Cairo University
Faculty of Computers and Artificial Intelligence

CS251

Intro. to Software Engineering

GoFo - “Go Football”

Software Requirements Specifications

Version 1.0

Youssef Hesham Mohamed

Marwan Mohamed Abd-El-moneam

Mohamed Ashraf Hassan

Month & Year

Contents

Team

ID	Name	Email	Mobile
20190648	Youssef Hesham Mohamed	Elhaddad751@gmail.com	01012059214
20190513	Marwan Mohamed Abdelmoneam	Marwanko666@gmail.com	01555388428
20190421	Mohamed Ashraf Hassan metwally	Mohamed22ashraf21@gmail.com	01027207801

Document Purpose and Audience

- The purpose of this document is to clarify the working of this project "Go football" and how it works
- We 'll list the important key figures in the project to ensure that the client will work perfectly
- This document is intended for product owner, clients (playground owners, players)

Introduction

Software Purpose

- To facilitate the booking system for football playgrounds. It serves the players and playground owners as well

Software Scope

- The scope of this application is to make a system for booking and connecting the playground owners with the players and to help individual players find a team to play in
- This application doesn't support payment process so it's done externally through eWallet
- It requires online connection to validate any process done within the system
- It acts as a marketing platform for playground owners and reduce the cost of finding a suitable playground for players and helps players in knowing new individuals

Definitions, acronyms, and abbreviations

Playground owner	The one who will rent the field
eWallet	A software that stores your data of money to allow you to pay online
Player	The one who wants to book the field

Requirements

Functional Requirements

- A sign-up tab is displayed (if the user doesn't have an account) whether to add an account as a playground owner or as a player and it asks to enter his name, ID and password, email, phone and default location
- A sign in tab if is the user already has an account
- Playground owner window has three tabs:

  - "My playground" which allows the owner to add (or modify) a number of playgrounds, their location, availability, price per hour, cancellation period, photo of the field, type of playground and its capacity

  - "My requests" allow the owner to review the incoming requests from players and its details and whether to accept or reject them, owner can't reject a request after accepting it

  - "Books" allows the owner to preview the list of his playgrounds where which one is booked and which is not, the available hours of each one

  -"eWallet" allows the user to make transactions to/from the system then to/from the end users and shows the current money

- Player window has four tabs:
  - "Book playground" allows the user to preview all available playgrounds for booking and he can search by name or filtering by location, pricing, desired hours, types of playground or its capacity and to book one or many and selecting the number of players

  - "My requests" allows the user to preview his requests where after booking it stores the booking information as a pending request waiting for the owner response, if the owner accept it, it changes from pending to accepted or to rejected if owner reject it, the user can cancel his request if its still in the cancellation period

  - "Join team" allows the user to join a team which already booked a playground where it previews the incomplete teams and after filling max individuals it will delete this object

  "My Books" allows the user to preview the accepted requests and the time to end it

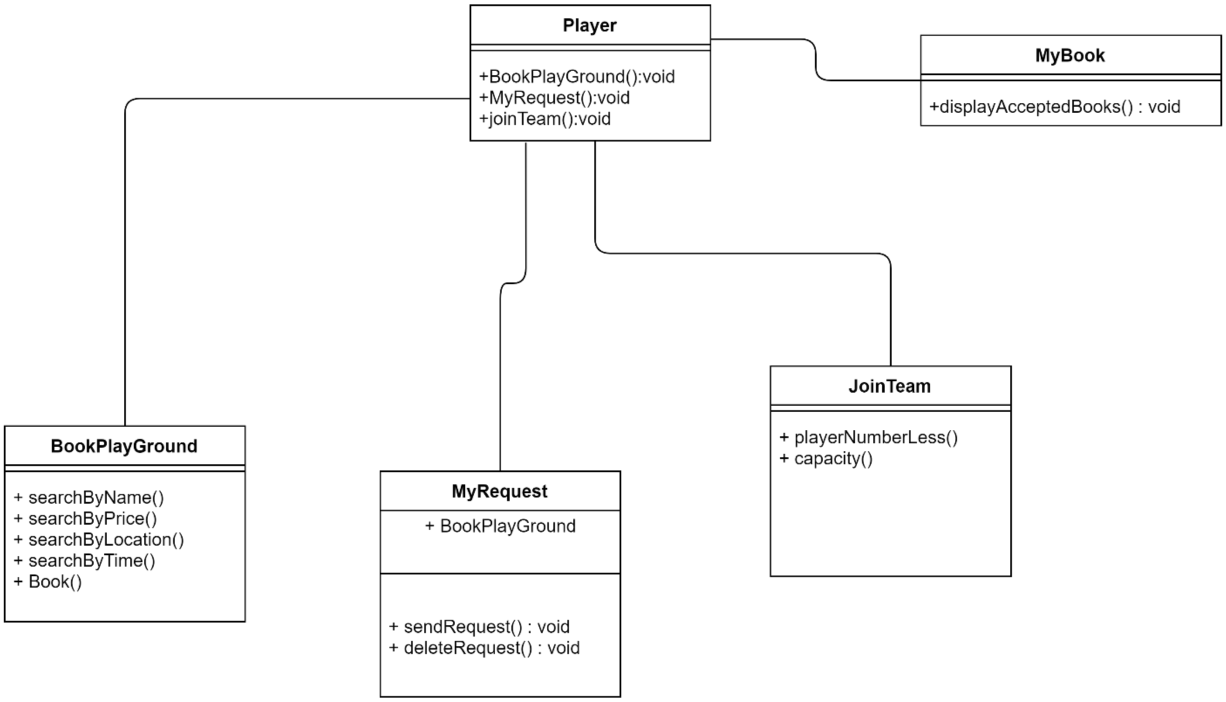
  -"eWallet" allows the user to make transactions to/from the system then to/from the end users and shows the current money

