

**Cairo University**

**Faculty of Computers and Artificial Intelligence**



# **CS251**

## **Intro. to Software Engineering**

### **GoFo - “Go Football”**

#### **Software Requirements Specifications**

**Version 1.0**

**May / 2021**



CS251: Phase 1 – Team Name  
Project: <Project Name>

## Software Requirements Specifications

### Contents

Team .....	2
Document Purpose and Audience .....	3
Introduction .....	3
Software Purpose .....	3
Software Scope .....	3
Definitions, acronyms, and abbreviations .....	4
Requirements .....	4
Functional Requirements .....	4
Non-Functional Requirements .....	5
System Models .....	6
Use Case Model .....	6
Enriched User Stories .....	6
System Navigation Map .....	7
Tools .....	8
Ownership Report .....	8

### Team

ID	Name	Email	Mobile
20190059	Ahmed Magdy Abd-El-Fattah Baseony	<a href="mailto:ahmedmagdy3abdelfattah@gmail.com">ahmedmagdy3abdelfattah@gmail.com</a>	01112223040
20190547	Mostafa Nassr El-deen El-sayed	<a href="mailto:mostafanassr200083@gmail.com">mostafanassr200083@gmail.com</a>	01112165365
20190552	Manar Emad El-din Hassan	<a href="mailto:Manaremad5112000@gmail.com">Manaremad5112000@gmail.com</a>	01119172229



# CS251: Phase 1 – Team Name

## Project: <Project Name>

# Software Requirements Specifications

## Document Purpose and Audience

Egyptians from old to young are fond of Football, so they always practice. Lots of young people book playgrounds to play in, but they face a lot of problems like: failed to find a booking at the needed hours because it is such a busy day, spent a lot of time, transportation cost and effort to go to all these playgrounds or call the owners and others, so we resorted to this project to help who want to book a playground and playground owners (people who expected to read it) in the simplest way.

We have motivated to this project to solve these problems; our system is a football playground booking system. It benefits both the players and the playground owners. An administrator who has the right to delete a playground or suspend it manages the system's overall activities and ensures that no fraud occurs. Anyone can register and create a profile on the system. He can see playgrounds in his immediate vicinity, in a distant area, or all of them.



It will not be our intended audience a specific person will be a group of people who is interested in booking a playground to play football (the players), the playground owners who wants to register his playground.

## Software Purpose

This software is made for playground owners who can rent their playground through the system to other players for a specific amount of time and money, also made for players who are searching for playgrounds to rent so the system allows them to search for the nearest playground available and the system is safe from fraud because there is an administrator who supervises anything that happens in the system.

## Software Scope

The booking system for football playgrounds is built to serve two characters, the players and playground owners, and they both have different accessibility to fulfill their needs. The system has an administrator who manages the system's overall activities and ensures that no fraud occurs.

**The administrator:** An administrator has the authority to remove or suspend a playground. This is generally the case if the owner participates in unethical behavior. A playground is not available until it has been reviewed by the administrator, who can verify the accuracy of the information provided.

**Playground owners:** The person who wishes to have his/ her playground registered. He/she has the ability to set and adjust the hours the playground is available for booking, and has access to his/her reservations. He/she can also check the balance in his/her e-Wallet (Digital Wallet).

**The player:** The person who wishes to reserve a playground. He/she can display the playgrounds in a specific location on specific dates and book them if they are available to book.



# CS251: Phase 1 – Team Name

## Project: <Project Name>

## Software Requirements Specifications

### Definitions, acronyms, and abbreviations

- 1- **The administrator:** he has the right to delete a playground if the owner does some fraudulent activities so he keeps the software safe from frauds.
- 2- **e-Wallet:** an electronic payment system used to allow players to pay for their booking and allow ground owners to collect their money so any user can check the money in his wallet or transfer some money to another e-Wallet.

## Requirements

### Functional Requirements

Functional Requirement No.	Function requirement description
FR#1	A playground owner should be able to register his playground after registering himself as a user on the system, he also registers its name, its location, its size, the available hours, the price per hour and the cancelation period.
FR#2	An administrator should be able to activate the playground after he makes sure that the given information about the playground is true and should deactivate any playground if he found a fraudulent activity.
FR#3	A player should be able to see the nearest playground available to him or in a specific location on specific dates. He can filter them by the hours and date he selects. He can book a time slot of 1 hours or more if available.
FR#4	There should be an e-Wallet to deal with any financial operations on the software.
FR#5	Booking should include checking available grounds and time slots, Selecting the free time slot(s), calculating the total price from the player's eWallet to the owner's e-Wallet, the system updates the status of the booked slots and optionally the player can send invitation to his team members via email.



