Cairo University  
Faculty of Computers and Information



**CS251**

**Software Engineering I**

Project Name

Software Requirements Specifications

Version 1.0

Mohammad El-Ramly

m.elramly@fci-cu.edu.eg

March 2020

Contents

[Revision History 2](#_Toc37925644)

[Document Purpose and Audience 3](#_Toc37925645)

[Introduction 3](#_Toc37925646)

[Software Purpose 3](#_Toc37925647)

[Software Scope 3](#_Toc37925648)

[Definitions, acronyms, and abbreviations 3](#_Toc37925649)

[Requirements 4](#_Toc37925650)

[Functional Requirements 4](#_Toc37925651)

[Non Functional Requirements 5](#_Toc37925652)

[System Models 6](#_Toc37925653)

[Use Case Model 6](#_Toc37925654)

[Sample Use Case Tables 7](#_Toc37925655)

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Email** | **Mobile** |
| 00000000 | Mohammad El-Ramly | [m.elramly@fci-cu.edu.eg](mailto:m.elramly@fci-cu.edu.eg) | 000000000 |
| 00000000 | Hoda Tawakl | h.tawakl@fci-cu.edu.eg | 000000000 |
| 00000000 | Marwa Hussein | m.hussein@fci-cu.edu.eg | 000000000 |

|  |
| --- |
|  |

# Document Purpose and Audience

* This document includes SRS description for GoFo football playground booking system. It describes the functions of the system and the uses cases.
* It is meant for all stakeholders, especially the client to understand what features will in the system.
* It also serves as the basis for the contract between the company and the client.
* It serves as a guide for the developers to understand what they will develop.

# Introduction

## Software Purpose

This software is a system for booking playing hours in football playgrounds.

## Software Scope

* The software has some stakeholders such as: Player, administrator, Playground owner
* The player can reserve playground and pay for it via an eWallet (Externally connected to an eWallet system). He can view his bookings or cancel them.
* The playground owner can register his playground and its availability, receive money from bookings and view the booking and allows.
* The administrator to manage the reservation process and can interfere with necessary actions against fraudulent acts.

## Definitions, acronyms, and abbreviations

* Playground is a football field for rent
* eWallet is an external electronic payment system
* Booking is booking an hour or more on a football playground

# Requirements

## Functional Requirements

* The software allows the **playground owner** to:
  + **Register** in the system with and ID, password, name and valid email. A verification code will be sent to his email that he must reenter to complete the registration. System verifies that the provided password is strong. (<https://tech.wayne.edu/about/strong-password-standard>)
  + **Login** with his account credentials.
  + **Create profile** with his address and mobile. A verification code is sent to his mobile to verify him.
  + **Add playground.** This allows the owner to add a new playground by entering its name, its location (as address and as GPS location), its description, its availability, booking number and price /hour. he can possibly enter some photos or link to an external site with more information. The playground will need to be verified and approved by the admin.
  + **Update playground** information by changing its availability in certain hours or days or changing its description, price, booking number, link or photos. He cannot change the availability of booked hours.
  + **View bookings** by seeing the bookings made for his playground(s).
  + **Check the status of his eWallet.** This will just connect him with the eWallet service which is run by a third party.
* The software allows the **player** to :
  + **Register** in the system with and ID, password, name and valid email. A verification code will be sent to his email that he must reenter to complete the registration.
  + **Login** with his account credentials.
  + **View** available playground in an area or around his location and the time slots available. He can also **filter** them by selecting an area, specific range of date and time , available only hours, price range.
  + **Book playground** in specific time slot by selecting it and paying for it using his eWallet. Both payer and owner get a notification of the booking. The booked hour is not available for booking by any other player.
  + **Create a team** by registering their emails and names.Then he can **modify** or change his team.
  + **Send invitation** to his team or to other players or a mix of his team members and others.
  + **Cancel booking** and get refunded if within cancellation period.
* The software allows the **Administrator** to:
  + **Login** with his account credentials.
  + **Approve playground** by checking the validity of the information and entered by the owner.
  + **Suspend the playground** by deactivating it and sending a notice to the owner. This should stop any future bookings but should not affect existing ones.
  + **Delete the playground** completely from the system. But it should remain visible until all bookings made on it pass or get cancelled.It should be suspended till all bookings are finished.
  + **Activate playground** by lifting suspension and allowing further booking on it.

## Non Functional Requirements

* **Platform:**  System will have a Web front-end, supporting major browsers like Chrome, IE and Firefox. It will also have a mobile app supporting Android and iPhone.
* **Development tools:**  Back-end will be developed with Java, Spring and PostgreSQL.Mobile app should b developed with a cross-platform technology.
* **Availability:**  Expected system availability is 24x7. Downtime should not exceed 30 min per week, except cases of unforeseen downtimes or irregular updates.
* **Usability:**  A novice user should be able to use the system with ease within30 min.
* **Safety**: Application should process authenticate users with a password. Passwords should be strong following XYZ guidelines. Web system should be tested for vulnerabilities. Communications between mobile app and server should be encrypted.
* **Integrity:** The application must be ready to eliminate collisions in case of incorrect user requests. In order to avoid these problems, the application should use the principle of transactions to eliminate logical data inconsistencies.