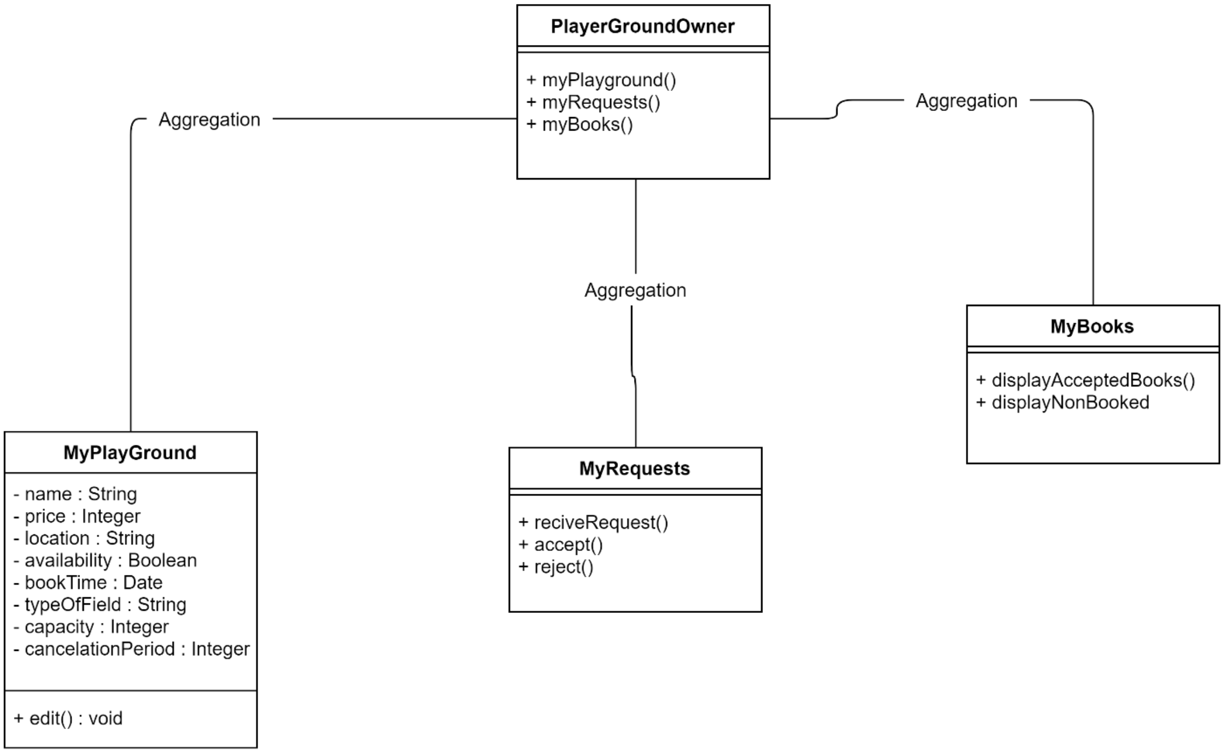
Non Functional Requirements

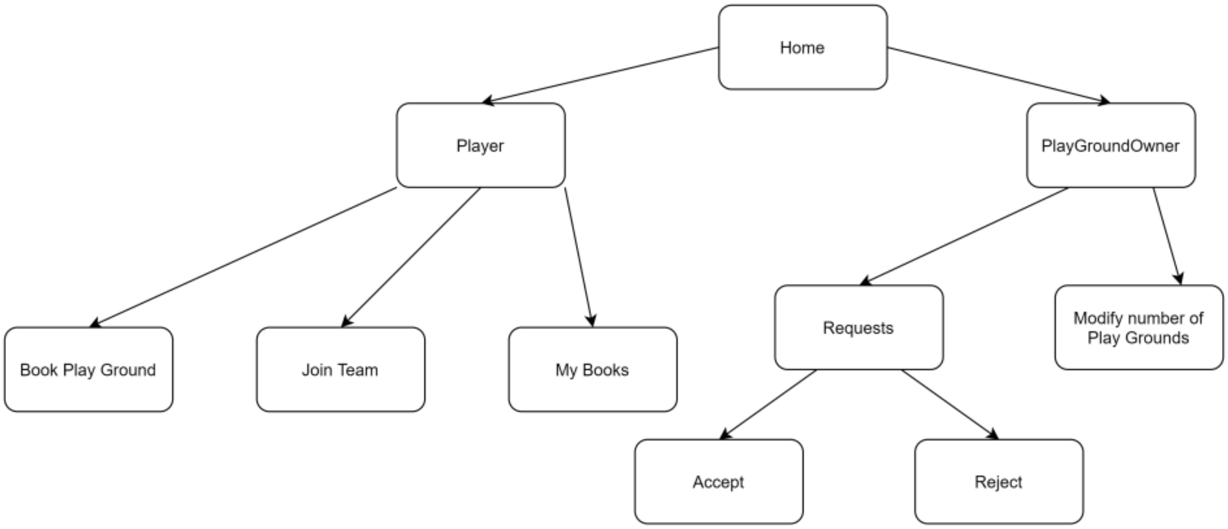
	Details
Performance	- **Booking operation will be done in 3 sec**
Scalability	- **System should be able to support up to 50 simultaneous booking operation.**
Quality	- **Administrator should remove any false info as soon as its discovered**
Privacy	- **Payment transferee is guaranteed to be discretion where no other player or owner can have these info**
Maintainability	- **Frequently updated patches and removing any discoverable bugs**
Data integrity	- **Maintaining true information about the users and the playgrounds and its pricing**

System Models

Use Case Model


Player
+BookPlayGround():void
+MyRequest():void
+joinTeam():void
MyBook
+displayAcceptedBooks() : void
JoinTeam
+ playerNumberLess()
+ capacity()
BookPlayGround
+ searchByName()
+ searchByPrice()
+ searchByLocation()
+ searchByTime()
+ Book()
MyRequest
+ BookPlayGround
+ sendRequest() : void
+ deleteRequest() : void


PlayerGroundOwner
+ myPlayground()
+ myRequests()
+ myBooks()
Aggregation
Aggregation
Aggregation
MyPlayGround
- name : String
- price : Integer
- location : String
- availability : Boolean
- bookTime : Date
- typeOfField : String
- capacity : Integer
- cancelationPeriod : Integer
+ edit() : void
MyRequests
+ reciveRequest()
+ accept()
+ reject()
MyBooks
+ displayAcceptedBooks()
+ displayNonBooked


Home
Player
PlayGroundOwner
Book Play Ground
Join Team
My Books
Requests
Modify number of Play Grounds
Accept
Reject

Enriched User Stories

- **Using below table template, for each requirement write an enriched user story specifying the details of each use case and showing the interaction to implement this use case.**
    - **If one requirement is so big, you could divide it to more than one user story.**
    - **If some requirements are not major, you could plugin them in other user stories.**
- **Flow of events should be very detailed**

- ### User Story #1

User Story ID	US #1
User Story Name	Sign up
Actors	**Playground owner/player**
Description	**A window for creating accounts**
Per condition	Doesn't have an account / wants to new account
Post condition	
Acceptance Criteria	**Given** I am a newcomer to the system user or I want to add a new account. **When** I fill in the “Username” and “Password” fields with my authentication credentials and I click the Sign-up button **Then** the system signs me up and save my data on the system database

