagram provides an overview of all functional requirements.

User case diagram is mentioned below:



3.1.2 Requirement 1 User should be able to find restaurant near him/her

Description & Priority

This is the core requirement to find nearby Locations with details by using GPS and Google Maps. User should get the list of all Food, Bars, Cafes or others which are near them based on rating. Priority of this requirement is high.

Use Case

Find Locations near you

Scope

The scope of this use case is to display a list of locations to meet up, food and drink near the user.

Description

This use case describes the flow of sequence of actions for users in order to search Locations near user. Based on location of user that is captured using Google Maps, applications suggests the nearby restaurants. It consists of four steps that are mentioned below:

Flow Description

Precondition

Main Screen of RM is active and GPS is enabled

Activation

This use case starts when user clicks on button “Locations near you” on main screen

Main flow

1. User clicks on button “Locations Near you” on main screen
2. User Waits for GPS to find location
3. User Waits for Google Maps API to get Location
4. application renders List of nearby locations

Alternate flow

No alternative flow.

Exceptional flow

No exceptional flow

Termination

A list of restaurants is displayed

Post condition

User gets list of Locations near him

3.1.3 Requirement 1 Get Information about Location

Description & Priority

User should be able to get all information of a Location .Application should display rating, menus, location, contact number and images. Priority of this task is medium.

Use Case

Get all information of a Location

Scope

The scope of this use case is to display details of a Location that is searched among the list of nearby Locations.

Description

This use case describes the flow of sequence of actions for users in order to get details of a Locations near user. It consists of four steps that are mentioned below:

Flow Description

Precondition

Main Screen of RM is active and GPS is enabled

Activation

This use case starts when user clicks on button “Locations near you” on main screen

Main flow

1. User clicks on button “Locations Near you” on main screen
2. Application find Locations near user

3. Application displays list of searched Locations
4. Click on a restaurant Name

Alternate flow

No alternative flow.

Exceptional flow

- 2 .a) User goes back to previous screen by clicking back button
- 2 .b) Application does not find any nearby Locations
4. A) User goes back to previous screen by clicking back button

Termination

Application will display name, rating (if available), location, phone number (if available), picture or icon

Post condition

Application will display a detailed information of a Locations

3.1.4 Requirement 1 Make a plan

Description & Priority

User should be able to create a plan. Priority of this requirement is high.

Use Case

Make a Plan

Scope

The scope of this use case is to create a user plan

Description

This use case describes the flow of sequence of actions for users in order to create. User can also select Locations while planning for hangout. It consists of eight steps that are mentioned below:

Flow Description

Precondition

Main Screen of RM is active and GPS is enabled

Activation

This use case starts when user clicks on button “Create Hangout plan” on main screen

Main flow

1. User clicks on “Create plan”
2. Application shows screen of “Create a Plan”
3. User enters name of plan
4. User enters date and time of plan
5. User selects Location
6. User selects Save

Alternate flow

No alternative flow.

Exceptional flow

- 3 .a) User goes back to previous screen by clicking back button

Termination

The plan would be save

Post condition

Application will store

3.1.5 Requirement 1 Share a plan after saving

Description & Priority

After saving plan, user should be able to share a plan on Facebook and Google after log in. User should also be able to tag his friends while sharing post. Priority of this requirement is high.

Use Case

Share a plan after saving

Scope

The scope of this use case is to share a created plan by a user on social website and tag friends.

Description

This use case describes the flow of sequence of actions for users in order to share created plan on social. User can also tag his friends while sharing post for hangout. It consists of eight steps that are mentioned below:

Flow Description

Precondition

GPS is enabled, Plan is saved

Activation

This use case starts when user clicks on button "Create plan" on main screen

Main flow

1. User clicks on "share" to share with available options in phone
2. User clicks on icon of social website

Alternate flow

- 2 .a) User clicks “post on Facebook” to post
- 2 .b) User clicks “post on google plus”
- 2 .c) User clicks “sms to make the text ready for you”

Exceptional flow

- 2 .a) User goes back to previous screen by clicking back button

Termination

The plan would be save

Post condition

Application will share the plan

3.1.6 Requirement 1 Delete a plan

Description & Priority

After saving plan, user should have option to delete his plans. Priority of this requirement is high.