CS251: Phase 1 – Team Name  
Project: <Project Name>

## Software Requirements Specifications

### Non-Functional Requirements

- **Performance Requirements**

Requirements ID	Details
PR#1	The system must respond the business operation in less than 3 seconds.
PR#2	The system must respond the message operation to the users (players and playground owners) within 1 second.
PR#3	The system should be compatible with all modern browsers and devices.
PR#4	The system should be reliable.

- **Safety and Security Requirements**

Requirement ID	Details
SSR#1	The system must handle safe login and logout through sessions
SSR#2	The system must be safe from fraudulent activities.

- **Reliability**

Requirement ID	Details
R#1	The system should be consistent and reliable and have exception catching such that unintended results, such as system crashes or data validation failures do not occur.

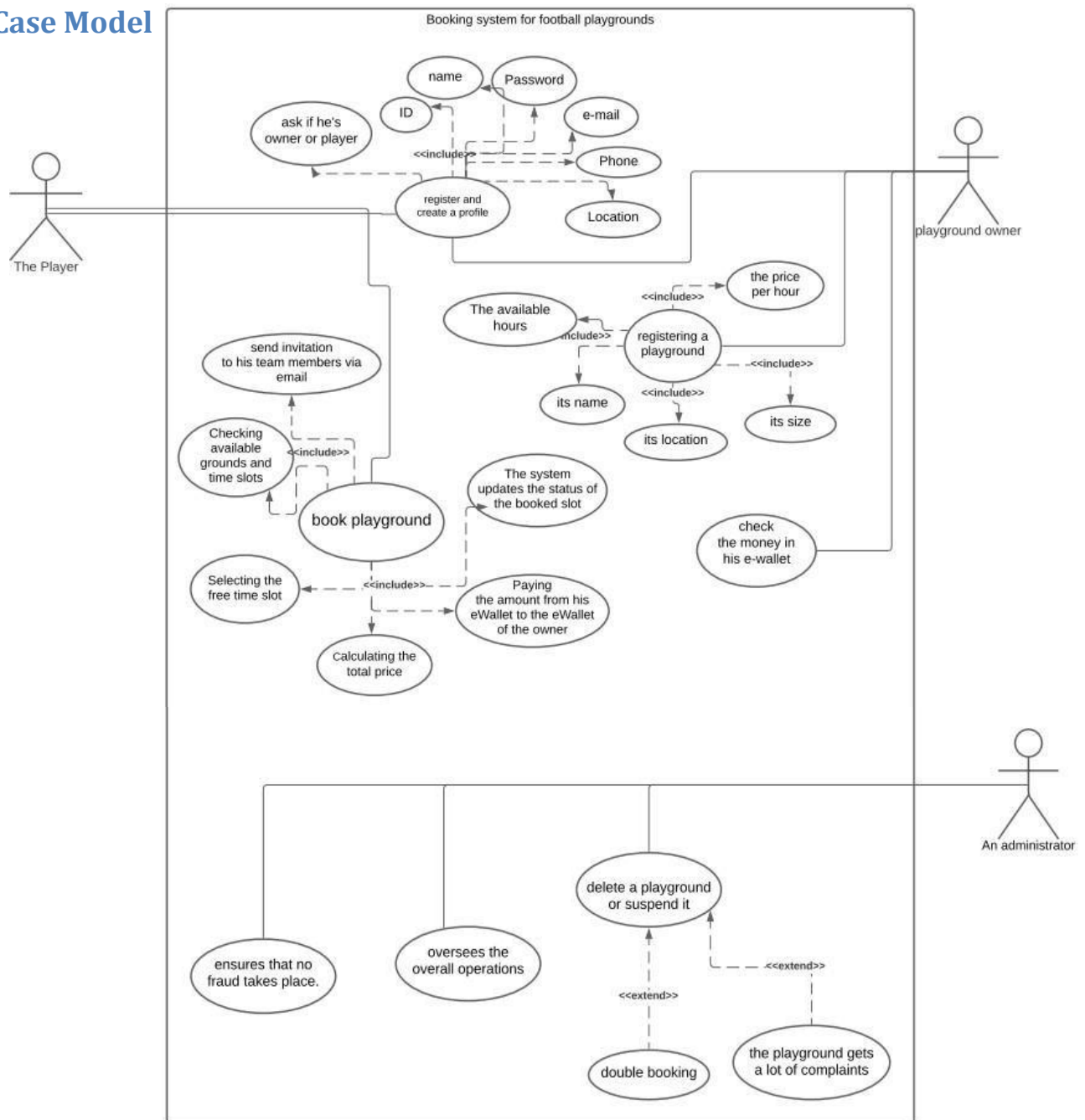
- **Availability**

Requirement ID	Details
A#1	The system must be available 24/7 (24 hours in the 7 days), with no more than 30 mins down time per day.

# Software Requirements Specifications

# System Models

## Use Case Model





# CS251: Phase 1 – Team Name

## Project: <Project Name>

## Software Requirements Specifications

Definition of the Player is the person who always interested in playing sports like football, and practice it by booking playground.

His role is using this program by creating an account to help him to find the right playground.

Definition of the playground owner is the person who opposed his playground for booking.

His role is using this program too by creating an account to oppose his playground with his requests.

Definition of the administrator is the person who oversees the overall operations of the system. His role is deleting a playground or suspending it, ensuring that no fraud takes place.

### Enriched User Stories

- User Story #1**

<b>User Story ID</b>	US #1
<b>User Story Name</b>	Deactivate Playground
<b>Actors</b>	Administrator
<b>Description</b>	Remove or suspend a playground from an owner.
<b>Per condition</b>	Fraudulent activity found.
<b>Post condition</b>	-----
<b>Acceptance Criteria</b>	<b>Given:</b> The owner has a playground. <b>When:</b> He/she double books a playground / A playground gets a lot of complains. <b>Then:</b> The playground gets removed or suspended from the system.

- Scenarios**