- ### Scenarios

Normal Scenario

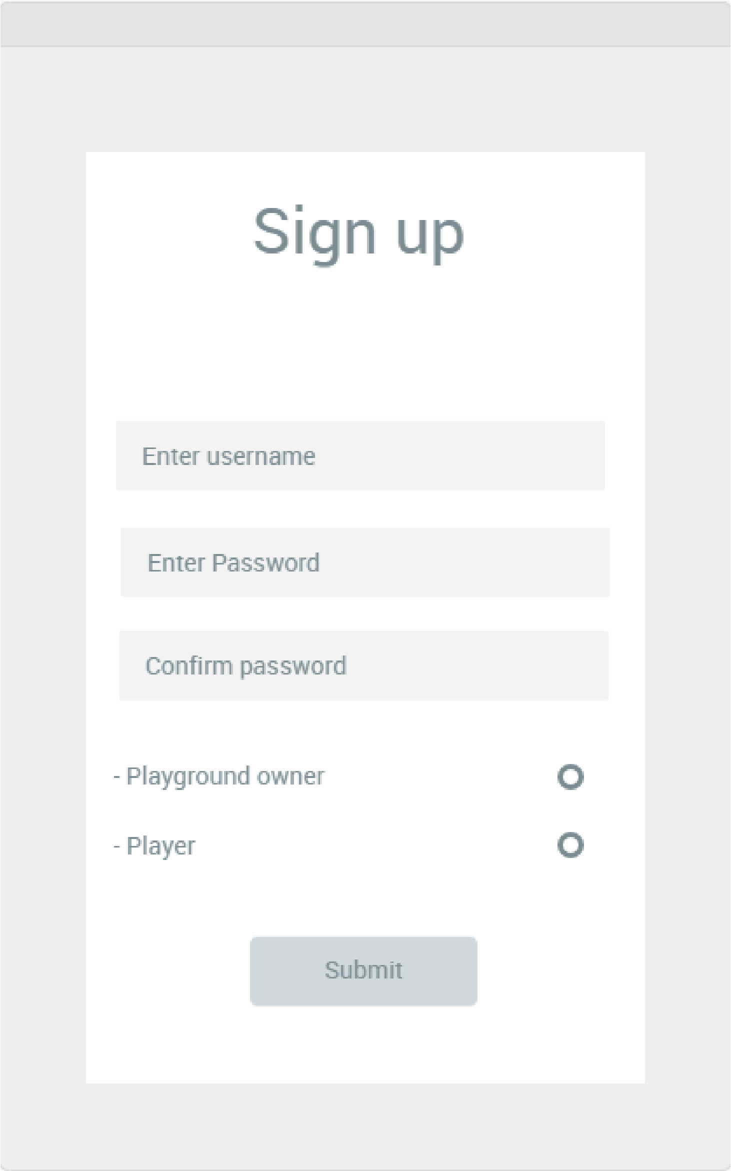
Actor Action	System Response
1- User Enter username and Password. 2- Click **Submit**	
	3- System saves user data 4- System displays to continue as an owner or a player
5- User Select one of them	
	6- System displays another window

Exceptional Scenario

Actor Action	System Response

1- User Enter Card and Password. 2- Click **Submit**	
	3- username or password are already taken 4- Systems rejects sign up process and displays an error message

- **Screen Design**



- **Data Dictionary:**

Element Label	Type/Length	Data Validation / Business Rule
Username	array	It accepts any data type
Password	array	It accepts any data type

____
___

- ## User Story #2

User Story ID	US #2
User Story Name	Sign in
Actors	**Playground owner/player**
Description	**As** a user. **I like** to be able to sign in **So it allows the user to access the application**
Per condition	User are saved in the system database
Post condition	
Acceptance Criteria	**Given** I'm a logged-out system user and I'm on the Sign-In page **When** I fill in the "Username" and "Password" fields with my authentication credentials and I click the Sign-In button **Then** the system signs me in

- ## Scenarios

Normal Scenario

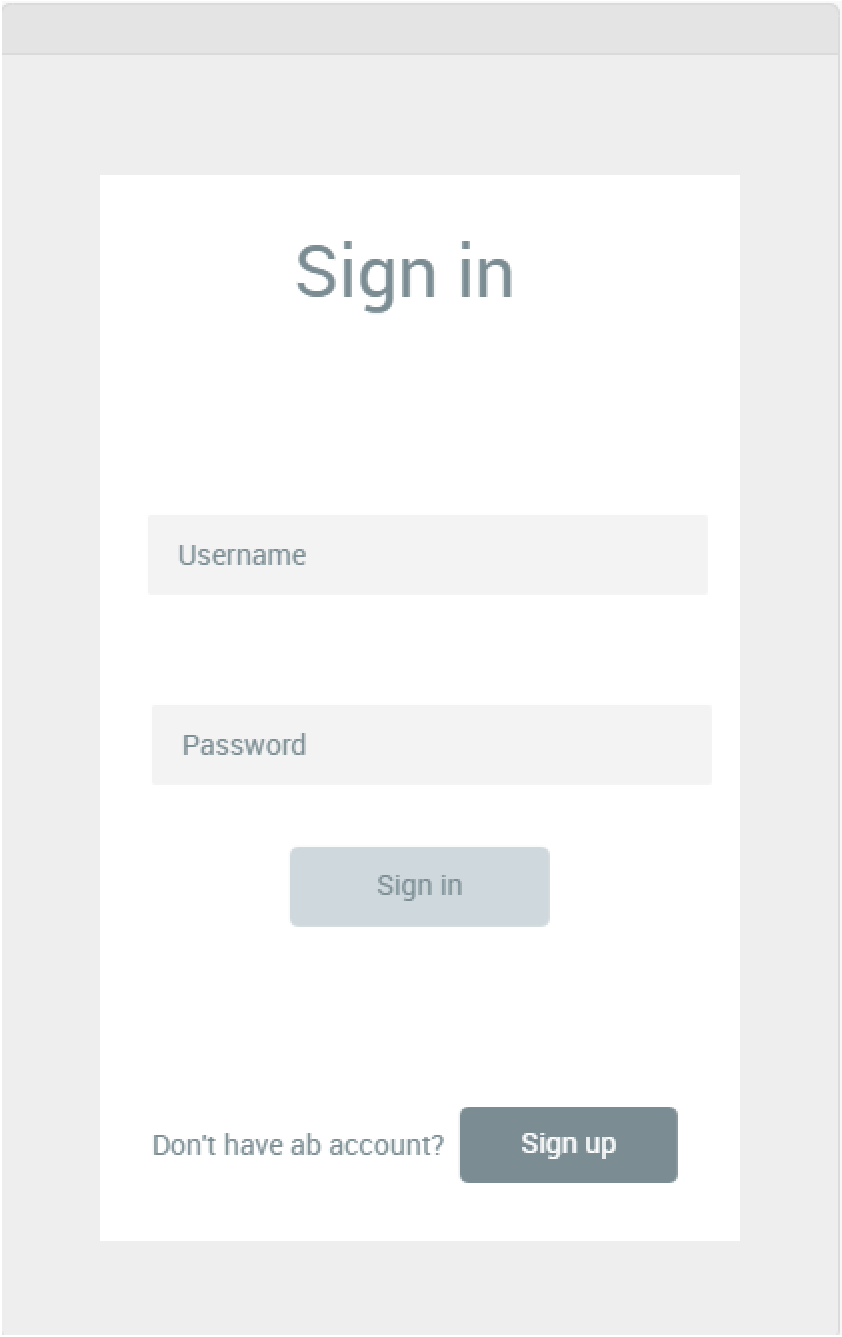
Actor Action	System Response
1- User Enter username and Password. 2- Click **Submit**	
	3- System verify user's data 4- system display main window

Exceptional Scenario

Actor Action	System Response
1- User Enter Card and Password. 2- Click **Submit**	
	3-either username or password is invalid 4- Systems rejects sign up process and displays an error message

- **Screen Design**

- **Data Dictionary:**



Element Label	Type/Length	Data Validation / Business Rule
Username	array	It accepts any data type
Password	array	It accepts any data type

____

___

- ## User Story #3

User Story ID	US #3
User Story Name	My playground
Actors	**Playground owner**
Description	**As** a user. **I like** to be able to add/modify playground. **So it allows the user to** add/modify playground
Per condition	
Post condition	
Acceptance Criteria	**Given** I'm a playground owner and in the main menu **When** I select My playground, it allows me to add / modify my playgrounds **Then** the system adds/ modify the data

