Table of Contents

[Chapter one: Introduction 3](#_Toc444469127)

[1.1 Project overview 3](#_Toc444469128)

[1.2 Project Scope 4](#_Toc444469129)

[1.3 Problem Statement 4](#_Toc444469130)

[1.4 Target users 6](#_Toc444469131)

[1.5 Project objectives 6](#_Toc444469132)

[Chapter two: Analysis and Designs 7](#_Toc444469133)

[2.1 Features matrix 7](#_Toc444469134)

[2.2 System architecture 8](#_Toc444469135)

[2.2.1 Introduction: 8](#_Toc444469136)

[2.2.2 System Architecture Breakdown: 8](#_Toc444469137)

[2.3 Planning and control 10](#_Toc444469138)

[2.3.1 Methodology used and explanation 10](#_Toc444469139)

[2.3.2 Reason for choosing methodology 10](#_Toc444469140)

[2.4 Functional requirements 11](#_Toc444469141)

[2.4.1 Application’s features 11](#_Toc444469142)

[2.4.2 User stories narrative 12](#_Toc444469143)

[12](#_Toc444469144)

[2.5 Planning 16](#_Toc444469145)

[2.5.1Time plans of sprints 16](#_Toc444469146)

[2.4 Database design 19](#_Toc444469147)

[2.6 Scenarios 19](#_Toc444469148)

[2.6.1 Introduction 19](#_Toc444469149)

[2.6.2 User scenarios 20](#_Toc444469150)

[2.7 Use cases 21](#_Toc444469151)

[2.8 Sequence diagrams 24](#_Toc444469152)

[Chapter Three: Implementation 26](#_Toc444469153)

[3.1 Introduction 26](#_Toc444469154)

[3.2 Code in brief 26](#_Toc444469155)

[Chapter Four: Testing 26](#_Toc444469156)

[4.1 Introduction 26](#_Toc444469157)

[4.2 Test cases 26](#_Toc444469158)

[Chapter Five: Evaluation 26](#_Toc444469159)

[5.1 Introduction 26](#_Toc444469160)

[5.2 Features implemented 26](#_Toc444469161)

[5.3 Actual plan VS Estimated plan 27](#_Toc444469162)

[5.4 Challenges of the project 27](#_Toc444469163)

[5.5 What we have learned 27](#_Toc444469164)

[5.6 How to get Around the block application 27](#_Toc444469165)

# Chapter one: Introduction

## 1.1 Project overview

Around the Block a mobile phone application that focuses on filling a need rather than solving a problem for its clients. It basically provides them with all needed information about services and venues geographically around them, hence the name.

Many applications out there provide that exact service to its clients; however, what Around the Block would be introducing is data about small services. Nonetheless it will be equipped with data of all huge known venues as well; as for completion purposes.

Now these small services vary from dry cleaning shops, ironing shops, and fabric shops to repair shops, supermarkets, and the corner restaurants that only locals know about.

By adding such edge to the mix we would have the ability to not only target people who are looking for fancy places to go, or those who can afford to go to known supermarkets that are pinpointed on normal maps, but also give the opportunity to the average person to find places that’s in his/her budget.

The type of data that is given & maintained by the application has a broad range from Address, Tel Number, and images to app rating, user rating, and venue budget. Furthermore, the app will grant users a lot of features based on these data that cannot be found in similar apps. For example we will give the user to ability to use their location to navigate to the address using integrated Google maps.

Around the Block is designed to be as user friendly as possible. E.g. Giving users a very low entry level requirements to sign-up(App installation). By doing so, it has the ability to customize its features to each user preferences. By gathering data on the type of places the user visits and their price benchmarks; it can easily identify & recommend other similar venues the user may be interested in visiting.

In a nutshell, we are aiming to construct the application “Around the Block” in a way that makes it as appealing and easy to use as possible; establishing its ranking in the market to be the Facebook of service locating applications.

## 1.2 Project Scope

Our project scope is finishing the application due July with the user able to sign-up search for a specific location that he/she desires navigate to this location, see and do the ratings and reviews of the specified location.