#### Normal Scenario

Actor Action	System Response
1- Search for a playground by ID 2- Enter ID	
	3- Playground found. 4- Delete or Suspend
5- Delete	
	5- System deletes the playground



CS251: Phase 1 – Team Name  
Project: <Project Name>

## Software Requirements Specifications

### Exceptional Scenario

Actor Action	System Response
1- Search for a playground by ID 2- Enter ID	
	3- Playground was not found 4- Systems displays an error message

- **Screen Design**

**Screen name:** Application interface

**Screen number:** #1



**Screen name:** Searching for the playground

**Screen number:** #2







CS251: Phase 1 – Team Name  
Project: <Project Name>

## Software Requirements Specifications

**Screen name:** Deleting the playground

**Screen number:** #3

**Screen name:** Playground has been deleted

**Screen number:** #4



- Data Dictionary:**

Element Label	Type/Length	Data Validation / Business Rule
playgroundID	INT	Playground ID



CS251: Phase 1 – Team Name  
Project: <Project Name>

## Software Requirements Specifications

- User Story #2**

<b>User Story ID</b>	US #2
<b>User Story Name</b>	Register a Playground
<b>Actors</b>	Playground Owner
<b>Description</b>	The owner registers a playground after registering himself as a user on the system.
<b>Per condition</b>	A playground owner should sign in with his/her user account as a playground owner
<b>Post condition</b>	A playground is available if it has been reviewed by the administrator
<b>Acceptance Criteria</b>	<p><b>Given:</b> The owner wishes to register a playground.</p> <p><b>When:</b> He/she registers a playground.</p> <p><b>Then:</b> He/she has to wait for the administrator to activate the playground.</p>

- Scenarios**

### Normal Scenario

Actor Action	System Response
1- Sign up a playground 2- Click <b>Sign Up</b>	
	3- System displays the playground registration form
4- The owner enters playground name, location, playground size, available hours, price per hour and cancelation period 5- Click <b>Submit Playground</b>	
	6- Admin verifies playground data 7- The playground has been approved by the Administrator



CS251: Phase 1 – Team Name  
Project: <Project Name>

## Software Requirements Specifications

### Exceptional Scenario

Actor Action	System Response
5- Click <b>Submit Playground</b>	
	6- Admin verifies playground data 7- “This playground has been registered before” 8- The playground has been rejected by the Administrator