Use Case

Delete a Plan

Scope

The scope of this use case is to delete a created plan by a user.

Description

This use case describes the flow of sequence of actions for users in order to delete created plan. It consists of four steps that are mentioned below:

Flow Description

Precondition

GPS is enabled, Plan is saved, and User is on main Screen

Activation

This use case starts when user clicks on button “Manage Plan” on main screen

Main flow

1. User clicks on “Manage Plan”
2. Application opens a new screen “Manage Hangout Plan”
3. User clicks on a button
4. User clicks on “delete”

Alternate flow

- 3 .a) User clicks “active plan”
- 3 .b) User clicks “inactive plan”

Exceptional flow

- 1 .a) User goes back to previous screen by clicking back button

Termination

The plan would be save

Post condition

Application deletes plan

3.1.7 Requirement 1 Share existing plan

Description & Priority

After saving plan, user should have option to share his old plans. Priority of this requirement is high.

Use Case

Share existing plan

Scope

The scope of this use case is to share an old plan by a user.

Description

This use case describes the flow of sequence of actions for users in order to share old plans by user. It consists of eight steps that are mentioned below:

Flow Description

Precondition

GPS is enabled, Plan is saved, and User is on main Screen

Activation

This use case starts when user clicks on button "Manage Plan" on main screen

Main flow

1. User clicks on "Manage Plan"
2. Application opens a new screen "Manage Plan"
3. User clicks on a button
4. User clicks on "share"
5. Same steps as for use case 3.1.5.2

Alternate flow

- 3 .a) User clicks "active plan"
- 3 .b) User clicks "inactive plan"

Exceptional flow

- 1 .a) User goes back to previous screen by clicking back button

Termination

The plan would be save

Post condition

Application shares plan

3.2 Non-Functional Requirements

Non-functional requirements are requirements that specify criteria that can be used to judge the operation of a system, rather than specific behaviors. This should be contrasted with functional requirements that define specific behavior or functions.

3.2.1 Performance/Response time requirement

Application should not halt in case of any action. Response of any action should be less than 3 seconds at maximum and long processing tasks should be done asynchronously.

3.2.2 Availability requirement

The typical objectives of availability requirements are to:

- Ensure that something is available for use when it is needed.
- Maximize the amount of time that it is available for use.
- Minimize both the scheduled and unscheduled down-time during which it is not available for use.
- Enable users to start using it at arbitrary user-chosen times.
- Enable its users to continue using it once they start.

3.2.3 Recover requirement

Application should be able to recover after crash due to any type of run time error which can be achieved through proper exceptional handling.

3.2.4 Reliability requirement

Reliability is the degree to which something operates without failure under given conditions during a given time period. All features as required by users should work without any possible failure.

3.2.5 Maintainability requirement

The typical objectives of a maintainability requirement are to:

- Ensure that minor defects in an application or component are easy to correct.
- Ensure that minor enhancements to an application or component are relatively easy to implement.
- Minimize maintenance costs.
- Minimize maintenance organization staffing needs (e.g., to free up maintenance programmers for new development).

So, Application should be developed by following all programming standards that makes it flexible for future enhancements.

3.2.6 Portability requirement

Application should be able to run in any android device.

3.2.7 Reusability requirement

Reusability Requirement is a developer-oriented quality requirement that specifies a minimum required amount of the quality factor reusability.