# System Models

## Use Case Model



## Sample Use Case Tables

* **This are samples of use cases; the rest of them should follow the same format**

|  |  |  |
| --- | --- | --- |
| **Use Case ID:** | **UC1.1** | |
| Use Case Name: | Sign Up / Register | |
| Actors: | Player or Playground Owner | |
| Pre-conditions: |  | |
| Post-conditions: | The player or playground owner are added to the system. | |
| Flow of events: | **User Action** | **System Action** |
| 1- Actor open welcome screen |  |
| 2- Actor chooses Sign Up / Register |  |
|  | 3- System shows registration screen |
| 4- Actor provides an ID, password, name and valid email. |  |
|  | 5- System verifies that (1) all fields are filled, (2) Email is valid, (3) password is strong.  6- System sends a verification email to the actor with a secret code. |
| 7- Actor enters verification code. |  |
|  | 8- System confirms actor registration |
| Exception 1: Payment Failure | **User Action** | **System Action** |
|  | 5- System finds invalid data like (1) wrong email format, (2) missing fields or (3) weak password  6- System asks the actor to fix data. |
| 7- Actor fixes the wrong data. |  |
|  | 8- System goes to step 5 |
| Exception 2: Player Cancels | 7- Actor doesn't enter code in 30 min |  |
|  | 8- System cancel egisteration. |
| Includes: |  | |
| Notes and Issues: |  | |

|  |  |  |
| --- | --- | --- |
| **Use Case ID:** | **UC1.2** | |
| Use Case Name: | Add a Playground | |
| Actors: | Playground Owner | |
| Pre-conditions: | Playground Owner is logged in | |
| Post-conditions: | The playground is added | |
| Flow of events: | **User Action** | **System Action** |
| 1- Owner selects to register a playground |  |
|  | 2- Systems open playground registration form |
| 3- Owner adds playground name, its location (as address and as GPS location), its description, its availability, booking number and price /hour and (optionally) some photos or URL link. |  |
|  | 4- System verifies that all the fields are filled.  5- System sends to the admin to notify him of this request. |
|  | 6- System receives confirmation from admin. (Approve Playground **UC X.X)** |
|  | 7- Playground is registered.  8- Owner is notified. |
| Exception 1: Approval rejected | **User Action** | **System Action** |
|  | 6- System receives rejection. |
|  | 7- System notifies owner. |
| Includes: | Approve Playground **UC X.X** | |
| Notes and Issues: |  | |

|  |  |  |
| --- | --- | --- |
| **Use Case ID:** | **UC1.3** | |
| Use Case Name: | Book Playground (for specific time slot(s)) | |
| Actors: | Player | |
| Pre-conditions: | 1. Player is logged in 2. The playground is activated (not suspended or deleted) | |
| Post-conditions: | The time slot is booked successfully on the playground and is not available to other players | |
| Flow of events: | **User Action** | **System Action** |
| 1- Player views playgrounds and Views Playing Hours (UC 1.4) |  |
| 2- Player selects playing hours for the desired playground. |  |
|  | 3- System calculates the total  4-The system prompts the user to confirm or cancel booking. |
| 5- Player chooses confirm |  |
|  | 6- System connects the user to eWallent system. |
| 7- Player pays for the booking |  |
|  | 8- System confirms booking and sends emails to owner and player |
| Exception 1: Payment Failure | **User Action** | **System Action** |
| 7- Player payment fails |  |
|  | 8- Cancel booking and inform user. |
| Exception 2: Player Cancels | 5- Player chooses to cancel |  |
|  | 6- Cancel booking and inform user. |
| Includes: | UC 1.4: View Playing Hours | |
| Notes and Issues: |  | |

|  |  |  |
| --- | --- | --- |
| **Use Case ID:** | **UC1.4** | |
| Use Case Name: | View Playing Hours | |
| Actors: | Player | |
| Pre-conditions: | 1. Player is logged in | |
| Post-conditions: |  | |
| Flow of events: | **User Action** | **System Action** |
| 1- Player selects to view playgrounds |  |
|  | 2- System lists all active playgrounds |
| 3- Player narrows his selection and Filters Playing Hours (UC1.5) (Optional) |  |
|  | 4- System updates active playgrounds |
| 5- Player clicks on a playground to view its details and availability.  7- Player repeats 5 till finished |  |
| Exceptions: | **User Action** | **System Action** |
|  |  |
| Extensions: | UC 1.5: Filter Hours | |
| Notes and Issues: |  | |

|  |  |  |
| --- | --- | --- |
| **Use Case ID:** | **UC1.5** | |
| Use Case Name: | Filter Hours | |
| Actors: | Player | |
| Pre-conditions: | 1. Player is logged in 2. Player is in View Playing Hours screen | |
| Post-conditions: |  | |
| Flow of events: | **User Action** | **System Action** |
| 1- Player selects to filter playgrounds and playing hours |  |
|  | 2- Systems open filter page |
| 3- User selects a city and area (from lists), or enter a location (GPS), specify dates and times on a calendar and select a price range per hour or any price. He can choose to show ones available in these constraints only. |  |
|  | 4- System displays a list of active playground in the selected area or within 5 km from given location. |
| 5- Continue with UC1.4 |  |
| Exception 1: | **User Action** | **System Action** |
|  |  |
| Extends: | UC 1.4: View Playing Hours | |
| Notes and Issues: |  | |