- **Screen Design**

**Screen name:** Sign up playground

**Screen number:** #5



**Screen name:** Registration form

**Screen number:** #6





CS251: Phase 1 – Team Name  
Project: <Project Name>

## Software Requirements Specifications

**Screen name:** Submit playground

**Screen number:** #7

GoFo

Playground Registration

Name: Lorem

Location: Lorem

Playground size: Lorem

Available hours: Lorem

Price per Hour: Lorem

Cancellation Period: Lorem

Submit

**Screen name:** Playground is being processed

**Screen number:** #8

GoFo

Playground Registration

Playground is being processed by the administrator..

- Data Dictionary:**

Element Label	Type/Length	Data Validation / Business Rule
playgroundName	STRING	Playground Name
playgroundLocation	STRING	Playground Location
playgroundSize	INT	Playground Size
availableHours	INT	Available Hours
pricePerHour	DOUBLE	Price Per Hour
cancellationPeriod	INT	Cancellation Period



CS251: Phase 1 – Team Name  
Project: <Project Name>

## Software Requirements Specifications

- User Story #3**

<b>User Story ID</b>	US #3
<b>User Story Name</b>	Accept bookings
<b>Actors</b>	Playground owner
<b>Description</b>	Accept the bookings for the players who booked a playground
<b>Per condition</b>	The playground must be available
<b>Post condition</b>	The booking is accepted and the playground is ready to be used
<b>Acceptance Criteria</b>	<p><b>Given:</b> The playground owner wishes to view their booking list and accept booked playgrounds</p> <p><b>When:</b> He/she checks if the playground is available</p> <p><b>Then:</b> The playground booking is accepted</p>

- Scenarios**

### Normal Scenario

Actor Action	System Response
1- Click My Requests	
	2- System displays list of playground bookings 3- Accept/Reject
4- Accept	
	5- "Booking is accepted"

### Exceptional Scenario

Actor Action	System Response
1-Click My Requests	
	2- "No Current Bookings"



# CS251: Phase 1 – Team Name

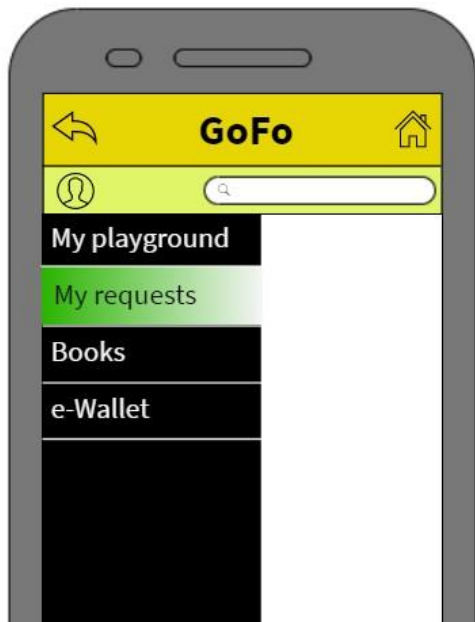
## Project: <Project Name>

### Software Requirements Specifications

- Screen Design

Screen name: My Requests

Screen number: #9



Screen name: Requested bookings

Screen number: #10



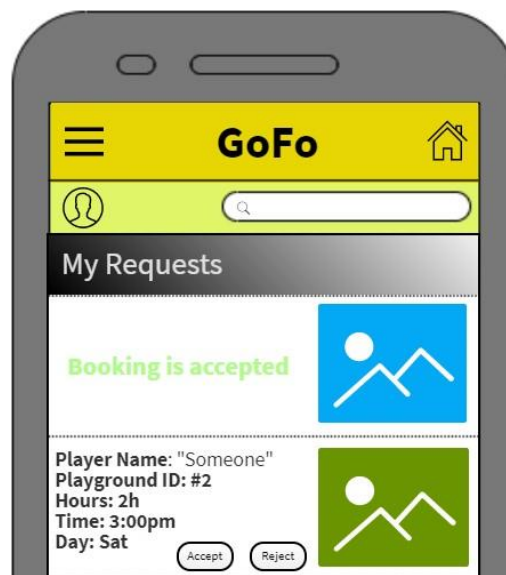
Screen name: Accept

Screen number: #11



Screen name: Booking is accepted

Screen number: #12





# CS251: Phase 1 – Team Name

## Project: <Project Name>

## Software Requirements Specifications

- Data Dictionary:**

Element Label	Type/Length	Data Validation / Business Rule
accept()	BOOL	Accept the player's request
myRequest()	Void	Request of the playground owner