- ## Scenarios

- ## Normal Scenario

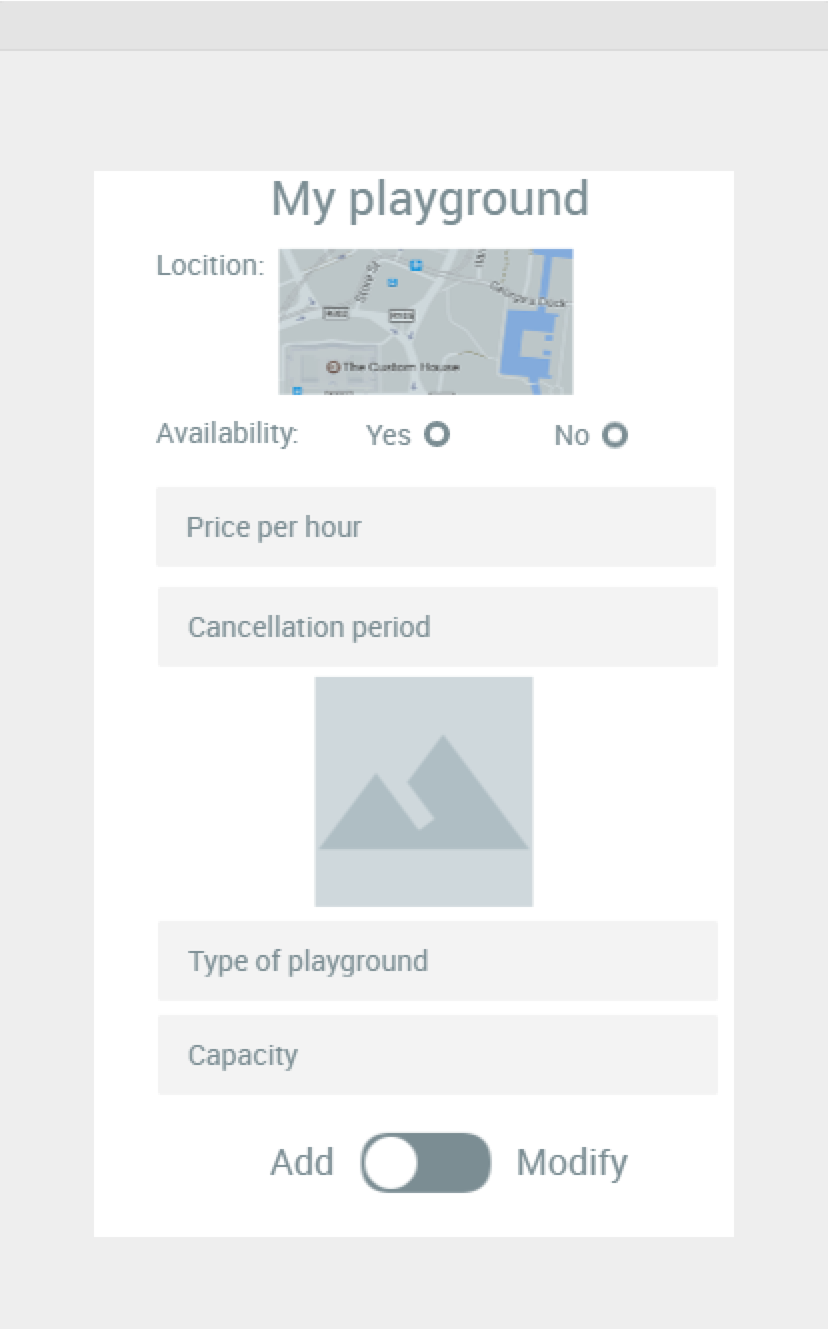
Actor Action	System Response
1- User selects my playground	
	2- System display my playground window
3-User select add/modify 4- User add/modify location, availability, price per hour, cancellation period, photo of the field, type of playground and its capacity	
	4-system adds /modify playground and its info if administrator accept

- ## Exceptional Scenario

Actor Action	System Response
1- User selects my request.	
	2- System display my playground window
3-User select add/modify	

4- User add/modify location, availability, price per hour, cancellation period, photo of the field, type of playground and its capacity	
	4-system display error message if administrator reject

- **Screen Design**



- **Data Dictionary:**

Element Label	Type/Length	Data Validation / Business Rule
My playground	array	It accepts any data type , its divided into categories

Element Label	Type/Length	Data Validation / Business Rule

____
___

- ## User Story #4

User Story ID	US #4
User Story Name	My requests
Actors	**Playground owner**
Description	**As** a user. **I like** to be able to review the incoming requests from players **So it allows the user to** accept or reject requests
Per condition	A request is sent from player or it will be empty
Post condition	The processed request is deleted, and price of booking is transferred automated
Acceptance Criteria	**Given** I'm a playground owner and in the main menu **When** I select My requests, it allows me **to** accept or reject requests. **Then** the system save and send the reply to the player and automated transferring money process

- ## Scenarios
- ## Normal Scenario

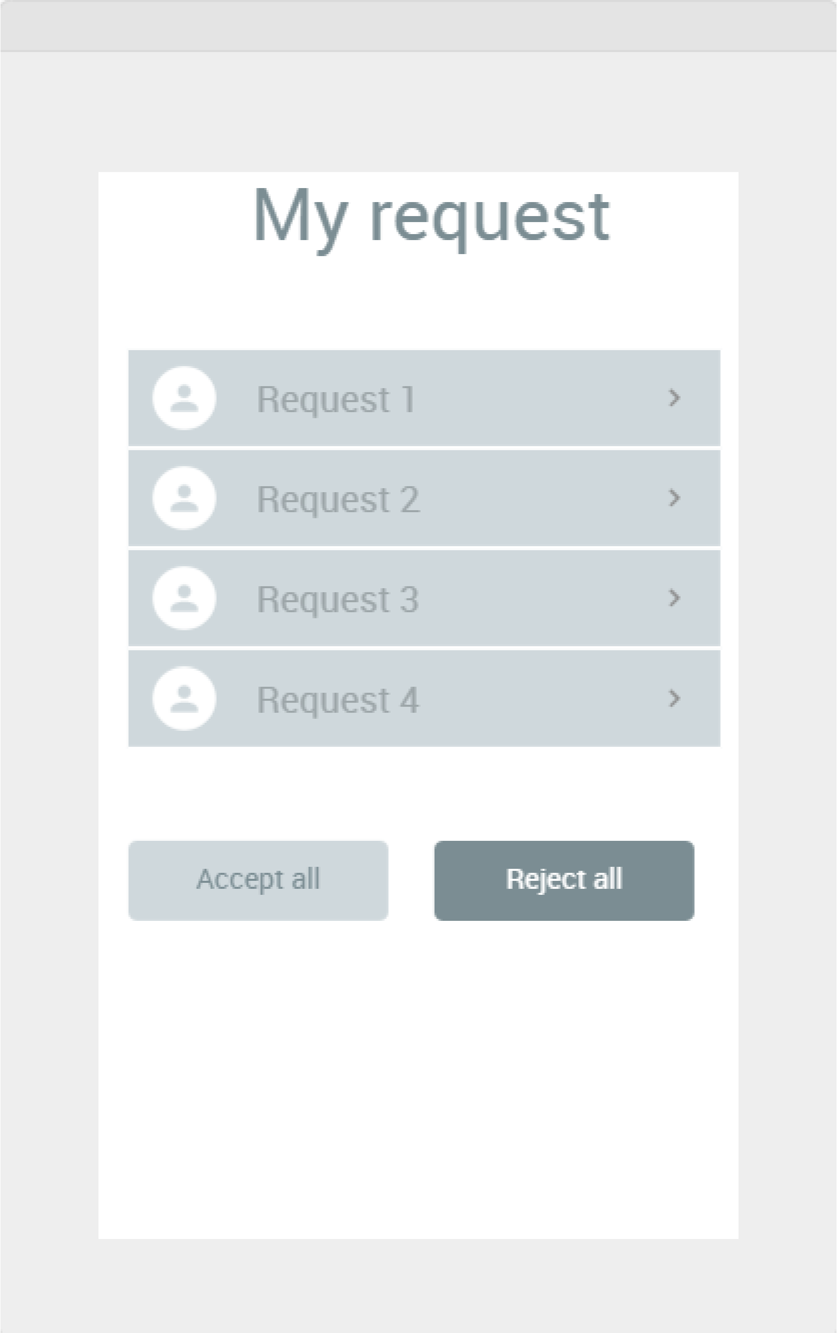
Actor Action	System Response
1- User selects my request.	
	2- System display my request window
3-User choose accept/reject request	
	4-system save the reply 5- system send the reply to player account who send the request 6- if User accept System deduct the price value from player's eWallet account 7- system deletes the processed request
8- And so on	