## 1.3 Problem Statement

There are loads of problems & needs that this application is offering to fix & fill. The idea of the application arose from the fact that loads of people including ourselves though provided maps still struggle & spend a lot of time figuring out how to reach the closest mechanic, supermarket or even a dry cleaner. We still use the same old pre-technological method of asking people around to guide us to our destination. Now, one obvious issue with this is the enormous expenditure of time in doing so. Nonetheless, the possibilities of getting lost oblivious where we are heading.

Furthermore, in many occurrences we try to find something new to do with friends or a new place to explore often only to find ourselves faced with limited options on the virtual world. Hence, choose the old way of calling people or even cruising around just to find this corner café or that bowling place that only locals know about. Yes, this is another issue. Locals know how to enjoy their time whenever. On the other hand, tourists or people whom are new to living in some city find themselves obliged to go to the known places in tourists guides. Which we all know is not where inhabitants of this city go to enjoy themselves. Not to mention how tourists also suffer from the first problem of not finding basic shops to carry on their daily operations.

Apart from the fact that other service locating applications and guides don’t provide small services information; often times their data is outdated and no longer useful. The venues might be shut down or under maintenance.

One last complication is bound to arise as soon as we figure out solutions for the above. It is that because there is actually abundance in the amount of places we can go to, and that it’s just not mapped for people to know about; the user will have to figure out a way to choose between these places. Moreover, just knowing that there is that restaurant around the corner doesn’t mean we know that it’s cheap, tasty, or even clean. How can users know if that mechanic two blocks away is trust worthy to leave the car with him for a day or two?

For all of these dilemmas we decided to build an app which can be the ultimate navigation guide that would handle these issues once and for all.

## 1.4 Target users

Our main target users are the teenagers and the youth who use their smartphones daily and are always updated with new technologies and applications, our secondary users are the people who own the services and might be interested to advertise their service and market for it on their own profile (out of the scope ).

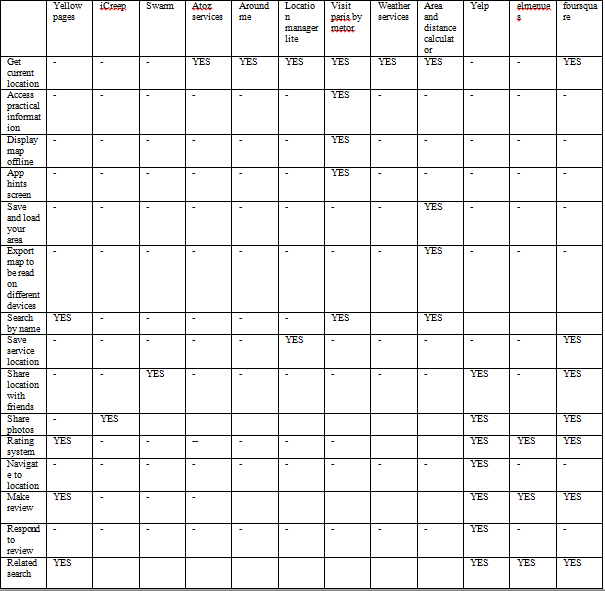
## 1.5 Project objectives

Around the block objectives are quite simple, we just want to make it easier for everyone. Our main objectives are:- saving the enormous time consumption for the user searching for some service that might be two blocks away, fill in the need for trying to figure out where is the nearest new place that might be my type of place to hang out in.