- User Story #4**

<b>User Story ID</b>	US #4
<b>User Story Name</b>	Bookings
<b>Actors</b>	Player
<b>Description</b>	Create a profile then select playground to Book
<b>Per condition</b>	Register his name, id, phone and his location then select the appropriate playground to book it.
<b>Post condition</b>	Paying the amount from his e-wallet to e-wallet of the owner.
<b>Acceptance Criteria</b>	<p><b>Given:</b> the player looks for the nearest and appropriate playground to book it.</p> <p><b>When:</b> He selects one then check if the playground is available or not.</p> <p><b>Then:</b> pay for it from his e-wallet.</p>

- Scenarios**

### Normal Scenario

Actor	System Response
<b>Action</b>	
1- player Signs up	
2- Click <b>Sign Up</b>	
	2- System creates profile for each player
	3- Display all available options.
4- Select a playground	
	5- Checked if this playground available or not
	6-if it's available booked for him.



# CS251: Phase 1 – Team Name

## Project: <Project Name>

### Software Requirements Specifications

#### Exceptional Scenario

Actor Action	System Response
4-Select a playground.	
	5- Checked if this playground available or not. 6- Find this playground already booked.

- Screen Design

**Screen name:** Player Account

**Screen number:** #13



**Screen name:** Playgrounds

**Screen number:** #14



**Screen name:** Displaying Playgrounds

**Screen number:** #15



**Screen name:** Booking a playground

**Screen number:** #16







# CS251: Phase 1 – Team Name

## Project: <Project Name>

## Software Requirements Specifications

- **Data Dictionary:**

Element Label	Type/Length	Data Validation / Business Rule
signUp()	VOID	Signing up for a player new account
playgroundID	INT	Playground that should be booked
availability()	BOOL	Checks if the playground is available

- **User Story #5**

<b>User Story ID</b>	US #5
<b>User Story Name</b>	Cancellation
<b>Actors</b>	Player
<b>Description</b>	The player has a problem with his booking so he needs to cancel.
<b>Per condition</b>	Player login with his account then books a playground.
<b>Post condition</b>	A player has to get back his money and playground back to be available again.
<b>Acceptance Criteria</b>	<b>Given:</b> player want to cancel his reservation. <b>When:</b> He has a problem with his reservation or he doesn't want this playground anymore. <b>Then:</b> he asks to refund his money.

- **Scenarios**

### Normal Scenario

Actor Action	System Response
1- Login with his account 2- Click <b>Login</b>	
	3- The player profile appears.
4- Click <b>My Books</b>	
6- Choose the playground. 7- Click <b>Cancel</b> .	5- System displays list of booked playgrounds
	8- Accept the cancellation.



CS251: Phase 1 – Team Name  
Project: <Project Name>

## Software Requirements Specifications

### Exceptional Scenario

Actor Action	System Response
6- Choose the playground.	
7- Click <b>Cancel</b> .	
	8- Reject the cancellation.

- **Screen Design**

Screen name: My Books

Screen number: #17



Screen name: booked playgrounds

Screen number: #18



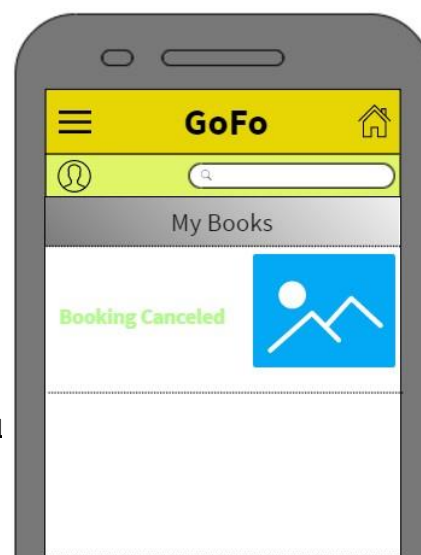
Screen name: Cancel

Screen number: #19



Screen name: Cancel Confirmation

Screen number: #20





# CS251: Phase 1 – Team Name

## Project: <Project Name>

## Software Requirements Specifications

- Data Dictionary:**

Element Label	Type/Length	Data Validation / Business Rule
playerProfile()	CLASS	Player Profile
playgroundID	INT	Playground ID

