Cairo University

Faculty of Computers and Information

CS251

Software Engineering I

YallaCheck

Software Requirements
Specifications

NYN

October 2015

Table Of Contents

Team

ID	Name	Email	Mobile
20130268	Nada Emad	nada.elkhattib@gmail.com	01092531888
20130276	Nahla Mohamed Ragab	nanosh46@gmail.com	01117240700
20130293	Yara Mansour	yaramansour@outlook.com	01141844077

Document Purpose and Audience

It is a document for describing the overview of the software and the software requirements specifications.

The target audience:

- **Customer:** To ensure that the requirements stated in this document are final as it is very hard to change these things once we start implementing the application "Whole project".
- **Project manager:** To make up a good plan for implementing the project.

Introduction

Software Purpose:

The purpose of the “YallaCheck”software is to develop a location based social network. The application is simple, fast and easy. This location based social network is supposed to connect the people of different communities together by sharing locations and messaging .

Software Scope:

Our system has many features but the main components that will define the system are: creating account that maybe normal or premium account ,checking in , searching for places , adding friends and be able to interact with them thought messages and creating group conversations , companies may create pages and users are able to follow theses pages , receiving notifications , list all locations saved in the system database ,Allowing users to add tastes to their profiles and change their tastes at any time .

Requirements

1. Functional Requirements:

- Creating accounts: user can account with his/her username, email, password, gender and birth date.

  There are two types of accounts:

  1-Normal account: can only check in on already registered places.

  2-Premium account: can check in on a new place and register this place in the places database with new name and can create new brand.

- Multiple mechanisms for forgotten password cases are provided: if the user forgot his/her password he can restore his/her password by answering some security questions.

- Check in : system recognizes user's current position and the user can write something with the check in activity and other users can comment and like this check-in.

  System contains popular places in Egypt
  categorized as following :
  -Restaurants
  -hospitals
  -Facul9es
  -schools
  -airports
  -streets & districts
  -libraries
  -gardens

  Each place is characterized by the la9tude and longitude,and each place has aknown number of check-ins.

Software Requirements Specifications

- Reviews and tips : user can share his/her opinion about the place and also give a recommendation to other users. Each tip has a number of likes and the tips of a place should be sorted in descending order according to number of likes of each tip.

- Extract directions: user can get directions from his/her position to the place positions using 3rd party extension.

- Messaging: users can texts another user and can also create group conversations.

- Brands: premium users such as companies can create pages and allow other users to follow the page and receive special expert tips from these pages when they check-in at certain locations.

Facebook and twitter links for this company must be listed in the page,Also brand's owner can texts following users with new places.