# 

# Chapter two: Analysis and Designs

## 2.1 Features matrix



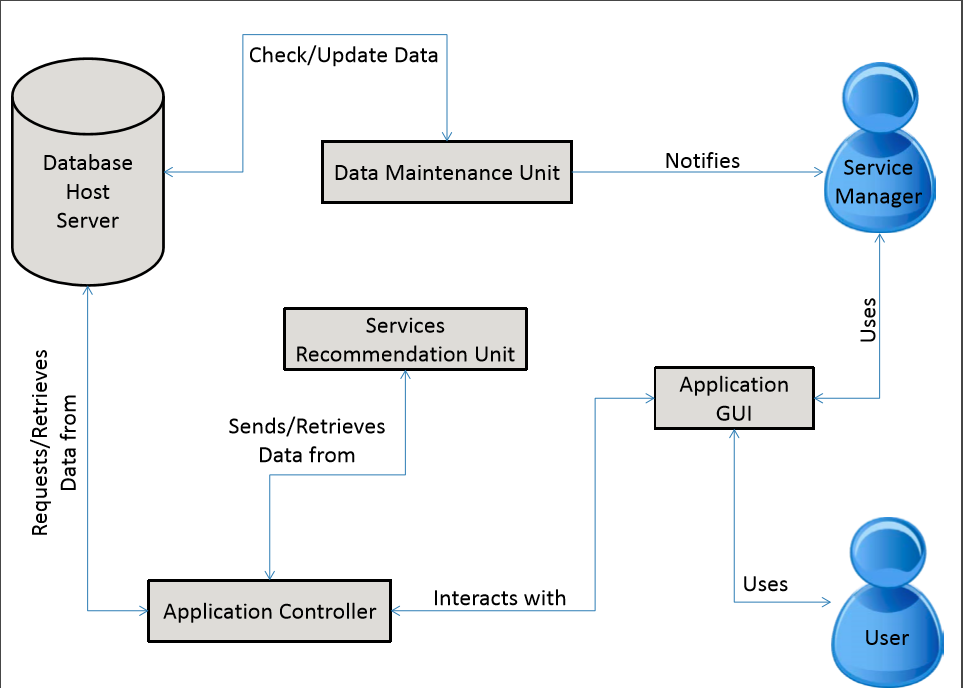
## 2.2 System architecture

### 2.2.1 Introduction:

To give an overview of how each part of the system interacts with each other. It’s also meant to act as a guide for the implementation of such system.

### 2.2.2 System Architecture Breakdown:

1. **Database Host Server:** This part of the system is responsible for holding all the data of services info, users info, and service managers info.
2. **Application GUI:** This is where the user actually interacts with the system. S/He is allowed to search and browse through different services, find out information about them. S/He also has the ability to review, and rate them.
3. **Application Controller:** Now this part basically connects to the database server; allowing requests and retrieval of data from it. It also processes the input coming from the user through the GUI, and shows the results back through the GUI.
4. **Services Recommendation Unit:** Gathers data about the user through the Application Controller, process it, and gives back services recommendations to what this user may find interesting.
5. **Data Maintenance Unit:** Makes sure that services registered on the system are up to date. It essentially prompts/notifies service managers to update their service status. And for the most part handle exceptions to those who don’t interact.

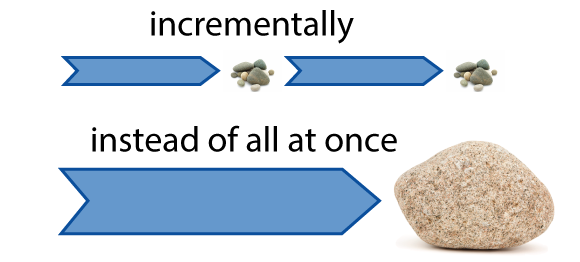


## 2.3 Planning and control

### 2.3.1 Methodology used and explanation

We used the Agile methodology.

Agile is a time boxed, iterative approach to software delivery that builds software incrementally from the start of the project, instead of trying to deliver it all at once near the end. It is an alternative to traditional project management, typically used in software development. It helps teams respond to unpredictability through incremental, iterative work cadences, known as sprints. Agile methodologies are an alternative to waterfall, or traditional sequential development.



### 2.3.2 Reason for choosing methodology

It works by breaking projects down into little bits of user functionality called [user stories](http://www.agilenutshell.com/user_stories), prioritizing them, and then continuously delivering them in short two week cycles called  [iterations](http://www.agilenutshell.com/iterations), Thus this gives the chance to fully understand the requirements and make sure that they meet the user’s needs.

## 2.4 Functional requirements

### 2.4.1 Application’s features

#### 2.4.1.1 Main features

The main features are Signing-up, searching, writing reviews, and ranking a service.

#### 2.4.1.2 Future features

There are nice to have features which we can add in the future like: get current location, show location on map, navigate to requested location and save location on map.