- ## Exceptional Scenario

Actor Action	System Response
1- User selects my request.	
	2- System display my request window

3-User choose accept/reject request	
	4-system cant accept if balance of the requested player is less than the price

- **Screen Design**



- **Data Dictionary:**

Element Label	Type/Length	Data Validation / Business Rule
My request	array	Saves the selected playground and send it to the owner
My Request	array	Takes My request array and present it

Element Label	Type/Length	Data Validation / Business Rule
choice	bool	Decides if accepts or reject

- ## User Story #5

User Story ID	US #5
User Story Name	My books
Actors	**Playground owner**
Description	**As** a user. **I like** to be able to preview the detailed info about all playground and whether its booked or not. **So it allows the user** to preview playground
Per condition	At least one playground exists
Post condition	
Acceptance Criteria	**Given** I'm a playground owner and in the main menu **When** I select My books **it allows me** to preview playground **Then** the system displays it

- ## Scenarios

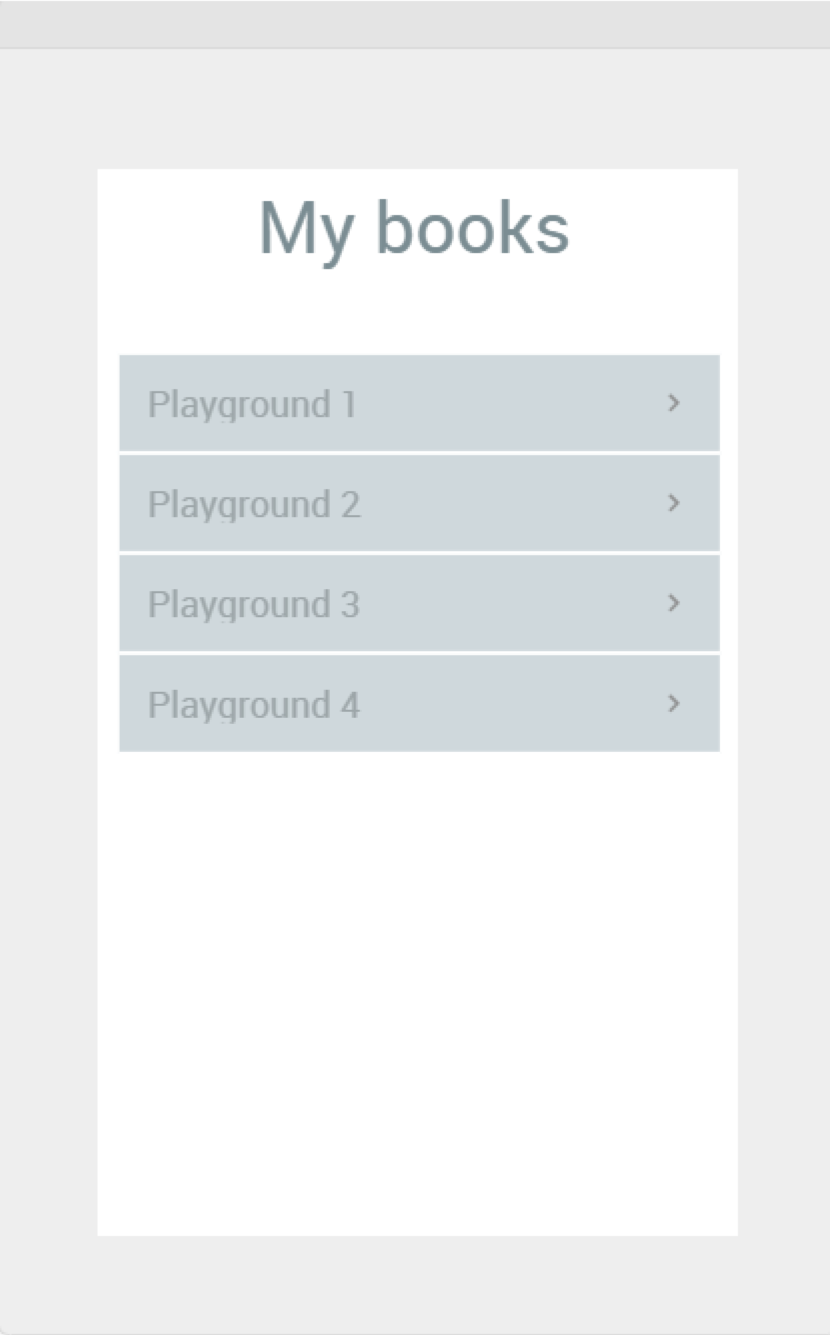
- ### Normal Scenario

Actor Action	System Response
1- User selects my books	
	2- System display my books window 3- System display the current details about all the owner playgrounds

- ### Exceptional Scenario

Actor Action	System Response
1- User selects my books	
	2- System display my books window 3- System display no playgrounds if no playgrounds exist

- **Screen Design**



My books

Playground 1 ›

Playground 2 ›

Playground 3 ›

Playground 4 ›

- **Data Dictionary:**

Element Label	Type/Length	Data Validation / Business Rule
My playground	array	It accepts any data type , its divided into categories

