****

**CIE203**

**Software Engineering**

Project Location-based Social Network

Software Requirements Specifications

2M&K

Khaled Hefnawy

November / 2016

Contents

[Team 2](#_Toc465779096)

[Document Purpose and Audience 3](#_Toc465779097)

[Introduction 3](#_Toc465779098)

[Software Purpose 3](#_Toc465779099)

[Software Scope 3](#_Toc465779100)

[Definitions, acronyms, and abbreviations 3](#_Toc465779101)

[Requirements 3](#_Toc465779102)

[Functional Requirements 3](#_Toc465779103)

[Non Functional Requirements 4](#_Toc465779104)

[System Models 4](#_Toc465779105)

[Use Case Model 4](#_Toc465779106)

[Use Case Tables 4](#_Toc465779107)

[Ownership Report 5](#_Toc465779108)

[Policy Regarding Plagiarism: 6](#_Toc465779109)

# Team

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Email** | **Mobile** |
| 201300812 | Khaled hefnawy | [s-Khaled.mohamed@zewailcity.edu.eg](mailto:s-Khaled.mohamed@zewailcity.edu.eg) | 01066394056 |
| 201301691 | Mohamed Wasfy | [s-mohamed.safieddin@zewailcity.edu.eg](mailto:s-mohamed.safieddin@zewailcity.edu.eg) | 01110594119 |
| 201305056 | Mohamed Alaa el-din Fargaly | [s-mohamed.fargaly@zewailcity.edu.eg](mailto:s-mohamed.fargaly@zewailcity.edu.eg) | 01002680520 |
| 201306057 | Hazem Hamdy | [S-hazem.alromisy@zewailcity.edu.eg](mailto:S-hazem.alromisy@zewailcity.edu.eg) | 01288449661 |

# Document Purpose and Audience

* This document represents the contract between the client and the developer. It gives all things needed to develop a fully functional software program. It states the functional and nonfunctional requirements that are to be taken into consideration while developing the app to satisfy the client.it explains many use cases that covers almost all possible scenarios of interactions of the user. It covers the purpose and scope of the program. It also gives all the needed definitions needed to be known be the developer.

# Introduction

## Software Purpose

The purpose of this software is to get people together literally. As ordinary social media is somehow accused of separate people and not giving them enough face to face meetings, so this location based is to notify people when friends are near them, besides enhancing common taste of places between friends.

## Software Scope

* The main scope of the system is locating people through GPS, commenting on the places giving tips commenting on them, liking others’ posts who checked in a place, locating friends in the same place, giving a taste for each place, premium users defining new places in the map, having the ability to send messages and chat with others and having the ability to see notifications when brands make popular events and finally seeing friend’s notification of checking in a place.

## Definitions, acronyms, and abbreviations

* Brand: a commercial page that market a certain item and give tips for followers.
* Check-in: to record the place that the user has been to in a certain time.
* Caption: description on the check-in event.

Notifications: message to notify the user if an important event has just happened such as a friend request or chat request in form of a pop up message or just a small message in the notification bar

Tastes: the theme of the place like class of the museum or type of food for restaurant.

Normal user: the identity of a user that register with free account and have normal access to free services.

Premium user: the user that pay for the account have the same basic access of a free user in addition to special access to services like registering new place.

GPS: Global positioning system is a standard system to detect the location using the satellites signals, the device is found almost in all new mobiles phone.

Registration: the act of creating new account and new user using vailed username, email and password.

Log-in: the act of using username and password to access the service.

Create Place: To add new entity to a location on the map giving the place description and class.

Rate: evaluate a place on scale of stars depending on its quality.

Review place: to give your opinion and evaluation on the place that you had visited helping other people to get impression of the place.

Friend Request: it is the request which a user send to another user to add him to his contact list and be able to follow each other.

Follow Friend: to have another user in your list and get access to his sharing content while giving him access to yours.

Friends List: the list that define each user contacts

Like and Comment: like is the act of showing users’ preferences to other users while comment is the act of writing user opinion on other user’s content, both acts access depend on the sharing level of the content.

Brand Tip: it is a special advice usually short that give the best possible faces of some place in brief words.

Chat & group Chat: chat is mutual message system between two users after sending a request from one side and accept it from the other if number of people get more than two it is called a group chat.