4 Interface requirements

UI is an important part of any software or hardware or hybrid system. A software is widely accepted if it is -

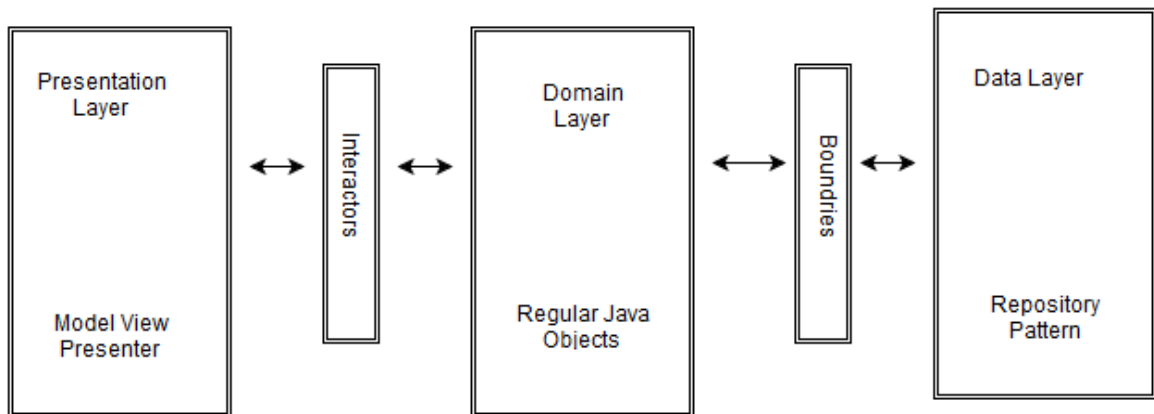
- easy to operate
- quick in response
- effectively handling operational errors
- providing simple yet consistent user interface

User acceptance majorly depends upon how user can use the software. UI is the only way for users to perceive the system. A well performing software system must also be equipped with attractive, clear, consistent and responsive user interface. Otherwise the

functionalities of software system cannot be used in convenient way. A system is said be good if it provides means to use it efficiently.

4.1 Design and Architecture

System Architecture is divided into 3 different layers in order to achieve separation of concerns by keeping the business rules not knowing anything at all about the outside world, thus, they can be tested without any dependency to any external element. Each Layer one has its own purpose and works separately from the others. It is worth mentioning that each layer uses its own data model so this independence can be reached .Here is an schema so you can see how it looks like:



Presentation Layer

In this layer, the logic related with views and animations happens. Presenters in this layer are composed with interactors (use cases) that perform the job in a new thread outside the android UI thread, and come back using a callback with the data that will be rendered in the view.

Domain Layer

Business rules here: all the logic happens in this layer. Regarding the android project, you will see all the interactors (use cases) implementations here as well.

This layer is a pure java module without any android dependencies. All the external components use interfaces when connecting to the business objects.


Data Layer

All data needed for the application comes from this layer through a User Repository implementation (the interface is in the domain layer) that uses a Repository Pattern with a strategy that, through a factory, picks different data sources depending on certain conditions.

The idea behind all this is that the data origin is transparent for the client, which does not care if the data is coming from memory, disk or the cloud, the only truth is that the data will arrive and will be got.


4.2 Implementation

The manifest here uses the android SDK in connection to the application.

 **Android Manifest**

▼ **Manifest General Attributes**
Defines general information about the AndroidManifest.xml

Package	<input type="text" value="com.resturanhangout"/>	<input type="button" value="Browse..."/>
Version code	<input type="text" value="1"/>	
Version name	<input type="text" value="1.0"/>	<input type="button" value="Browse..."/>
Shared user id	<input type="text"/>	<input type="button" value="Browse..."/>
Shared user label	<input type="text"/>	<input type="button" value="Browse..."/>
Install location	<input type="text"/>	<input type="button" value="v"/>

Manifest Extras 

Here is the Main activity XML Document that sets of the Relative Layout the application to be started, we set our main Drawable here that may be unique to the rest if required. This XML document also links to the android SDK tools and links to the GUI.

