Cairo University  
Faculty of Computers and Information



**CS251**

**Software Engineering I**

Masroka

Software Requirements Specifications

November, 2018

# Team

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Email** | **Mobile** |
| 20160322 | Mostafa Mahmoud Abd El-Aleem | Mostafa.ma7moud@hotmail.com | 01125880484 |
| 20160320 | Islam Ibrahim Amin | eslam98ebraheem@gmail.com | 01123475588 |

# Document Purpose and Audience

* This document completely describes the system in terms of functional and nonfunctional requirements and serves as a contractual basis between the client and the developers.
* The audience for the SRS includes the client, the users and the project management. The first part of the document, including use cases and nonfunctional requirements, is written during requirements elicitation.

# Introduction

## Software Purpose

* The main purpose of this software is to help making a connection between two type of people the first is a person who lost an item and the second is a person who found that item so they can meet to get this lost item back for its owner.

## Software Scope

* If you found a lost item take a photo of it and post it to the application.
* If you lost an item open the application and search for it and check periodically.

# Requirements

## Functional Requirements

* **Join the application**

Create your own account using Google mail or yahoo any mail. Enter your username and password enter some information about yourself like your name, phone number ...etc.

* **Creating post**

All thing is when you find something, take a picture for the founded item and create a post with some details not all details. Anyone installed the application before, could see the post and comment and communicate with the post creator.

* **Searching for lost item**

When someone lose an item, he can open the application and try to search for the lost item, he can search by apply a lot of filters that make the area of searching closer and closer by applying more filters like choosing the city, type, place of losing the item and more of categorized filters.

* **Finding the owner**

During communication ask the person about special something in shared item, if he reply correctly, he is the right owner. Determine a date to take his own things. after the right owner get his lost, must delete the post.

## Non Functional Requirements

**Usability**

1. It is easy to use it, all thing is when you find item, take a picture for the founded item and create a post with some details not all details.
2. Allow to anyone to comment and communicate with founder throw the application
3. That is search space you can use it if you want to search for anything you lost.
4. The application improve trust between users.

**Reliability & security**

1. 1-The application should be crash safe in 95% of its runtime.
2. 2-the application is using secured protocols.

**Performance**

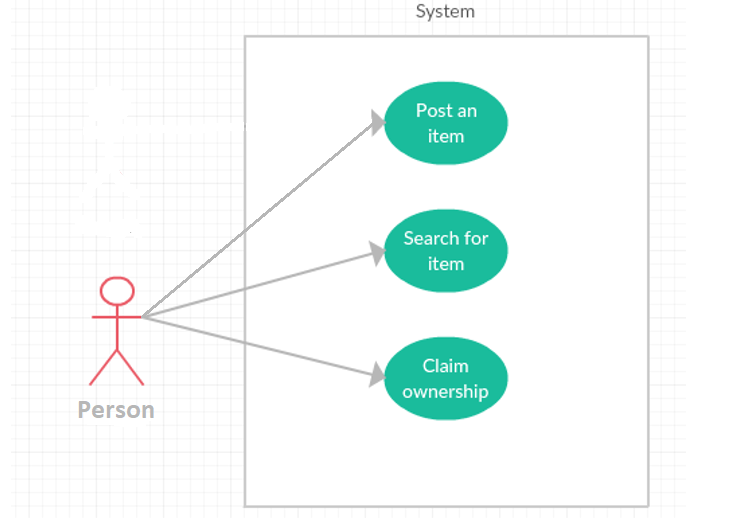
1. 1-Short response time: The loading time of the application must be smaller than 10 seconds.
2. 2-All other response times must be below 2 seconds.
3. 3-the post must have deleted after arriving the shared item to the right owner to reduce size of application to achieve high Performance.

**Supportability**

1. It must support adding various items
2. Support adding a post at any time without waiting to approval form admin or someone else.
3. Easy communication between person and owner.
4. The shared item must back to the right owner.

# System Models

# Use Case Model



## 

## Use Case Tables

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 1 | |
| Use Case Name: | Post an item | |
| Actors: | Person | |
| Pre-conditions: | Person found lost item | |
| Flow of events: | **User Action** | **System Action** |
| 1- User open new post. |  |
|  | 2- System show new post form. |
| 3- User upload photo and fill the form. |  |
|  | 4- System retrieves form info and verify it then publish the post. |
| Exceptions: | **User Action** | **System Action** |
| 1- User Enter invalid information. |  |
|  | 2- System reject the post. |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 2 | |
| Use Case Name: | Search an item | |
| Actors: | Person | |
| Pre-conditions: | Person lost an item | |
| Flow of events: | **User Action** | **System Action** |
| 1- User open and fill search form. |  |
|  | 2- System show all possible item with the similar information. |
| Exceptions: | **User Action** | **System Action** |
| 1- User Enter invalid information. |  |
|  | 2- System ask for correct information or item maybe not found. |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 3 | |
| Use Case Name: | Claim ownership | |
| Actors: | Person | |
| Pre-conditions: | Person found lost item on the application | |
| Flow of events: | **User Action** | **System Action** |
| 1- User open item’s post. |  |
|  | 2- System show ownership form. |
| 3- User fill the form. |  |
|  | 4- System retrieves form info and verify it correctness.  5-System show the owner all possible contact info of the item founder. |
| Exceptions: | **User Action** | **System Action** |
| 1- User Enter invalid ownership information. |  |
|  | 2- System reject the ownership. |

# Ownership Report

|  |  |
| --- | --- |
| **Item** | **Owners** |
| Use case model – Document Purpose and Audience | *Mostafa Mahmoud* |
| Requirements - Introduction | *Islam Ibrahim* |

**Git Repository Link:** https://github.com/mostafa-ma7moud/Masroka