### 2.4.2 User stories narrative

**User Story for Sign Up:**

**Story Narrative**

**Title: Sign Up**

**As** A User of Around The Block

**I want to** Sign up and gain access to the application with the

least amount of personal information taken from me

**So that** I can make sure that my private information are not used by anyone

## 

**Acceptance criteria**

**Title: Sign Up**

**Given** That the user have no account on Around The Block

**When** The user opens the application and click on Sign up button

**Then** The user must give the required information and click on submit, then an account will be created for this user

**User Story for Sign-in with facebook:**

**Story Narrative**

**Title: Sign in with facebook**

**As** A User of Around The Block

**I want to** Sign in with facebook so I can share places with my

Friends.

**So that** I can make sure that my private information are not used

by anyone

**Acceptance criteria**

**Title: Sign in with facebook**

**Given** That the user have no account on Around The Block

**When** The user opens the application and click on the

Sign in with facebook button

**Then** The user is redirected to homepage and his account is created.

**User Story for Search:**

**Story Narrative**

**Title: Search**

**As** A User of Around The Block

**I want to** get involved with the application to be able to search the nearby places or to search by category

**So that** I can reach any place easily

**Acceptance criteria**

**Title: Search**

**Given** That the user wants to search for a specific place

**When** The user opens the application and chooses to search

by nearby places or by category

**Then** The user must give the required information and click

on submit, then all the required places will appear

**User Story for rating:**

**Story Narrative**

**Title: Rating**

**As** A User of Around The Block

**I want to** Rate the places I have visited before.

**So that** I can make it easy to other user to know if this place is good Or not.

**Acceptance criteria**

**Title: Rating**

**Given** That the user signed in to the application.

**When** The user searches for a specified location

**Then** The user can give rating to this location,

And by submitting, the ratings will be saved in the

database

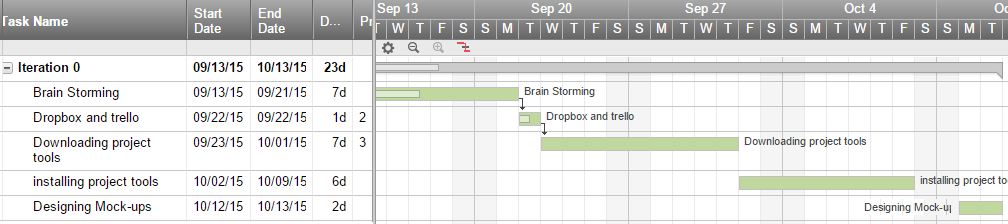
## 2.5 Planning

### Time plans of sprints

**Sprint zero:**

Deliverables of sprint 0:

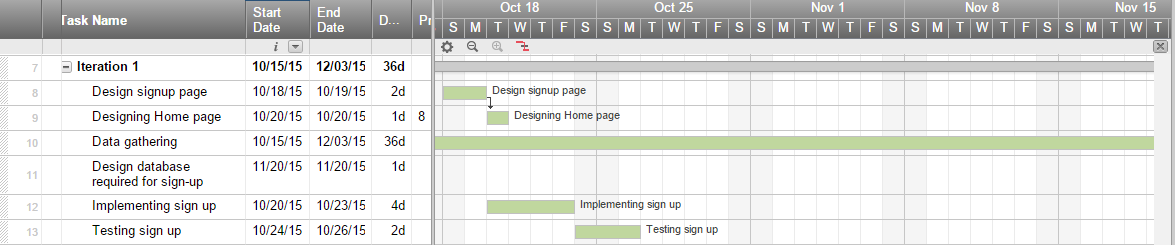
1. Idea of project
2. Tools ready



**Sprint one:**

Deliverables of sprint 1:

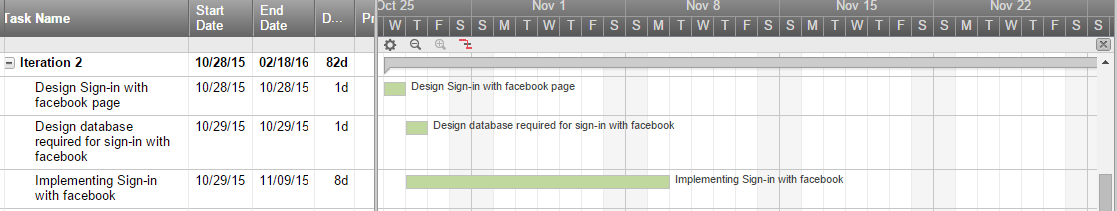
1. Sign-up
2. Home page



#### Sprint two:

Deliverables of Sprint 2 (sign-in using facebook):