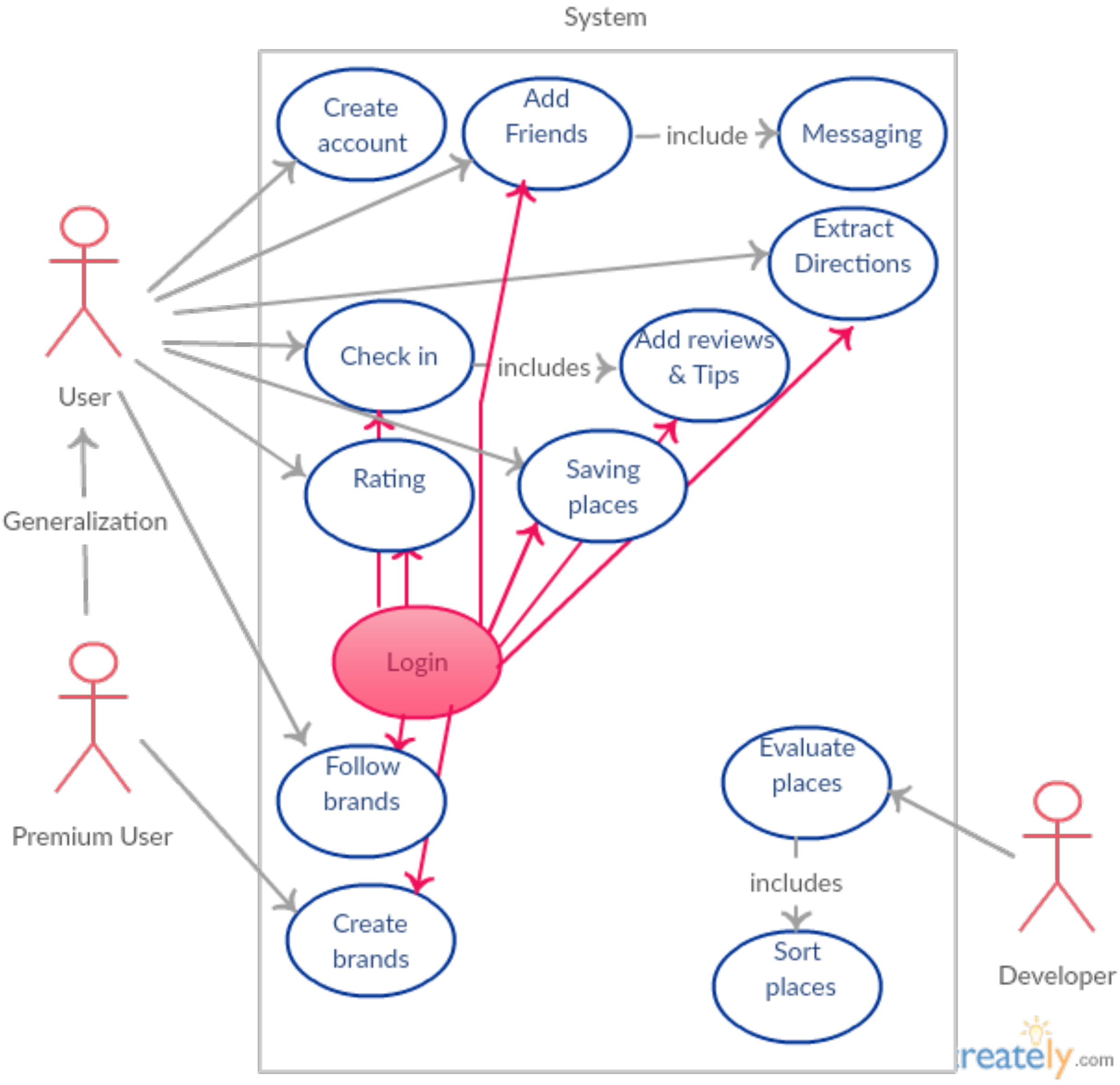
Software Requirements Specifications

2. Non-Functional Requirements:

	Details
Availability	There's no downtime in this application as long as there's internet connection.
Flexibility	System can be updated, some options may be added to the application based on users feedback.
Portability	Software can be installed on a certain platform which is Android operating devices.
Integrity	Users can have their account public so that anyone can follow them and see there feed or they can have it private only for their friends, Users also can protect their personal data on the application from their friends.
Performance	If a user is checking in and messaging at the same time, since messaging is more time-sensitive so the messaging task will be performed first.
Scalability	Software must be scalable, so as when the number of users increases the hardware capacity must increase to handle all the users and their activities.
Usability	Automatic tips while using the application to let the user know how the application works.

Software Requirements Specifications

Use Case Model



Use Case Tables

Use Case ID:	1	
Use Case Name:	Create account.	
Actors:	User	
Pre-conditions:	There’s no account with the same data.	
Post-conditions:	Account has been created with that data and contained in the database.	
Flow of events:	**User Action**	**System Action**
	1- User Enter username, email, password, gender, birth date and Select type of account.	
		2- System Verify user data and signs him/her in.
Exceptions:	**User Action**	**System Action**
	1- User Enter username, email, password, gender, birth date and Select type of account.	
		2- Information already exists. 3- System refuses creating account.

Use Case ID:	2	
Use Case Name:	Check in	
Actors:	User	
Pre-conditions:	There exists the place the user wants to check in at.	
Post-conditions:		
Flow of events:	**User Action**	**System Action**
	1-User login using username and password	
		2- System Verify user data and signs him/her in.
	3- User Enter the place he wants to check in at.	
		4- System gets the place the user wants.
	5- User checks in at the place.	
Exceptions:	**User Action**	**System Action**
	1- User Enters a place.	
		2- Place is not on the database. 3- System returns not found.
	User Action	**System Action**
	1- If premium account, he/she creates the place.	
		2-System inserts the place in the database
Includes:	Add reviews and tips.	

Use Case ID:	3	
Use Case Name:	Add reviews and tips.	
Actors:	User	
Pre-conditions:	Places have some reviews & tips.	
Post-conditions:	User’s review or tip is added to that place.	
Flow of events:	**User Action**	**System Action**
	1-User login using username and password	
		2- System Verify user data and signs him/her in.
	3- User enters the place he/she wants to check-in at and gives reviews and tips about.	
		4- System gets the place for the user.
	5-User leaves tips or reviews or recommendations.	
		6-System submits the tip or review or recommendation on the place.
Exceptions:	**User Action**	**System Action**
	1- User Enters a place.	
		2- Place is not on the database. 3- System returns not found.
	User Action	**System Action**
	1-User login using username and password	

		2-Username or Password is incorrect so user is asked to enter correct data

Use Case ID:	4	
Use Case Name:	Adding Friends.	
Actors:	User.	
Pre-conditions:	User may request to add people to his friends list and accept other requests.	
Post-conditions:	User can interact with other users, See their check-ins, like and comment on them.	
Flow of events:	**User Action**	**System Action**
	1- User login using username and password	
		2- System Verify user data and signs him/her in.
	3- User sends a friend request.	
		4- System notifies the other user that he /she received a request.
	5- The other user accepts or rejects the request.	
		6- If he/she accepted the request the system notifies the first user that now they are friend and can interact.
	User Action	**System Action**
	1-User login using username and password	
		2-Username or Password is incorrect so user is asked to enter correct data
Include:	Messaging.	