# Requirements

## Functional Requirements

* Downloading the system: for mobility use it will be dedicated as an application so it will be downloadable from different stores (google play, apple store, and Microsoft store …. etc.). (1)
* Running the app: by choosing it from the list of apps depending on the platform and clicking on it if the app is already running it should open on the last updated state. (1)
* Closing the app: it should have two option of closing first to totally close it and all its running components, which means it will be disconnected or to close it, but letting it keep updating places and GPS status and notifications update. (2)
* The system should have login and logout function: the system should allow the user to login by entering their user name and password and click “login” button, this should appear only when the user is not logged-in. if the user is already logged in, the “log out” button should appear instead. (2)
* Retrieving password: by tapping “forgot password” button the user will be asked to enter the email they registered with (or phone number), a new temporary password will be sent to him and after entering it the system gives him ability to change password immediately (not full login without entering the new password) (2)
* Differentiate between normal user and premium user: the user type introduces different amount of available services for each type, so the system should be able to get user type from the data base once the user logged in and give them their available services. (1)
* Saving all the user activity: any activity the user does, the system should keep record of it and the user to be able to visit it back from “activity log” menu under this menu the activities should be able to be filtered according their type (check-in, comment, ratting …… etc.) (3)
* Access to GPS: the location for this app is important so the app automatically should keep following the changes in location and update the sate according to it, it should give the user a map to be able to get directions from the app. (3)
* Search places: the user will enter the place name in the search bar, the system finds and lists all places by the closest matches and gives the user “go to” option. (3)
* Getting directions: the user can get directions by entering the wanted place or coordinates to the map and the GPS access will give them directions to follow. (3)
* Detect movement and recognize the most famous places: while the user is moving the app should automatically give them marks of the famous place on the map and by tapping on it, it gives the description and previous rating of the place. The map displays the places in 1 km radius circle around the current location. (4)
* check in places: the user according to their type (normal or premium) are able to check-in by just tapping “check-in” button above the map, after that a new page appears to write caption and ratting for the place. (4)
* allow premium user to create new place to check in and this places will be open only for premium users, for premium users only the system allows them to create new places and naming them while other users are referred to the closest famous landmark. (2)
* add caption to his check-in: after tapping “check in “button new page appears to write the caption and when the user is done, they tap “post” button to record the check-in and the caption and starts new operation of updating data base and notify friends. (5)
* User should be able to follow other users and add them in friends list: the user opens the candidate friend page and tap “add to friends list” and the button changes to cancel if the user wants to cancel. (3)
* Accept or refuse the friend request: the added user will be notified to accept or refuse friend request according to choice the data base and friends list will be updated. (1)
* Like or comment on their check-in post: by tapping “like or comment” buttons the user can like their friend’s activity. (2)
* User should be able to access his list of places and friends: by tapping menu “friends” and “places” buttons will appear and by tapping them, app will open other menu with requested content (2)
* Evaluate, add opinion and add a taste for the place like (Italian food for restaurants and classic for a museum): user can rate a place by navigating STARS Bar and rate it from 1 STAR to 5, also can add opinion to appear in place page. (2)
* Brand marketing: brands will be only available to be followed by premium users. Brands should be able to add tips or piece of advice to its follower. (2)
* Messaging between users. The channel is always open between friends and needs approval if the two parties are not friends. It should also allow group chat. By tapping messages older messages will appear and other icon to open new message named of the user (or users) will be asked when “new message is chosen. (5)
* Notifications: when something important happen such as (a friend is nearby, a new tip from a followed brand, a new message, another friend is checking in the same place), the user is notified automatically as long as the app is running. (5)

## Non Functional Requirements

|  |  |
| --- | --- |
|  | Details |
| Usability: | * The system should have a friendly user interface that is self-explanatory |
| Performance: | * system should respond to the user request to show notification, messages, places of friends or list of places within 5 seconds * The system should respond to 50,000 request per month * The system should be able to respond to 10 requests per second * 1GB database storage |
| Platform requirement: | * The system will be deployed in the first phase on android platform it will be extended in the future to support IOS and a website |
| Extensibility: | * the application should be easily extended to support more users, storage and requests per month |
| Reliability: | * the system should have good exception handling for showing the user the right format of input and also to show a message of the required sensors to be opened such as GPS and Internet. |
| Implementation: | * the system will be implemented using java to build a native mobile app. |
| Security: | * the system should have high systems of security and high confidentiality for user data such as credit cards and passwords |
| Privacy: | * The data of the account including user email, data of birth and all registered data except for name and photo should be encrypted on the database server allowing only authorized access and be shared according to the user allowance`s level of content sharing * Data shouldn’t be shared with any 3rd party without user’s acceptance |
| Open source: | * All documentation of the app including the final report and the code itself will be published allowing for people to use and modify it according to content Creative Commons (cc) allowing more faster developing in coming updates, also all out source functions or libraries that is used in this app are following the Creative Commons license. |
| Back up: | * the database will have scheduled base backups including all user and places data to avoid any type of data loss |