- User Story #6**

<b>User Story ID</b>	US #6
<b>User Story Name</b>	Sign Up
<b>Actors</b>	Player and Owner
<b>Description</b>	The user uses the system to help him so he needs to create account.
<b>Per condition</b>	Download this application and open it and accept the conditions.
<b>Post condition</b>	He writes down the data required of him.
<b>Acceptance Criteria</b>	<b>Given:</b> open the application and create account. <b>When:</b> download this application. <b>Then:</b> he uses it by his account.

- Scenarios**

### Normal Scenario

Actor Action	System Response
1- Click <b>Sign Up</b>	
	2- System asks the user to write down the required data.
3- Dictate the data.	
4- Click <b>DONE</b> .	
	5- Activate the account to use it every time.

### Exceptional Scenario

Actor Action	System Response
4- Click <b>DONE</b> .	
	6- there is problem, can't activate.



CS251: Phase 1 – Team Name  
Project: <Project Name>

## Software Requirements Specifications

- **Screen Design**

**Screen name:** Sign Up

**Screen number:** #21



**Screen name:** Sign Up Form

**Screen number:** #22



**Screen name:** Done Signing Up

**Screen number:** #23



**Screen name:** User's Profile

**Screen number:** #24





# CS251: Phase 1 – Team Name

## Project: <Project Name>

## Software Requirements Specifications

- Data Dictionary:**

Element Label	Type/Length	Data Validation / Business Rule
signUp()	VOID	Signs up for the user
activate()	VOID	For activating the created account
Player	CLASS	Player profile
PlaygroundOwner()	CLASS	Playground owner profile
email	STRING	Email of the user
name	STRING	Name of the user
lastName	STRING	Last name of the user
password	STRING	Password of the user
location	STRING	Location of the user
phoneNumber	STRING	Phone number of the user

- User Story #7**

<b>User Story ID</b>	US #7
<b>User Story Name</b>	Login
<b>Actors</b>	Player, owner and administrator.
<b>Description</b>	The user needs to login to benefit with its tasks.
<b>Per condition</b>	User has to sign up first.
<b>Post condition</b>	Each user uses this application according to his need.
<b>Acceptance Criteria</b>	<p><b>Given:</b> The user wishes to log in.</p> <p><b>When:</b> He enters his email and password.</p> <p><b>Then:</b> The system displays his profile.</p>

- Scenarios**

### Normal Scenario

Actor Action	System Response
1- Login with his account	
2- Click <b>Login</b>	
	3- System displays the user profile



CS251: Phase 1 – Team Name  
Project: <Project Name>

## Software Requirements Specifications

### Exceptional Scenario

Actor Action	System Response
1- Login with his account 2- Click <b>Login</b>	
	3- System displays an error message 4- “Incorrect password”

- Screen Design

Screen name: Login

Screen number: #25



Screen name: Done Login

Screen number: #27



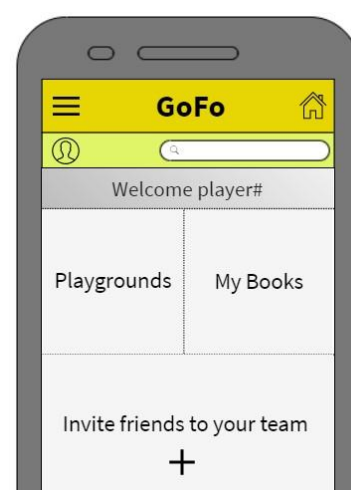
Screen name: Login Form

Screen number: #26



Screen name: User's Profile

Screen number: #28





# CS251: Phase 1 – Team Name

## Project: <Project Name>

## Software Requirements Specifications

- Data Dictionary:**

Element Label	Type/Length	Data Validation / Business Rule
email	STRING	Email of the user
password	STRING	Password of the user
login()	VOID	Login for the user

- User Story #8**

<b>User Story ID</b>	US #8
<b>User Story Name</b>	Invitation
<b>Actors</b>	Player
<b>Description</b>	The player invites his friends and his family to use it.
<b>Per condition</b>	Player selects who is really interested I football from his friends to invite.
<b>Post condition</b>	Friends either accept the invitation or deny it.
<b>Acceptance Criteria</b>	<b>Given:</b> player search for friends to participate him in playing. <b>When:</b> He found them, he sends invitations. <b>Then:</b> they accept the invitation.

