Cairo University  
Faculty of Computers and Information



**CS251**

**Software Engineering I**

GoFo

Software Requirements Specifications

Version 1.0

March 2020

Contents

[Team 3](#_Toc35185595)

[Document Purpose and Audience 3](#_Toc35185596)

[Introduction 3](#_Toc35185597)

[Software Purpose 3](#_Toc35185598)

[Software Scope 3](#_Toc35185599)

[Definitions, acronyms, and abbreviations 4](#_Toc35185600)

[Requirements 4](#_Toc35185601)

[Functional Requirements 4](#_Toc35185602)

[Non Functional Requirements 5](#_Toc35185603)

[System Models 6](#_Toc35185604)

[Use Case Model 6](#_Toc35185605)

[Use Case Tables 7](#_Toc35185607)

# Team

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Email** | **Mobile** |
| 20180250 | Mahmoud Ashraf | [ma5027300@gmail.com](mailto:ma5027300@gmail.com) | 01102488789 |
| 20180304 | Nader Fikry | [naderfikry245@gmail.com](mailto:naderfikry245@gmail.com) | 01097632795 |
| 20180193 | Fady Emad | [fadyemad14705@gmail.com](mailto:fadyemad14705@gmail.com) | 01203732443 |

# Document Purpose and Audience

* This document focuses on reducing the time and effort for booking a playground.
* It serves everyone interesting in booking a playground also the playground owners.

# Introduction

## Software Purpose

* The purpose of this software is to make it easy for those who want to book a playground by saving time and effort.

## Software Scope

There are some major features for this software:

* Display the available playgrounds.
* Display the available time slots for these playgrounds.
* Display the location for these playgrounds.
* Display the size and the price for the available playgrounds.

## Definitions, acronyms, and abbreviations

|  |  |
| --- | --- |
|  | Description |
| Slot | It is the way the time is measured in the playground.  1 Slot = 1 hour. |
| State | It is the condition of each slot in the playground.  It can be “Taken” or “Available”. |
| Suspend | It means that the playground is stopped from work may be from complaints or from other issues. |

# Requirements

## Functional Requirements

* Making a profile.
* Register a playground.
* Activate a playground.
* Change playground available slots.
* Display available playgrounds.
* Filter available playgrounds.
* Set favorite team.
* Book a playground.
* Select a free slot.
* Calculate the total price.
* Update the state of booked slots.
* Delete a playground.
* Suspend a playground.

## Non Functional Requirements

|  |  |
| --- | --- |
|  | Details |
| Performance | * Activate a playground, Change playground slots, Update the system, Delete a playground and Suspend a playground will be done in 1 to 10 seconds. |
| Platform | * This software will work on both iOS & Android. |