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/rest"
    android:gravity="bottom"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context=".MainActivity" >

</RelativeLayout>
```

4.3 Testing

My own Android device was used in the testing process of this Application. When using an Emulator it is sometimes good to test the main graphical user interface and Test each extends to the next through its interface. The Emulator is limited so testing right onto my

own android device was best with regards to testing one of the map features the google APIs and mapping systems with location and GPS tracking, so definitely the use of my Android was mostly used in Testing.

4.4 Graphical User Interface (GUI) Layout

App Name is the Main screen after loading the application itself. Once the application is

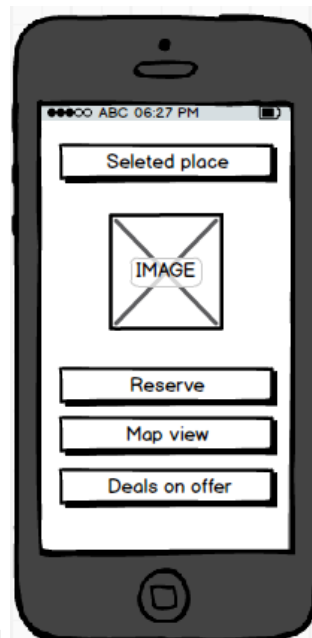


opened the user is presented with options from the first screen

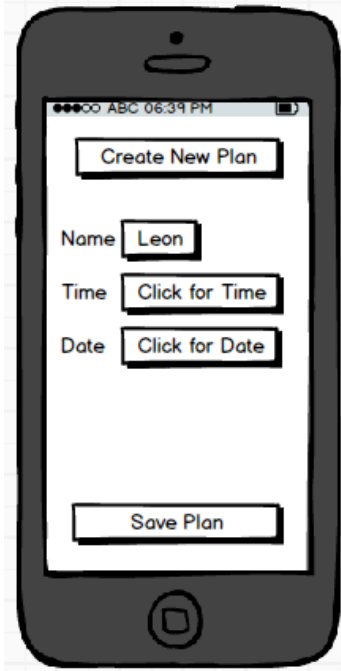
Here the user can locate places of interest close by, create and manage plans. When the



user clicks on “Places close by” The following page is displayed the user can see all available locations of interest on offer. On clicking on any item the



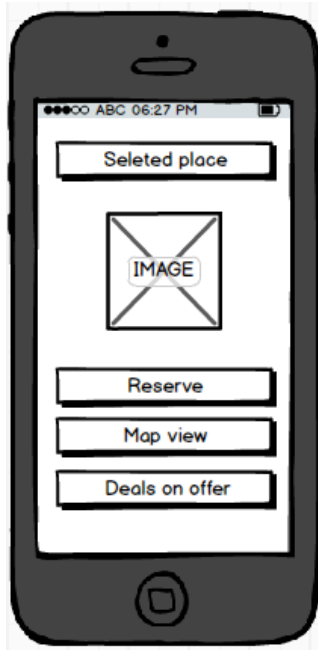
location chosen will be displayed. From here the user can check out the selected location or reserve. If reserve is chosen the user will be brought to the “create plan”. The map view will open the goggle API link to show google maps while your location tracking device is switched on your device. Deals will be possible deals on offer. When the user clicks on “create plan” this next screen will be show



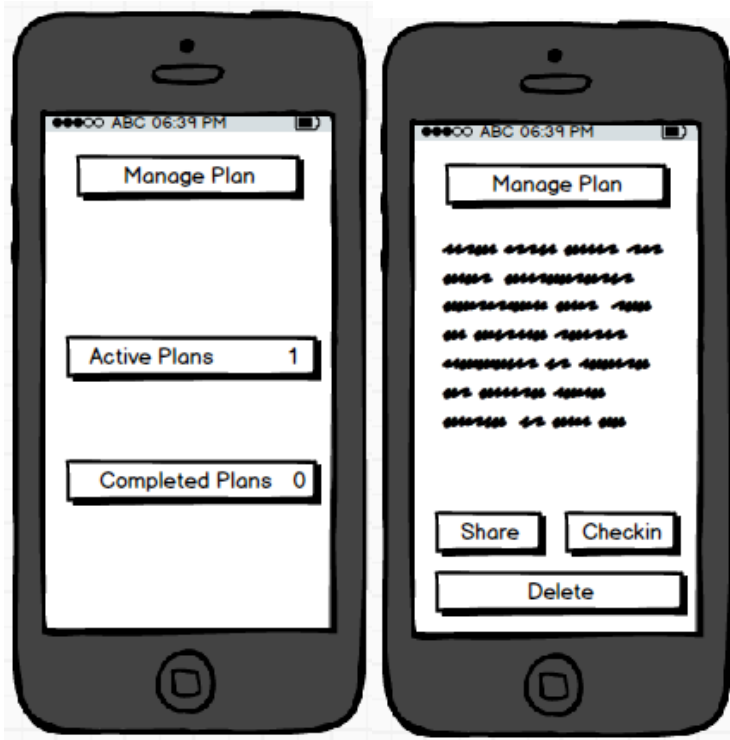
A plan is made, the calendar event on your computer is used from inside the application to set the time and dates and the plan is then saved. And at this point the user will be directed to.



From here the place of interest is selected and you are brought to



Once you reserve the selected, then you are brought to Facebook so as to post and tag friends that will meet up at the location. After selecting which friends are to meet you. You are done and another page will have a page where finished will be presented, and on pressing this button you will be show the plan in which you will be saving. You will save the plan and be redirected to your calendar event. The next screen will show “post to Facebook” and “google+” were your friends will be informed of the plan on offer were everyone will meet. Back at the main screen you can view “manage plan” were you can see available plans ,share with any sharing tool, check yourself in or delete them as show here.



Upon completion you can just press back, back, back to go to the main screen and from here you can look at the application in order of your own preference.

4.5 Customer testing

Unit testing was carried out amount friends – each consumer downloaded the APK file and individual tested it on their own time. Some feedback concerns were reported back and minor adjustments were done on these issues. Most bugs were corrected and further improvements are been made to adequately sort out and bug related issues of concern that came up.