Use Case ID:	5	
Use Case Name:	Messaging.	
Actors:	User.	
Pre-conditions:	User is allowed to send messages to people who are in his/her friends list.	
Post-conditions:	The other user receives notification and may reply to the message.	
Flow of events:	**User Action**	**System Action**
	1- User login using username and password	
		2- System Verify user data and signs him/her in.
	3-User sends a message.	
		4- System adds it to the incoming messages for the second user account and notifies him/her that he/her has a message.
	5- The second users chooses to reply or	
		6- If he/she replied, the system deliver it to the first user and so on..
Exceptions:	User Action	System Action
	1- The message fails to arrive	
		2- System notifies the user that his/her message wasn't sent.
	User Action	System Action
	1- User login using username and password.	
		2- Username and password is incorrect, so user is asked to enter

		correct data.

Use Case ID:	6	
Use Case Name:	Rating.	
Actors:	User.	
Pre-conditions:	User is allowed to rate any place from one to ten.	
Post-conditions:	The average place rating changes based on the rate the user gave.	
Flow of events:	**User Action**	**System Action**
	1- User login using username and password	
		2- System Verify user data and signs him/her in.
	3-User rates the place on checking in to it.	
		4- System adds this rate to this place’s rating and changes the average ratings of it.
Exceptions:	User Action	System Action
	1- User login using username and password	
		2- Username and password is incorrect, so user is asked to enter correct data.
Include:	Reviews and tips.	

Use Case ID:	7	
Use Case Name:	Evaluating places	
Actors:	Developer	
Pre-conditions:	Each place has a number of check-ins and rates	
Post-conditions:	Places are evaluated and sorted	
Flow of events:	**User Action**	**System Action**
	1- User checks in a place	
		2- System increases the number of checks in for this place.
	3-Developer evaluates the place based on the number of check ins it has.	
Includes:	Sorting places	

Use Case ID:	8	
Use Case Name:	Sorting Places.	
Actors:	Developer.	
Pre-conditions:	User has a list of places he added.	
Post-conditions:	These places are sorted based on the rating.	
Flow of events:	**User Action**	**System Action**
	1- User login using username and password	
		2- System Verify user data and signs him/her in.
	3-User adds a new place to his list.	
		4- System gets the rate of this place. 5- System sort the list based on this rate.
Exceptions:	User Action	System Action
	1- User login using username and password	
		2- Username and password is incorrect, so user is asked to enter correct data.
	User Action	System Action
	1-Two places have the same rating.	
		2- System sorts them alphabetically

Use Case ID:	9	
Use Case Name:	Saving places.	
Actors:	User.	
Pre-conditions:	User may save any place to get information about it anytime	
Post-conditions:	Saved place is added to list of saved places with information about them.	
Flow of events:	**User Action**	**System Action**
	1- User login using username and password	
		2- System Verify user data and signs him/her in.
	3- User chooses to save a place.	
		4- System creates a list of saved places if not already created and adds this place to the list.
Exceptions:	User Action	System Action
	1- User login using username and password	
		2- Username and password is incorrect, so user is asked to enter correct data.

Use Case ID:	10	
Use Case Name:	Extract directions.	
Actors:	User.	
Pre-conditions:	User may want to get directions to any place in the database of the application.	
Post-conditions:	User has the directions to that place.	
Flow of events:	**User Action**	**System Action**
	1- User login using username and password	
		2- System Verify user data and signs him/her in.
	3- User enters the name of the place he wants to go to.	
		4- System recognizes the user’s current location. 5- System chooses the shortest path and provides the user with the directions of it.
Exceptions:	User Action	System Action
	1- User login using username and password	
		2- Username and password is incorrect, so user is asked to enter correct data.
	User Action	System Action
	1- User enters a place, which is not in the database.	

		2- System notifies the user that the place he entered is not recognized.

Use Case ID:	11	
Use Case Name:	Create brands	
Actors:	Premium user	
Pre-conditions:	Company has a premium user and want to create a page.	
Post-conditions:	New page "Brand" is created for that company.	
Flow of events:	**User Action**	**System Action**
	1-Premium user login using username and password	
		2-System verify user data & login him
	3- Premium user Enter the brand name he wants to create.	
		4- System creates the page the user wants.
Exceptions:	**User Action**	**System Action**
	1-Premium user login using username and password	
		2-Username or Password is incorrect so user is asked to enter correct data
	User Action	**System Action**
	1-Premium user login using username and password	

		2-System verify user data & login him
	3- Premium user Enter the brand name he wants to create.	
		4- An existing brand has the same name, Page isn't created

Use Case ID:	12	
Use Case Name:	Follow brands.	
Actors:	User	
Pre-conditions:	User doesn’t have any information about this brand and want to get tips.	
Post-conditions:	Receive information and expert tips on this brand.	
Flow of events:	**User Action**	**System Action**
	1-User login using username and password.	
		2-System verify user data & login him.
	3- User searches for the brand he wants to follow.	
		4- System gets the brand for the user.
	5-User follow that brands	
Exceptions:	**User Action**	**System Action**
	1-User login using username and password	
		2-Username or Password is incorrect so user is asked to enter correct data
	User Action	**System Action**
	1-User login using username and password.	
		2-System verify user data & login him.

	3- User searches for the brand he wants to follow.	
		4-Brand isn’t on the database. 5- System returns not found.