# 

# System Models

## Use Case Model

## C:\Users\Khaled\Downloads\14936919_10211390266097151_1746598976_n.png

## Use Case Tables

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 1 | |
| Use Case Name: | Register | |
| Actors: | Normal User, Premium User, Place owner user | |
| Pre-conditions: | Actor has an email to register by. | |
| Post-conditions: | Update Users database | |
| Flow of events: | **User Action** | **System Action** |
| 1-Enter an e-mail address |  |
| 2-Enter username |  |
|  | 3-Check username availability |
| 4-Enter a password (twice) |  |
| 5-Accept app terms and proceed |  |
|  | 6-Send verification code to user mail |
| 7-Enter the sent code |  |
|  | 8-Update database & Log the user in. |
| Exceptions: | **User Action** | **System Action** |
| 1-Enter an e-mail address |  |
| 2-Enter username |  |
|  | 3-Check username availability (invalid) |
|  | 4-Messege” Enter another username” |
|  | **User Action** | **System Action** |
| 1-Enter an e-mail address |  |
| 2-Enter username |  |
|  | 3-Check username availability |
| 4-Enter a password (twice) |  |
| 5-Accept app terms and proceed |  |
|  | 6-Send verification code to user mail |
| 7-Enter Code sent |  |
|  | 8-Msg” Not Matched code” & send code |
| 9- Enter Code sent |  |
|  | 10- Update database & Log the user in. |
| Includes: | Log in | |
| Notes and Issues: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 2 | |
| Use Case Name: | Log in | |
| Actors: | Normal User, Premium User, Place owner user | |
| Pre-conditions: | Actor has an account on the app | |
| Post-conditions: |  | |
| Flow of events: | **User Action** | **System Action** |
| 1-Enter the username or email |  |
| 2-Enter the password and proceed |  |
|  | 3-Check username & password match |
|  | 4-Check the user Type |
|  | 5-Log in |
| Exceptions: | **User Action** | **System Action** |
| 1-Enter the username or email |  |
| 2-Enter the password and proceed |  |
|  | 3-Check username & password match |
|  | 4- Msg “Invalid username & password” |
| Includes: |  | |
| Notes and Issues: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 3 | |
| Use Case Name: | Forgot Password | |
| Actors: | Normal User, Premium User, Place owner user | |
| Pre-conditions: | Actor has an account on the app | |
| Post-conditions: | Update database | |
| Flow of events: | **User Action** | **System Action** |
| 1-Enter the e-mail address |  |
|  | 2-Send link with code |
| 3-Enter the secret code |  |
| 4-Enter new password (twice) |  |
|  | 5- Update database & Log the user in. |
| Exceptions: | **User Action** | **System Action** |
| 1-Enter the e-mail address |  |
|  | 2-Send link with code |
| 3-Enter the wrong secret code |  |
|  | 4-Msg “Wrong Code” |
| Includes: | Log in | |
| Notes and Issues: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 5 | |
| Use Case Name: | Check-in Registered Places | |
| Actors: | Normal User, Premium User, Place owner user | |
| Pre-conditions: | Actor logged in | |
| Post-conditions: |  | |
| Flow of events: | **User Action** | **System Action** |
| 1-Get Location using GPS or search |  |
|  | 2-Display a list of nearby reg. places |
| 2-Choose from registered places list |  |
| 3-Write description or something |  |
| 4-Post Check-in |  |
| Exceptions: | **User Action** | **System Action** |
| 1-Get Location using GPS or search |  |
|  | 2-Msgs “Enable GPS on your device” |
| Includes: | Post Check-in (ID=4) | |
| Notes and Issues: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 6 | |
| Use Case Name: | Check-in & Register new Places | |
| Actors: | Premium User | |
| Pre-conditions: | Actor logged in | |
| Post-conditions: | Update the database | |
| Flow of events: | **User Action** | **System Action** |
| 1-Get Location using GPS or search |  |
|  | 2-Display a list of nearby reg. places |
| 2-Add new place to the list |  |
| 3-Write description or something |  |
| 4-Post Check-in |  |
| Exceptions: | **User Action** | **System Action** |
| 1-Get Location using GPS or search |  |
|  | 2-Msgs “Enable GPS on your device” |
| Includes: | Post Check-in (ID=4) | |
| Notes and Issues: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 8 | |
| Use Case Name: | Add Friend | |
| Actors: | Normal & Premium Users | |
| Pre-conditions: | Actor logged in | |
| Post-conditions: |  | |
| Flow of events: | **User Action** | **System Action** |
| 1- Search username or use GPSs |  |
|  | 2-Display a list of users or nearby users |
| 2-Choose from a user profile |  |
| 3-Click add friend |  |
|  | 4-Send Notification for the user. |
| Exceptions: | **User Action** | **System Action** |
| 1- Search username or use GPS |  |
|  | 2-Display a list of users or nearby users |
| 2-Choose from a user profile |  |
| 3-Click add friend |  |
|  | 4-Msg “You are already friends” |
| Includes: |  | |
| Notes and Issues: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 9 | |
| Use Case Name: | Accept Friend | |
| Actors: | Normal & Premium Users | |
| Pre-conditions: | Actor logged in, Friend request notification | |
| Post-conditions: | Send Notification for the other user | |
| Flow of events: | **User Action** | **System Action** |
| 1-Open Notifications &select friend request |  |
|  | 2-Display other user profile |
| 2-Accept Friendship request |  |
|  | 3-Send Notification & update lists |
| Exceptions: | **User Action** | **System Action** |
| 1-Open Notifications &select friend request |  |
|  | 2-Display other user profile |
| 2-Reject Friendship request |  |
|  | 3-Send Notification |
| Includes: |  | |
| Notes and Issues: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 10 | |
| Use Case Name: | Rate Place | |
| Actors: | Normal User, Premium User | |
| Pre-conditions: | Actor logged in | |
| Post-conditions: |  | |