____
___

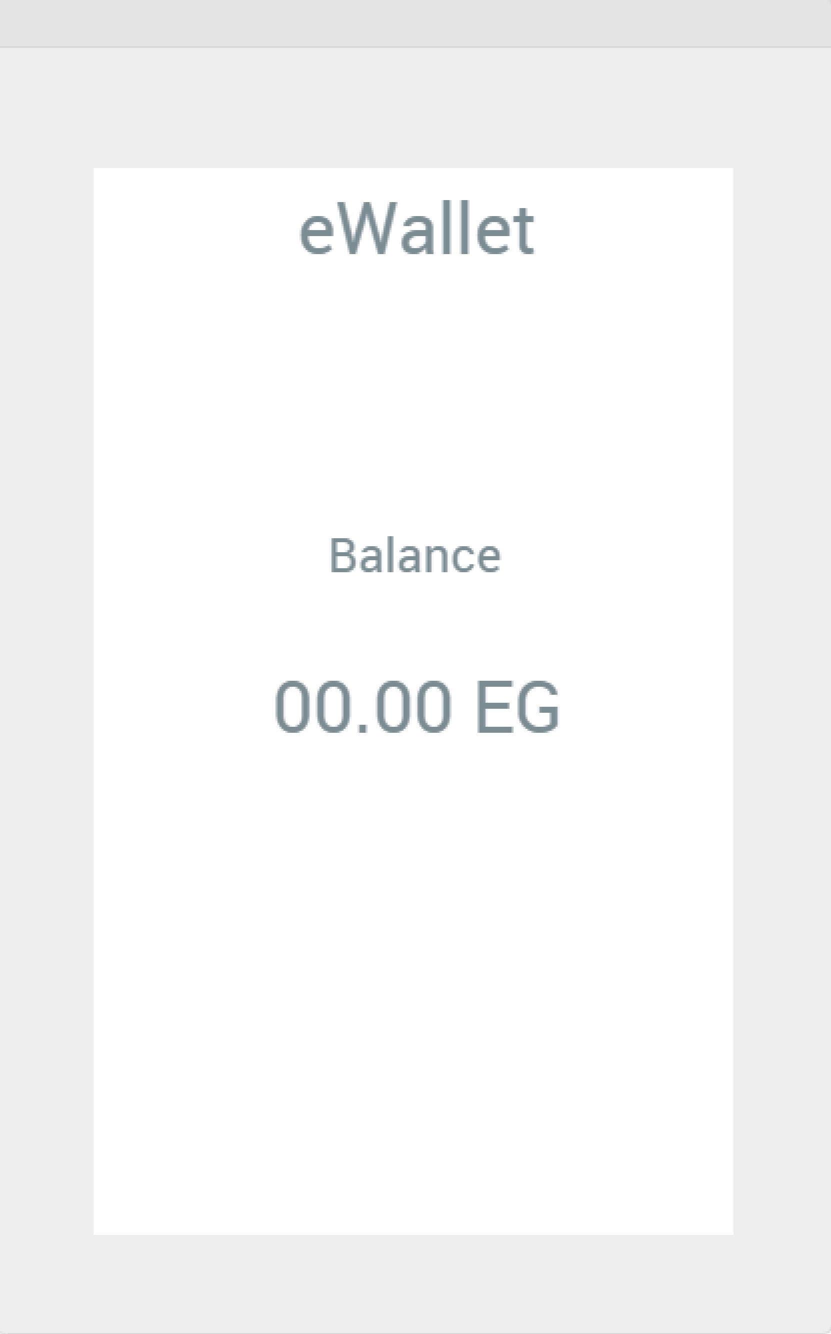
- ## User Story #6

User Story ID	US #6
User Story Name	eWallet
Actors	**Playground owner/player**
Description	**As** a user. **I like** to be able to preview my current money / pay playground owners for booking **So it allows the user** to preview current money
Per condition	At least one playground exists , a request must be processed
Post condition	
Acceptance Criteria	**Given** I'm a user and in the main menu **When** I select eWallet **it allows me** to preview current money **Then** the system process it

- ## Scenarios
- ## Normal Scenario

Actor Action	System Response
1- User selects eWallet	
	2- System display eWallet window 3- System display the current balance

- ## Screen Design



eWallet

Balance

00.00 EG

- **Data Dictionary:**

Element Label	Type/Length	Data Validation / Business Rule
balance	double	Positive values only

____

__

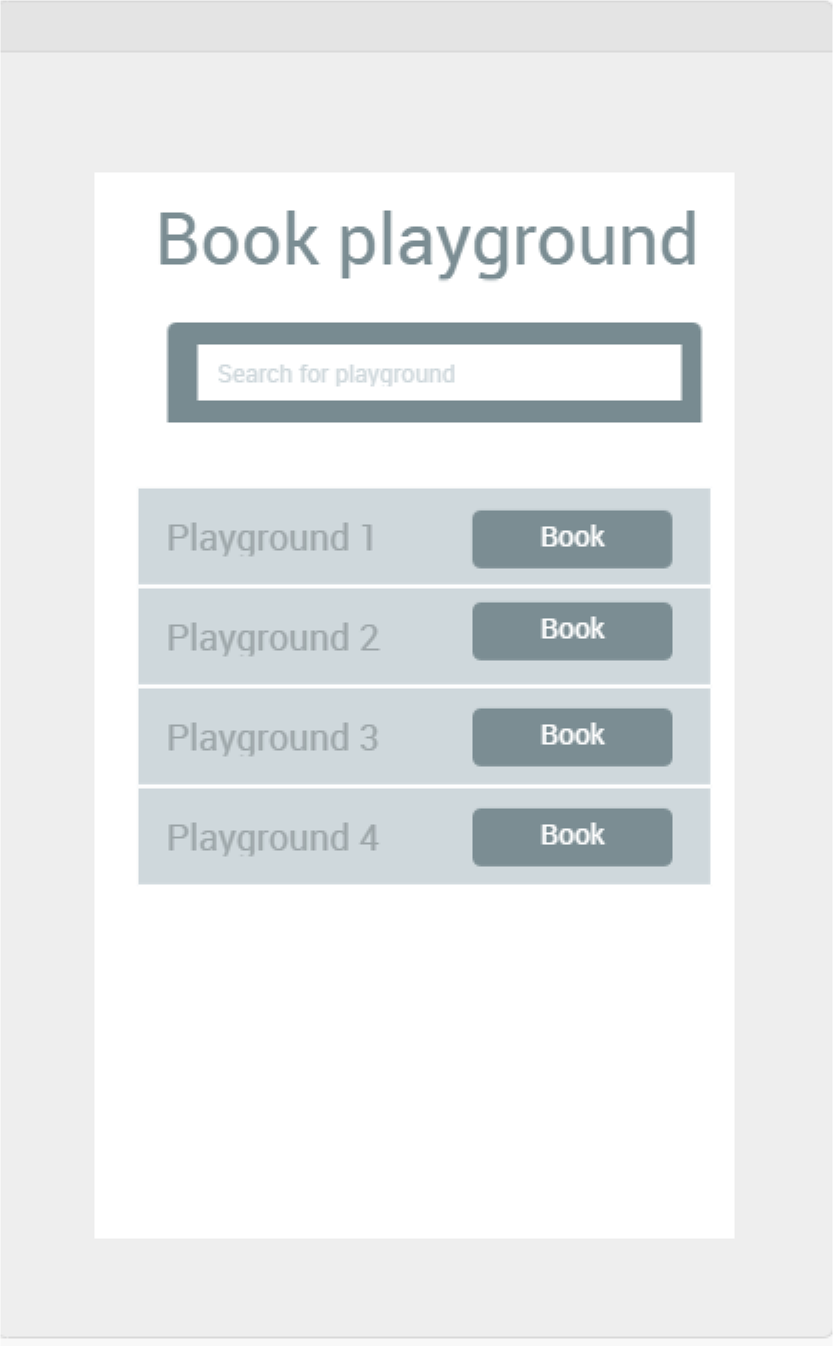
- ## User Story #7

User Story ID	US #7
User Story Name	Book playground
Actors	**Player**
Description	**As** a user. **I like** to be able to preview available playgrounds for booking **So it allows the user** to book playground
Per condition	At least one playground exists, a balance is equivalent or more than the price of the booking
Post condition	
Acceptance Criteria	**Given** I'm a user and in the main menu **When** I select Book playground **it allows me** to preview available playgrounds for booking **Then** the system reserve it and change this playground from available to booked and delete it from this list