The feedback that was reported back came back as positive, while some customers who tested the Application were happy with what they saw and would be happen to see the improvements updated. Some customers were not happy to share their events always on Facebook, but were delighted when they saw the options of sharing via SMS and Google Plus.

5 Conclusions

The Main Advantages with this project were that I was building a Native Application that will only be used for Android based phones. In doing this I could focus entirely on this rather than in some other applications where you are required to look at many different ways of implementing Applications such as responsive websites that are both for Mobile/Pad or Computer Based were more and more technologies would be required. Doing my Application my way insured that I could get more of an understanding of building an Android application rather than overly confusing myself with too many platforms.

Some of the disadvantages is that the application is based on android 5 and 6

And while this does bring the application to a wide range of devices it does on the other hand limit the applications audience.

The Application itself will have limits, but at the moment there is much room for improvement within the application and over time with some update patches to this Android Application I am positive it could be greatly improved and will evolve over time.

6 Further development or research

With the passage of time, quality of application is improved by fixing bugs and with consistent increase in user experience in order to generate revenue. I will re-launch version of application after a particular time by considering user feedback and bugs report.

In Restaurant Hangout's application, following improvements can be done in future:

- More social account i.e. integrate Twitter and Instagram.
- Improve GUI's by considering user experience
- Integrate web portal for restaurant managers for managing their menus and offers and that will also help them for publicity

References

- [1] Android documents – Product Information. Available from: <http://developer.android.com>
- [2] Developers, Android. "What is android." *http://developer.Android.com/guide/basics/what-is-android.html* 2 (2011).
- [3] <https://developers.facebook.com/docs/android>
- [4] Wasserman, Tony. "Software engineering issues for mobile application development." *FoSER 2010* (2010).
- [5] Cohen, David, Mikael Lindvall, and Patricia Costa. "An introduction to agile methods." *Advances in computers* 62 (2004): 1-66.
- [6] Beck, Kent, Mike Beedle, Arie van Bennekum, Alistair Cockburn, Ward Cunningham, Martin, James Grenning et al. "The agile manifesto." (2001): 2009.

Appendix

6.1 Project Proposal



Proposal Plan
Doc.docx

6.2 Project Analysis and Requirement Specifications Designs



Project
Analysis_Design.docx



ProjectRequirement
Specification (1).do

6.3 Monthly Journals



Reflective Journal
September.docx



Reflective Journal
october.docx



Reflective Journal
november.docx



Reflective Journal
December.docx



Reflective Journal
January.docx