| Flow of events: | **User Action** | **System Action** |
| 1-Use GPS or Search for Places |  |
|  | 2-Display list of places |
| 2-Choose place |  |
| 4-Rate place from 1 to 5 stars |  |
|  | 5-Compute Rate Average |
|  | 6-Update database |
| Exceptions: | **User Action** | **System Action** |
|  |  |
| Includes: | Search for Places (ID=24) | |
| Notes and Issues: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 14 | |
| Use Case Name: | Get Direction | |
| Actors: | Normal User, Premium User | |
| Pre-conditions: | Actor logged in | |
| Post-conditions: |  | |
| Flow of events: | **User Action** | **System Action** |
| 1-Use GPS or Search for Places |  |
|  | 2-Display list of places |
| 3-Choose place |  |
| 4-Get direction to place |  |
|  | 5-Display directions on Google Maps |
| Exceptions: | **User Action** | **System Action** |
| 1-Use GPS or Search for Places |  |
|  | 2-Display list of places |
| 3-Choose place |  |
| 4-Get direction to place |  |
|  | 5-Msg “Enable GPS” OR “Install Google Maps” |
| Includes: | Search for Places (ID=24) | |
| Notes and Issues: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 15 | |
| Use Case Name: | Send Message | |
| Actors: | Normal User, Premium User | |
| Pre-conditions: | Actor logged in, Follow the Message receiver | |
| Post-conditions: |  | |
| Flow of events: | **User Action** | **System Action** |
| 1-Choose from friends list or search |  |
|  | 2-Check Relation (friend or follow) |
| 3-Open Conversation |  |
|  | 4-Load all conversation |
| 5-Send new message |  |
|  | 6-Notify Message Receiver |
| Exceptions: | **User Action** | **System Action** |
| 1-Choose from friends list or search |  |
|  | 2-Check Relation (friend or follow) |
|  | 5-Msg “follow first to start conversation” |
| Includes: |  | |
| Notes and Issues: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 16 | |
| Use Case Name: | Create group Chat | |
| Actors: | Normal User, Premium User | |
| Pre-conditions: | Actor logged in, Follow the Message receivers | |
| Post-conditions: |  | |
| Flow of events: | **User Action** | **System Action** |
| 1-Create group chat from options |  |
|  | 2-Display list of friends |
| 3-Choose group members |  |
|  | 4- Notify Message Receiver |
| Exceptions: | **User Action** | **System Action** |
|  |  |
| Includes: |  | |
| Notes and Issues: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 17 | |
| Use Case Name: | Leave group Chat | |
| Actors: | Normal User, Premium User | |
| Pre-conditions: | Actor logged in, Follow the Message receivers | |
| Post-conditions: |  | |
| Flow of events: | **User Action** | **System Action** |
| 1-Choose group chat name from list |  |
|  | 2-Display Conversation |
| 3-leave group |  |
|  | 4- Notify Message Members |
|  | 5-Remove group chat from actor list |
| Exceptions: | **User Action** | **System Action** |
|  |  |
| Includes: |  | |
| Notes and Issues: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 18 | |
| Use Case Name: | Get Brand | |
| Actors: | Premium User | |
| Pre-conditions: | Actor logged in, Premium | |
| Post-conditions: |  | |
| Flow of events: | **User Action** | **System Action** |
| 1-Choose Get brand from options |  |
|  | 2-Check user type (premium only) |
|  | 3-Display brand details form |
| 4-Fill brand info |  |
|  | 5-Check brand name availability |
|  | 6-Creat Page for Brand |
|  | 7-Update database |
| Exceptions: | **User Action** | **System Action** |
| 1-Choose Get brand from option |  |
|  | 2-Check user type (premium only) |
|  | 3-Display brand details form |
| 4-Fill brand info |  |
|  | 5-Check brand name availability |
|  | 6-Msg “Choose another name for the brand” |
| Includes: |  | |
| Notes and Issues: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 20 | |
| Use Case Name: | Follow Company Brand | |
| Actors: | Normal User, Premium User | |
| Pre-conditions: | Actor logged in | |
| Post-conditions: |  | |
| Flow of events: | **User Action** | **System Action** |
| 1-Search Brand name |  |
|  | 2-Display list |
| 3-choose Brand |  |
| 4-Follow |  |
|  | 5- Add user to following list |
| Exceptions: | **User Action** | **System Action** |
|  |  |
| Includes: |  | |
| Notes and Issues: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 21 | |
| Use Case Name: | Text brand Followers | |
| Actors: | Premium User | |
| Pre-conditions: | Actor logged in, Receivers follow the brand | |
| Post-conditions: |  | |
| Flow of events: | **User Action** | **System Action** |
| 1-Open Brand page |  |
| 2-Choose text followers |  |
|  | 3-Display list of followers |
| 4-Choose from followers and send text |  |
|  | 5-Notify receivers |
| Exceptions: | **User Action** | **System Action** |
|  |  |
| Includes: |  | |
| Notes and Issues: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 202 | |
| Use Case Name: | Add tastes | |
| Actors: | Normal User, Premium User, Place owner | |
| Pre-conditions: | Actor logged in | |
| Post-conditions: |  | |
| Flow of events: | **User Action** | **System Action** |
| 1-Choose Add tastes from options |  |
|  | 2-list suggestions |
| 3-write tastes |  |
|  | 4-Notify Followers with new tests |
| Exceptions: | **User Action** | **System Action** |
|  |  |
| Includes: |  | |
| Notes and Issues: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 23 | |
| Use Case Name: | Read Notifications | |
| Actors: | Normal User, Premium User, Place Owner | |
| Pre-conditions: | Actor logged in | |
| Post-conditions: |  | |
| Flow of events: | **User Action** | **System Action** |
| 1-Open Notifications |  |
|  | 2-Display all Notification |
| 3-Click on Notification |  |
|  | 4-Redirect to specific target for each one |
| Exceptions: | **User Action** | **System Action** |
|  |  |
| Includes: |  | |
| Notes and Issues: |  | |