- ## Scenarios
- ## Normal Scenario

Actor Action	System Response
1- User selects book playground	
	2- System display playground list
3-User selects suitable playground	
	4- System send request to playground owner 4- System links this playground for this user until book ends 5- System hide this playground until book ends

- ## Screen Design



- **Data Dictionary:**

Element Label	Type/Length	Data Validation / Business Rule
My playground	array	It accepts any data type , its divided into categories
My choice	bool	Whether its booked or not

Element Label	Type/Length	Data Validation / Business Rule

- ## User Story #8

User Story ID	US #8
User Story Name	My request
Actors	**Player**
Description	**As** a user. **I like** to be able to preview the detailed info about all my requests **So it allows the user** to preview requests and be updated
Per condition	At least one playground exists , at least account made one request
Post condition	Price value is deducted if playground owner accepts
Acceptance Criteria	**Given** I'm a player and in the main menu **When** I select My request **it allows me** to preview sent request **Then** the system displays it as pending

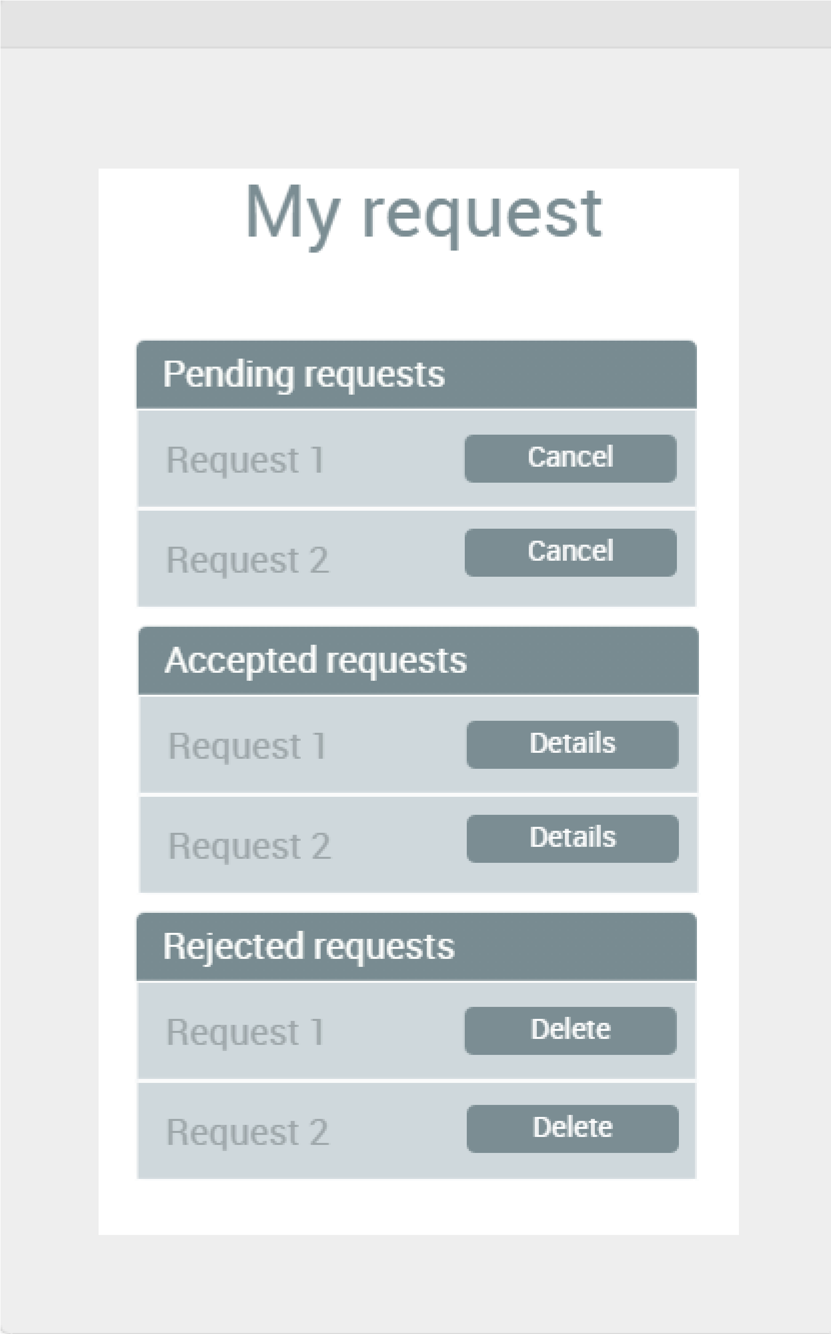
- ## Scenarios
- ## Normal Scenario

Actor Action	System Response
1- User selects my requests	
	2- System display my requests window 3- System display the current requests as pending 4- System changes pending request to accepted/rejected if owner accepts/rejects 5- System deletes rejected request

- ## Exceptional Scenario

Actor Action	System Response
1- User selects my requests	
	2- System display no requests have been made if user didn't book a playground

- **Screen Design**



- **Data Dictionary:**

Element Label	Type/Length	Data Validation / Business Rule
My playground	array	It accepts any data type , its divided into categories

Element Label	Type/Length	Data Validation / Business Rule
My request	array	Saves the selected playground and send it to the owner