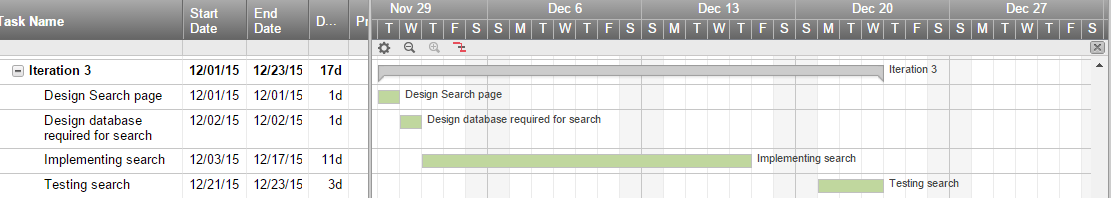
1. Not delivered



**Sprint three*:***

Deliverables of Sprint 3 (Search):

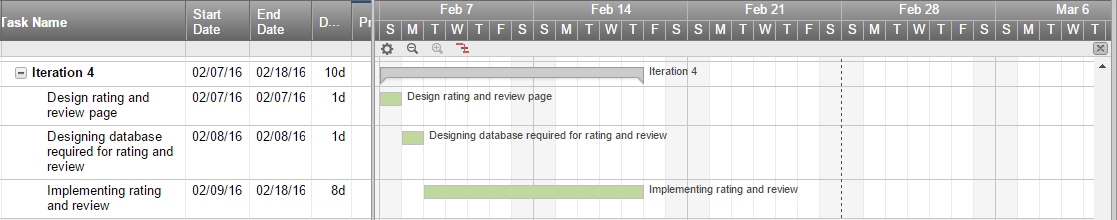
1. Search



#### Sprint four:

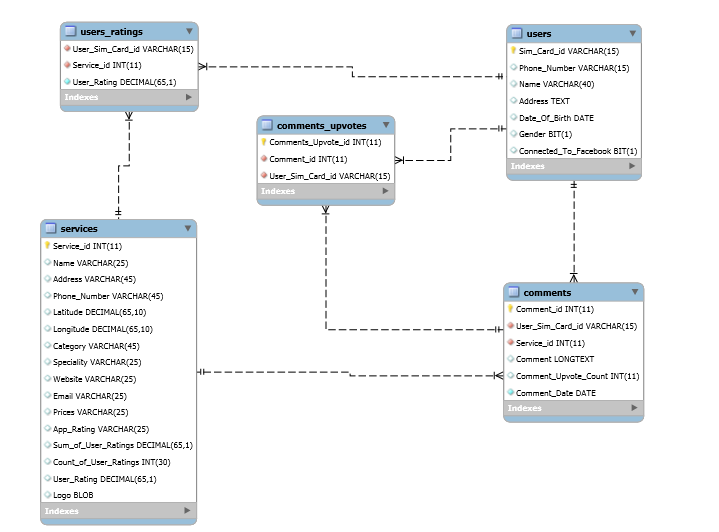
Deliverables of Sprint 4 (Rating and Review):

1. Ongoing iteration



## Database design

**ERD:**



## Scenarios

### 2.6.1 Introduction

Scenarios describe the stories and context behind why a specific user or user group comes to your site/application. They note the goals and questions to be achieved and sometimes define the possibilities of how the user(s) can achieve them on the site.

Scenarios are critical both for designing an interface and for usability testing.

**There are three types of scenarios:**

1- Goal- or Task-Based Scenarios.

2- Elaborated Scenarios.

3- Full Scale Task Scenarios.

We chose the elaborated scenarios to describe our systems.

### 2.6.2 User scenarios

**1- Sign up function:**

Sara, who is half American half Egyptian and has lived most of her life away in Miami with her parents. Her parents made a decision to settle down in Egypt and live in Cairo were sara would continue her education in Cairo university with her cousins but of course she doesn't have a clue on how to move around in such a big city so her aunt told her about a good application for her to use that would help here from getting lost, she successfully downloaded it and now has to sign up. Also, she is reluctant to give out personal information, so she want to know how safe it is… she opens the application, enters her name with little amount of information she signs up to the application and can use it anytime she wants without asking her to sign in again..