- Scenarios**

### Normal Scenario

Actor Action	System Response
1- Click <b>Login</b>	
	2-The player profile appears.
3- search for new players to send for them invitation.	
4- they accept the invitation.	
	5- New player joins the team

### Exceptional Scenario

Actor Action	System Response
3- Send for new players an invitation.	
	4- System can't find these players.



# CS251: Phase 1 – Team Name

## Project: <Project Name>

### Software Requirements Specifications

- Screen Design

**Screen name:** Invite Friends

**Screen number:** #29



**Screen name:** Invitation Form

**Screen number:** #30



**Screen name:** Send Invitation

**Screen number:** #31



**Screen name:** Invitation Sent

**Screen number:** #32







# CS251: Phase 1 – Team Name

## Project: <Project Name>

## Software Requirements Specifications

- Data Dictionary:**

Element Label	Type/Length	Data Validation / Business Rule
login()	VOID	Login for any user to his account
newPlayer()	VOID	New player joins the team
teamID	INT	Team ID
playgroundID	ING	Playground ID
newPlayerID	INT	The player that the invitation will be sent to

- User Story #9**

<b>User Story ID</b>	US #9
<b>User Story Name</b>	Check Money
<b>Actors</b>	Playground owner
<b>Description</b>	The owner needs to check if the money has been already successfully transferred.
<b>Per condition</b>	Player has to transfer the money from his e-wallet to owner e-wallet.
<b>Post condition</b>	Confirm the reservation.
<b>Acceptance Criteria</b>	<p><b>Given:</b> Owner needs to check his money.</p> <p><b>When:</b> Opens his e-wallet.</p> <p><b>Then:</b> He checks if the money has been successfully transferred.</p>

- Scenarios**

### Normal Scenario

	System Response
1- Click <b>Login</b>	
	2- System Logs in with owner profile
3- Owner opens his e-wallet	
	4- System displays a list of money transactions



CS251: Phase 1 – Team Name  
Project: <Project Name>

## Software Requirements Specifications

### Exceptional Scenario

Actor Action	System Response
1- Click <b>Login</b> .	
	2- "Incorrect Password"

- Screen Design

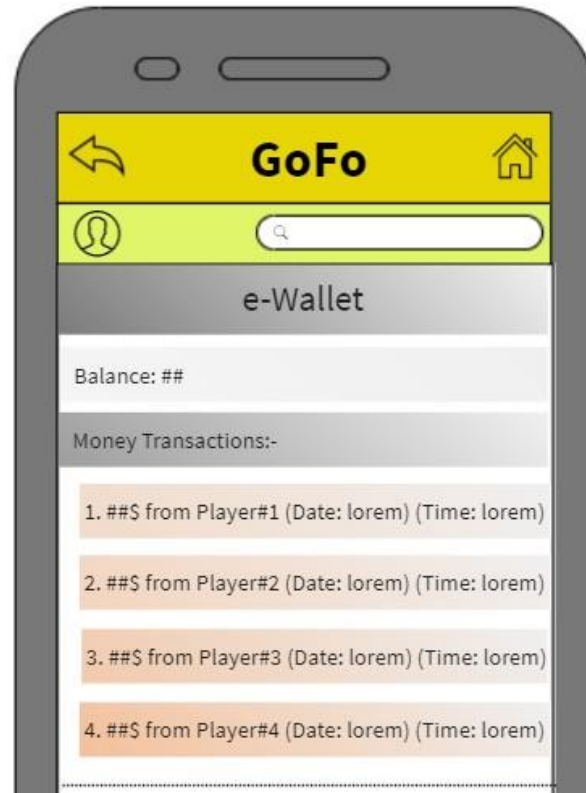
Screen name: e-Wallet

Screen number: #33



Screen name: Money Transactions

Screen number: #34



- Data Dictionary:

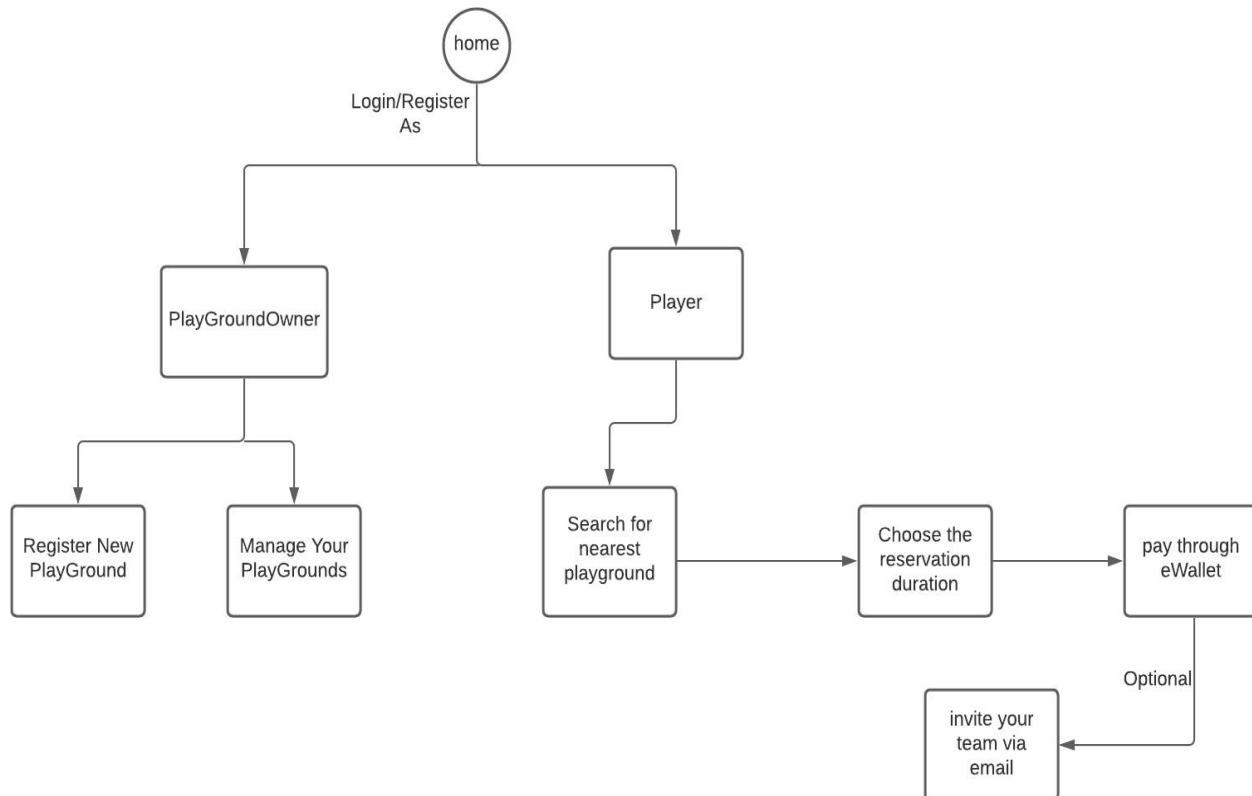
Element Label	Type/Length	Data Validation / Business Rule
login()	VOID	Login for user
e-wallet() getBalance	VOID INT	Displaying list of money transactions Balance of the owner's account



CS251: Phase 1 – Team Name  
Project: <Project Name>

## Software Requirements Specifications

### System Navigation Map



## Tools

Lucid App, Mockflow (Wireframe)

## Ownership Report

Student	Items he created
Ahmed Magdy Abd-El-fattah (Team Leader)	(Software Purpose) + (Definitions, acronyms, and abbreviations) + (Functional Requirements) + (User Stories {2, 3}) + (System Navigation Map)
Mostafa Nassr El-deen El-sayed	(Software Scope) + (Non-Functional Requirements) + (User story {1}) + {Screen designs for all user stories + Data Dictionaries}
Manar Emad El-din Hassan	(Document purpose and audience) + (Use Case Model) + (User Stories {4, 5, 6, 7, 8, 9})



CS251: Phase 1 – Team Name

Project: <Project Name>

## Software Requirements Specifications