# System Models

## Use Case Model

# D:\UseCaseDiagram.png

## Use Case Tables

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 1 | |
| Use Case Name: | Making a profile. | |
| Actors: | Player & Playground owner. | |
| Pre-conditions: | None. | |
| Post-conditions: | None. | |
| Flow of events: | **User Action** | **System Action** |
| 1- Enter name. |  |
| 2- Enter ID. |  |
| 3- Enter password. |  |
|  | 4- Create profile successfully. |
|  |  |
| Exceptions: | **User Action** | **System Action** |
| 1- Enter name, ID, password. |  |
|  | 2- Name is already taken. |
| Includes: |  | |
| Notes and Issues: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 2 | |
| Use Case Name: | Register a playground. | |
| Actors: | Playground owner. | |
| Pre-conditions: | Making a profile. | |
| Post-conditions: | None. | |
| Flow of events: | **User Action** | **System Action** |
| 1- Ask to register a playground. |  |
|  | 2- Ask to fill more information (Price, Size, Location, Free slots). |
| 3- Fill the information. |  |
|  | 4- Registered a playground successfully (Activate the playground). |
| 5- Change available slots. | 6- Slots changed successfully. |
| Exceptions: | **User Action** | **System Action** |
| 1- Enter invalid information. |  |
|  | 2- Can’t activate the playground. |
| Includes: | Size, Location, Price, Free slots. | |
| Notes and Issues: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 3 | |
| Use Case Name: | Display playground. | |
| Actors: | Player. | |
| Pre-conditions: | Making a profile. | |
| Post-conditions: | Booking a playground. | |
| Flow of events: | **User Action** | **System Action** |
| 1- Display playground. |  |
|  | 2- Display all playgrounds. |
| 3- Filter playgrounds (by available sots, by location). |  |
|  | 4- Filter and display playgrounds. |
|  |  |
| Exceptions: | **User Action** | **System Action** |
|  |  |
|  |  |
| Includes: |  | |
| Notes and Issues: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 7 | |
| Use Case Name: | Set Favorite Team | |
| Actors: | Player | |
| Pre-conditions: | Making a Profile | |
| Post-conditions: |  | |
| Flow of events: | **User Action** | **System Action** |
| 1 - Actor enters the name of the team he wants to set as favorite |  |
|  | 2 - System verifies that the team exists |
| Exceptions: | **User Action** | **System Action** |
| 1 - Actor enters the name of the team he wants to set as favorite |  |
|  | 2 – The Team specified by the actor does not exist |
| Includes: | Verify Team In Database | |
| Notes and Issues: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 8 | |
| Use Case Name: | Book a Playground | |
| Actors: | Player | |
| Pre-conditions: | Making a Profile | |
| Post-conditions: |  | |
| Flow of events: | **User Action** | **System Action** |
| 1 - Actor enters the name of the playground he wants to book |  |
|  | 2 - System verifies that the playground exists |
|  | 3 – Actor chooses the ‘Book’ option |  |
|  |  | 4 – System displays slots for the chosen playground |
|  | 5 - Actor chooses slot(s) he wants to book |  |
|  |  | 6 - System verifies that the slot(s) are free of other matches |
|  |  | 7 – System updates slot(s) values in database so these slot(s) are now busy for other bookers |
| Exceptions: | **User Action** | **System Action** |
| 1 - Actor enters the name of the playground he wants to book |  |
|  | 2 – Playground does not exist |
|  | 3 – Actor chooses the ‘Book’ option |  |
|  |  | 4 – System unable to find any slots for the chosen playground |
|  | 5 - Actor chooses slot(s) he wants to book |  |
|  |  | 6 – Slot(s) is busy |
| Includes: | Verify Playground in Database, Verify Payment Card Details, Check for Enough Funds, Select a Free Slot | |
| Notes and Issues: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 9 | |
| Use Case Name: | Select a Free Slot | |
| Actors: | Player | |
| Pre-conditions: | Book a Playground | |
| Post-conditions: |  | |
| Flow of events: | **User Action** | **System Action** |
| 1 - Actor chooses slot(s) he wants to book |  |
|  | 2 - System verifies that the slot(s) are free of other matches |
|  |  | 3 – System updates slot(s) values in database so these slot(s) are now busy for other bookers |
| Exceptions: | **User Action** | **System Action** |
| 1 - Actor chooses slot(s) he wants to book |  |
|  | 2 – Slot(s) is busy |
| Includes: | Check slot(s), Update the state of booked slots | |
| Notes and Issues: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 10 | |
| Use Case Name: | Update the state of booked slots | |
| Actors: | System | |
| Pre-conditions: | Book a Playground | |
| Post-conditions: | Slot(s) are busy | |
| Flow of events: | **User Action** | **System Action** |
| 1 - Actor communicates with database to establish connection |  |
| 2 – Actor updates the specified slots as ‘busy’ |  |
| Exceptions: | **User Action** | **System Action** |
| 1 – Unable to communicate with database |  |
| Includes: | Check slot(s) | |
| Notes and Issues: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 11 | |
| Use Case Name: | Delete a Playground | |
| Actors: | Administrator | |
| Pre-conditions: | Display Available Playgrounds | |
| Post-conditions: |  | |
| Flow of events: | **User Action** | **System Action** |
| 1 - Actor searches for the playground he wants to delete |  |
|  | 2 - System verifies that the playground exists |
|  | 3 – Actor selects the playground and chooses the ‘delete’ button |  |
|  |  | 4 – System deletes the playground and updates the database |
| Exceptions: | **User Action** | **System Action** |
| 1 - Actor searches for the playground he wants to delete |  |
|  | 2 – Playground does not exist |
|  | 3 – Actor selects the playground and chooses the ‘delete’ option |  |
|  |  | 4 – System unable to delete the playground or communicate with the database |
| Includes: | Verify Playground in Database, Delete Playground, Update Database | |
| Notes and Issues: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 12 | |
| Use Case Name: | Suspend a Playground | |
| Actors: | Administrator | |
| Pre-conditions: | Display Available Playgrounds | |
| Post-conditions: |  | |
| Flow of events: | **User Action** | **System Action** |
| 1 - Actor searches for the playground he wants to suspend |  |
|  | 2 - System verifies that the playground exists |
|  | 3 – Actor selects the playground and chooses the ‘suspend’ option |  |
|  |  | 4 – System suspends the playground and updates the database |
| Exceptions: | **User Action** | **System Action** |
| 1 - Actor searches for the playground he wants to suspend |  |
|  | 2 – Playground does not exist |
|  | 3 – Actor selects the playground and chooses the ‘suspend’ option |  |
|  |  | 4 – System unable to suspend the playground or communicate with the database |
| Includes: | Verify Playground in Database, Suspend Playground, Update Database | |
| Notes and Issues: |  | |