1. **Sign in with Facebook:**

Sara love hanging out with her friends in different places, she wants to share these moments with them and recommend these places to her other friends, so she needs the application to be linked with her social accounts such as Facebook, Twitter etc. To help her friends make use of the application.

1. **Search scenario:**

Sara stood stranded in the middle of El-Zamalek lost. She was new to this area and needed to withdraw money from her bank account but of course she didn’t know where the closest bank is so she used the application her aunt had told her about. She had two options either to search by category and choose the closest one to her, or to search any place.

* 1. **Rating Scenario:**

Sara visited lots of different places and used many services provided by the application so far. She wants to give her opinion about the services she used, so she used the rating feature in the application in order to help her friends and other people as well to use the best services.

* 1. Use cases

1-Sign up:

|  |  |
| --- | --- |
| **User case ID:** | A-001. |
| **User case name:** | Sign up event. |
| **Actors:** | User. |
| **Pre-condition:** | User has internet connection. |
| **Post condition:** | User is signed-in and redirected to home page. |
| **Flow of events:** | |  |  | | --- | --- | | User action | System action | | User opens the application. | 1)System collects sim card ID. | |  | 2) System checks if sim card ID is in the database | |  | 3)If not found system prompts user to enter his name | | 4) User enter his name and submits |  | |  | 5) System adds user sim card ID and name to database | |  | 6) System redirects user to home page | |
| **Exceptions:** | 1- If user submits an empty name; system gives error message & prompts the user to submit his/her name  2- If system found Sim Card ID then User is signed in and redirected to Home Page |

2- Sign in with facebook:

|  |  |
| --- | --- |
| **User case ID:** | A-002. |
| **User case name:** | Sign in event. |
| **Actors:** | User. |
| **Pre-condition:** | User has internet connection  User has facebook account |
| **Post-condition:** | User redirected to homepage. |
| **Flow of events:** | |  |  | | --- | --- | | User action | System action | | 1)User opens the application, press sign in with facebook | 1)System request connection with facebook | | 2) User accepts system’s request |  | |  | 3)System collects user information from facebook and add it with sim card ID to database | |
| **Exceptions:** | 1. If the user cancelled system request for connecting with facebook |

3- Search:

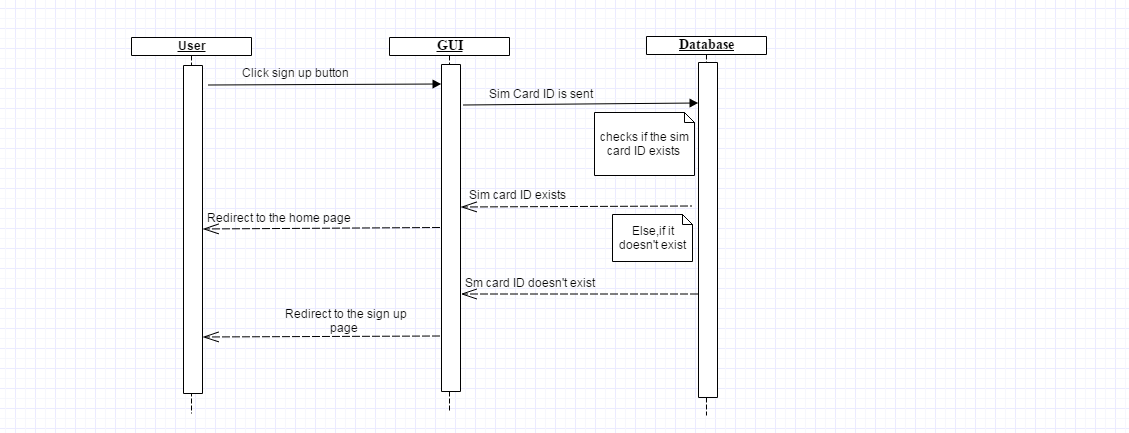
|  |  |
| --- | --- |
| **User case ID:** | A-003. |
| **User case name:** | Search event. |
| **Actors:** | User. |
| **Pre-condition:** | User has internet connection  User is signed-up |
| **Post-condition:** | None |
| **Flow of events:** | |  |  | | --- | --- | | User action | System action | | 1)Search places by category. | 1)System get all the info about the specified place from database | |  |  | |
| **Exceptions:** | 1. If the user searched for a place not in the database |

