Cairo University  
Faculty of Computers and Information



**CS251**

**Software Engineering I**

**Project Name**

**”** Yellow page social **“**

**“YPS”**

**Team Names**

|  |  |
| --- | --- |
| **ID** | **Name** |
| 20130359 | Ahmed Fathi Rashad |
| 20120036 | Ahmed Adel Ahmed |
| 20130039 | Ahmed Mahmoud AbdelWhab |

# Document Purpose and Audience

The purpose of this document is to present a detailed description of the YPS Software. It will explain the purpose and features of the system, the interfaces of the system, what the system will do, the constraints under which it must operate and how the system will react to external stimuli. This document is intended for client and developers.

# Introduction

## Software Purpose

The purpose of the project YPS is to develop a web Application (locations based social network) that you can search about places and check in it. Also share your Impression about such place with your friends. Also you can make events and celebrations in it by using GPS, mobile and emails. The application will be funny.

## Software Scope

This software system will be a web social network (locations based social network) having some features like: Chatting with only text messaging, group chatting, only 2 types of user normal user (check only the place which he registered), recognizing user's current location using GPS, Relational Microsoft database containing a list of all recommended places, running on Windows and android smart phones.

Loading the web page or the application will not exceeds 3 seconds with the average downloading rate (512 kbps), up-to-date backups every 3 hours using external server.

Not running on Mac OS and Linux, no streaming calls or video steaming, loading the web page or will exceeds 3 seconds if the average downloading rate under (512 kbps) .

## Definitions, Acronyms and Abbreviations

**Normal user:** free account who can check in only the places which he registered in it.

**Premium user:** paid account may be a company or restaurant has the same functions in normal user but also he can check in new places and create brands.

**Brands:** only for premium user which are companies, common places or restaurants to create pages and add its own information and description.

**Tastes:** it is a rate for each place show its particular item and user make it recognized tastes and able to change it any time he need.

**Lists:** places recommended to the user or recommended by his friends**.**

# Requirements

## Functional Requirements

**Sign up:** (create new account) the user will enter his information (name, email, address, and password), then the system will check the validation of the information, and then the system get the user logged in.

**Login:** user enters his email and password, the system will check the existence of the account in database, if exist then get the user in to his account, if not trying again.

**Logout:** the system get the user logged out and show the web homepage.

**Register in place**: the user will choose a place and click register, then the system, will validates the registration using GPS, if true place then evaluated from the user.

**Get list:** the user will click list then the system will give the user a list of popular places in Egypt.

**Chat:** the user will choose his friend and click chat then the system will open a conversation window, then the user can send and receive texts.

**Get Notifications:** if the user click notification the system will list all new events with followed places and friends.

**Add friend:** any user can send a friend requests to other users, then the system will send a notification to the other user who can confirm or not.

**Like, comment & follow:** The user can add comments about a places and he can like posts belongs to it, then the system give every place a rate depending on user impression.

**Personal profile:** user can add personal information, profile picture and so on, then the system get it updated

**Place profile:** Premium user view places profile he owns then the system will allows him to add new updates and so on.

## Non Functional Requirements

**Performance:** Loading the web page or the application will not exceeds 3 seconds with the average downloading rate (512 kbps).

**Availability:** the accessing of software will exceeds 98%. Of the time.

**Reliability:** up-to-date backups every 3 hours using external server.

**Recoverability:** if we defined a loss of data we will use backup of a 3 hours ago.

**Maintainability:** every year we have a new version of our software.

**Security:** each client has account it should be Synchronous and no fuller in access also security password for it by complexity function to difficult integrity on it.

## External Requirements

**Software:** the system will use google maps packages.

Use cases

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 1 | |
| Use Case Name: | Sign up | |
| Actors: | User | |
| Pre-conditions: | age over 18 | |
| Post-conditions: | the system allow the user to begin to create an account | |
| Flow of events: | **User Action** | **System Action** |
| 1- enter name | 1-check name first use |
| 2- enter mail | 2-check validate of mail |
| 3-enter password | 3-check complaxity of password |
| 4-enter confirm password | 4-check password is the same of pervious |
| 5-click submit | 5-the account has been created correctly |
| Exceptions: | if the user enters used account  1. Transaction is disapproved  2. Message to customer to re-enter account  3. user enters correct account | |

0

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 2 | |
| Use Case Name: | Login | |
| Actors: | user | |
| Pre-conditions: | has an account | |
| Post-conditions: | Homepage has been opened | |
| Flow of events: | **User Action** | **System Action** |
| 1- Enter name. | 1-check if the input name is correct |
| 2-enter password | 2-validate the password |
| Exceptions: | if the user enters WRONG password  1-reset password using the email | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 3 | |
| Use Case Name: | Add friend | |
| Actors: | user | |
| Pre-conditions: | The other user enables adding friends | |
| Post-conditions: | Confirm the friend request | |
| Flow of events: | **User Action** | **System Action** |
| 1-send a friend request | 1-send notification |
| Exceptions: | The user don’t approve the friend request | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 4 | |
| Use Case Name: | Register place | |
| Actors: | user | |
| Pre-conditions: | has an account | |
| Post-conditions: | Homepage has been opened | |
| Flow of events: | **User Action** | **System Action** |
| 1- Search the place name. | 1-show a list of all places that related to the required name |
| Exceptions: | The place is not found | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 5 | |
| Use Case Name: | Chat | |
| Actors: | user | |
| Pre-conditions: | Other user is a friend | |
| Post-conditions: | Send a massage | |
| Flow of events: | **User Action** | **System Action** |
| 1- send a massage | 1-the system send a notification |
| Exceptions: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 6 | |
| Use Case Name: | Group chat | |
| Actors: | user | |
| Pre-conditions: | All users are friends of the user | |
| Post-conditions: | Send massage | |
| Flow of events: | **User Action** | **System Action** |
| 1- add friends on this group chat | 1-the system send notification to the friends |
| Exceptions: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 7 | |
| Use Case Name: | Like, Comment & follow | |
| Actors: | user | |
| Pre-conditions: | The user registered the place | |
| Post-conditions: | Receive notification | |
| Flow of events: | **User Action** | **System Action** |
| 1-user may like or comment  2-user may follow this place | 1-send a notification to the admin of the place page  2-send a notification to the user |
| Exceptions: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 8 | |
| Use Case Name: | Rates | |
| Actors: | user | |
| Pre-conditions: | The user registered the place | |
| Post-conditions: |  | |
| Flow of events: | **User Action** | **System Action** |
| User will evaluate the place | The system re-evaluate the place according the evaluation from user |
| Exceptions: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 9 | |
| Use Case Name: | Log out | |
| Actors: | user | |
| Pre-conditions: | The user login | |
| Post-conditions: |  | |
| Flow of events: | **User Action** | **System Action** |
| The user click logout button | The system return the user to the login page |
| Exceptions: |  | |

Use case diagram