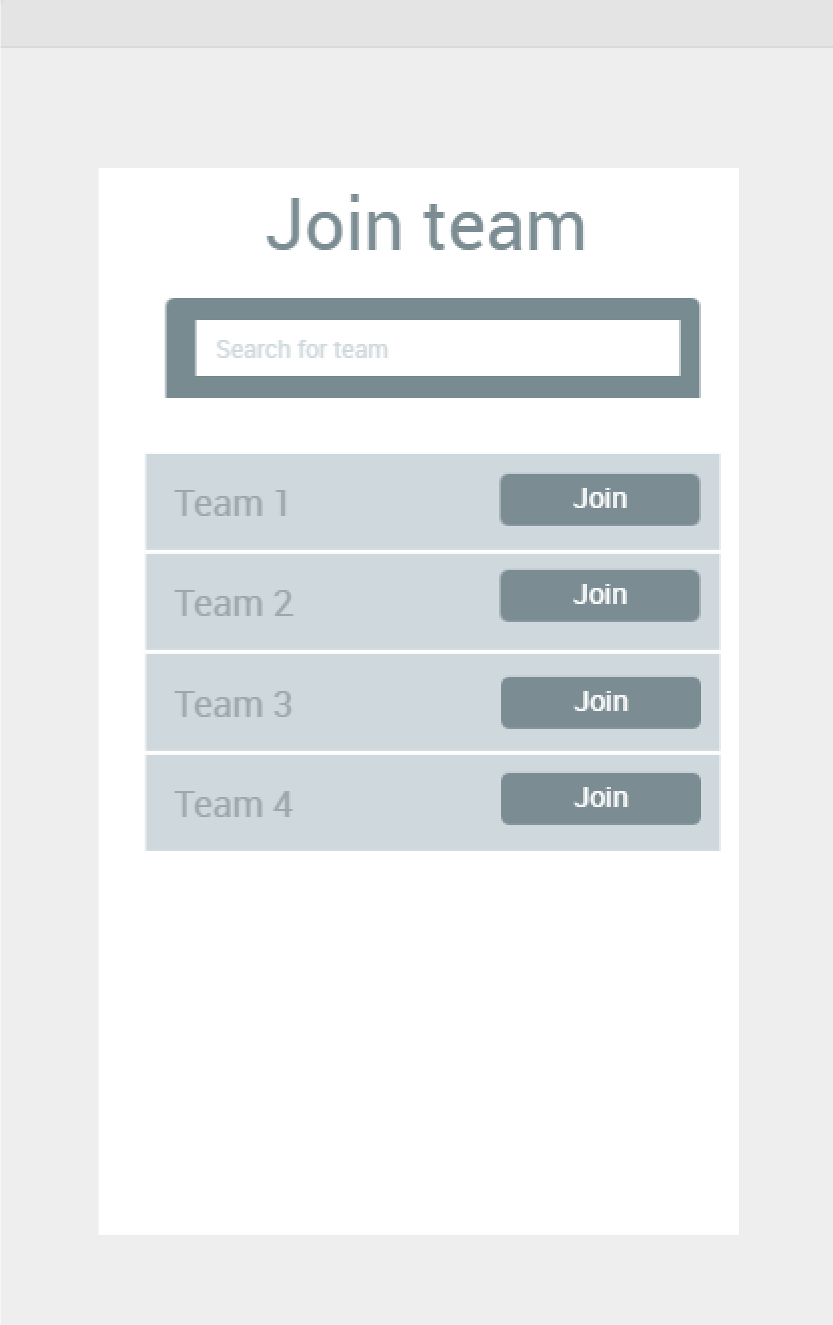
- ## User Story #9

User Story ID	US #9
User Story Name	Join team
Actors	**Player**
Description	**As** a user. **I like** to be able to preview available teams to join **So it allows the user** to join incomplete teams
Per condition	At least one playground exists, a balance is equivalent or more than the price of the booking, booking team is less than the capacity of the playground
Post condition	If joined team filled delete it from this tab
Acceptance Criteria	**Given** I'm a user and in the main menu **When** I select team, **it allows me** to preview available teams for joining **Then** the system adds this account to the team

- ## Scenarios
- ## Normal Scenario

Actor Action	System Response
1- User selects join team	
	2- System display teams list
3-User selects suitable team	
	4- System send request to playground owner 4- System links this player account with booked playground until book ends 5- System deletes this team if maximum reached

- ## Screen Design



- **Data Dictionary:**

Element Label	Type/Length	Data Validation / Business Rule
My team	Array / length of capacity of playground	If length reached max delete this variable

Element Label	Type/Length	Data Validation / Business Rule

____
___

User Story #10

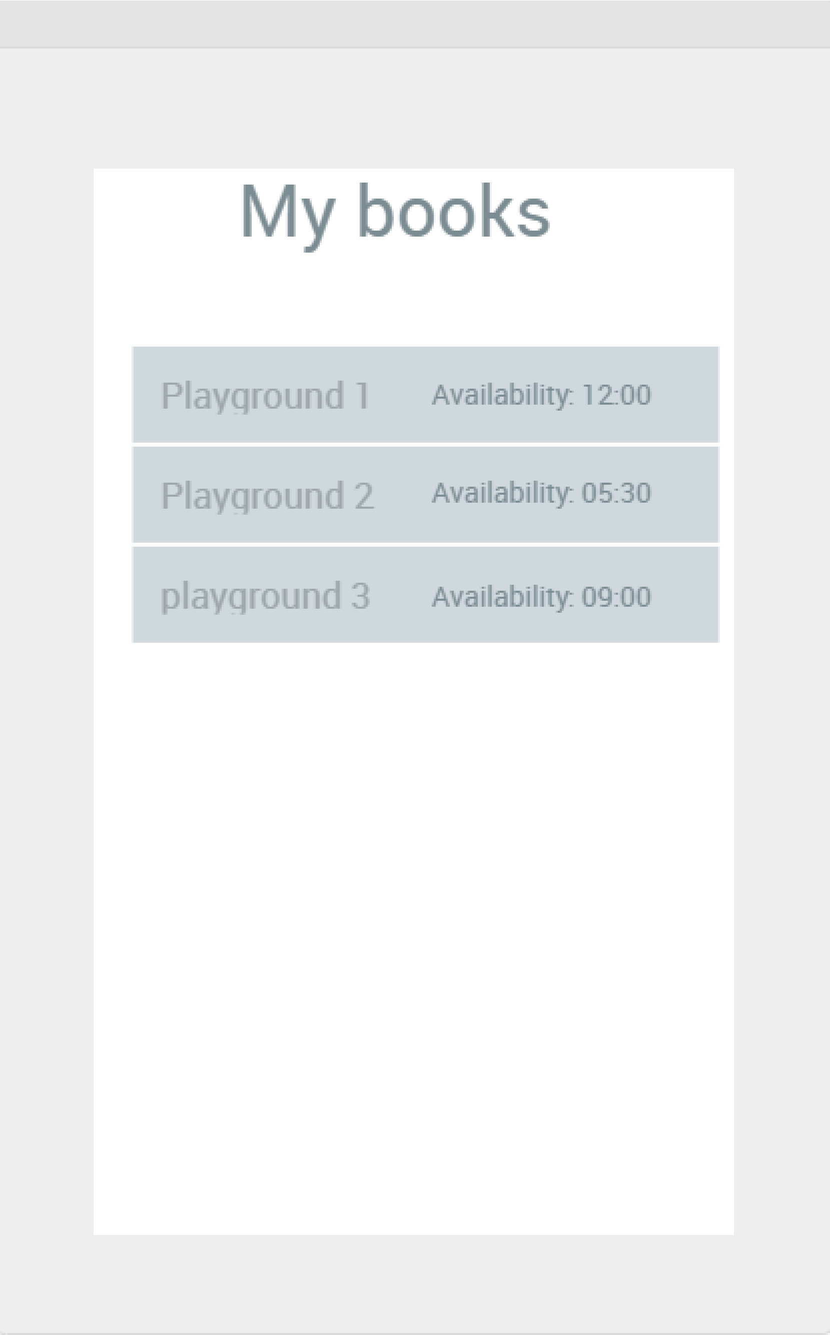
User Story ID	US #10
User Story Name	My books
Actors	**Player**
Description	**As** a user. **I like** to be able to preview my accepted books **So it allows the user** to preview the details of accepted books
Per condition	At least one playground exists, at least one request is accepted
Post condition	
Acceptance Criteria	**Given** I'm a user and in the main menu **When** I select My books, **it allows** preview my accepted books **Then** the system displays the list

Scenarios

Normal Scenario

Actor Action	System Response
1- User selects My books	
	2- System display books list

Screen Design



My books

Playground 1	Availability: 12:00
Playground 2	Availability: 05:30
playground 3	Availability: 09:00

- **Data Dictionary:**

Element Label	Type/Length	Data Validation / Business Rule
My books	array	Hold “My request” and “choice” if accepted
My Request	array	Takes My request array and present it
choice	bool	Decides if accepts or reject

Element Label	Type/Length	Data Validation / Business Rule

____
___

Ownership Report

Student	Items he created
Youssef Hesham Mohamed Ibrahim	Functional requirement, user story (1 to 5), use case
Marwan Mohamed Abd Elmonem	*Non functional requirements, user story (6 to 10)*
Mohamed Ashraf Hassan	*Introduction, system model , screen designs*