4-ratings:

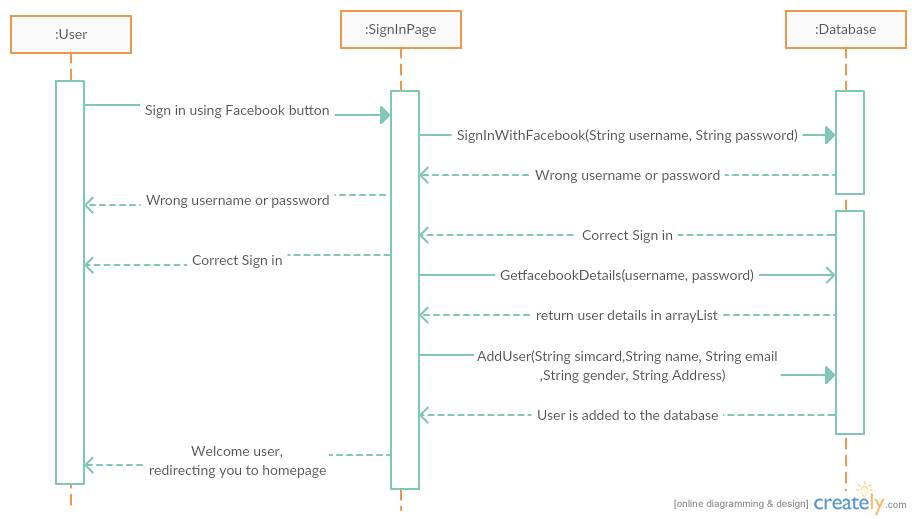
|  |  |
| --- | --- |
| **User case ID:** | A-004. |
| **User case name:** | ratings event. |
| **Actors:** | User. |
| **Pre-condition:** | User has internet connection  User is signed-up |
| **Post-condition:** | User can rate any place. |
| **Flow of events:** | |  |  | | --- | --- | | User action | System action | | User rates the place he has visited. | System saves user ratings to the database. | |  |  | |
| **Exceptions:** | None. |

## **2.8 Sequence diagrams**

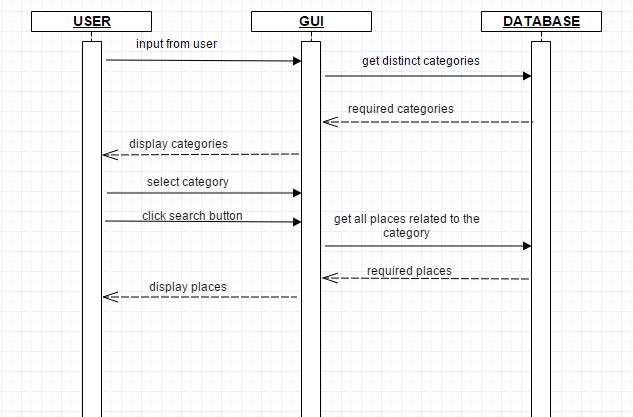
1. Sign-up



1. Sign-in with facebook :



1. Search



# Chapter Three: Implementation

## 3.1 Introduction

## 3.2 Code in brief

# Chapter Four: Testing

## 4.1 Introduction

## 4.2 Test cases

# Chapter Five: Evaluation

## 5.1 Introduction

## 5.2 Features implemented

Sign-up

Search

Rating

## 5.3 Actual plan VS Estimated plan

## 5.4 Challenges of the project

So far, we spent too much time installing the softwares needed to start the implementation, we had many failures to the windows and storage issues.

We also had very hard time collecting data from the streets as it was time and energy consuming.

## 5.5 What we have learned

So far, we have learned how to work in a group and reach agreement with all team members, use the agile methodology correctly and how to develop android application.

## 5.6 How to get Around the block application

We will put it on google play store as soon as we finish the main features and test it.