# Ownership Report

|  |  |
| --- | --- |
| **Item** | **Owners** |
| Functional Requirements and software purpose | Mohamed Wasfy |
| Use Case Model and Use Case Tables | *Mohamed Alaa* |
| Nonfunctional Requirements, software scope, Document Purpose and Audience | *Khaled Hefnawy* |
| definitions, acronyms, and abbreviations  Nonfunctional Requirements | *Hazem Hamdy* |

# Policy Regarding Plagiarism:

**Students have collective ownership and responsibility of their project. Any violation of academic honesty will have severe consequences and punishment for ALL team members.**

1. تشجع الكلية على مناقشة الأفكار و تبادل المعلومات و مناقشات الطلاب حيث يعتبر هذا جوهريا لعملية تعليمية سليمة
2. ساعد زملاءك على قدر ما تستطيع و حل لهم مشاكلهم فى الكود و لكن تبادل الحلول غير مقبول و يعتبر غشا.
3. أى حل يتشابه مع أى حل آخر بدرجة تقطع بأنهما منقولان من نفس المصدر سيعتبر أن صاحبيهما قد قاما بالغش.
4. قد توجد على النت برامج مشابهة لما نكتبه هنا أى نسخ من على النت يعتبر غشا يحاسب عليه صاحبه.
5. إذا لم تكن متأكدا أن فعلا ما يعد غشا فلتسأل المعيد أو أستاذ المادة.
6. فى حالة ثبوت الغش سيأخذ الطالب سالب درجة المسألة ، و فى حالة تكرار الغش سيرسب الطالب فى المقرر.