

**Cairo University**

**Faculty of Computers and Artificial Intelligence**

**CS251**

**Software Engineering I**

**GOFO**

**Software Design Specifications**

**Version 1.0**

**May,2021**



CS251: Phase 2 – **Creato Software Solution**  
Project: **GOFO**  
**Software Design Specification**

**Contents**

Team ..... 3

Document Purpose and Audience ..... 3

Purpose..... 3

Audience ..... 3

System Models ..... 4

I. Class Diagram(s) ..... 4

II. Class Descriptions..... 5

III. Sequence diagrams ..... 8

Class - Sequence Usage Table ..... 12

State Diagram..... 13

Tools ..... 13

Ownership Report..... 13



CS251: Phase 2 – Creado Software Solution  
Project: **GOFO**  
Software Design Specification

Team

ID	Name	Email	Mobile
20190586	Nahla Husain Ali	<a href="mailto:nahlahusain07@gmail.com">nahlahusain07@gmail.com</a>	01151477745
20190159	Gowhara eid abdelghany	<a href="mailto:gowharaeid@gmail.com">gowharaeid@gmail.com</a>	01095664988
20190519	Maryam Sayed Sadeek	<a href="mailto:20190519@stud.fci-cu.edu.eg">20190519@stud.fci-cu.edu.eg</a>	01066501403

Document Purpose and Audience

Purpose

This document is software design specification for **GOFO** application containing class diagrams,class description and their responsibilities, and sequence diagram.

Audience

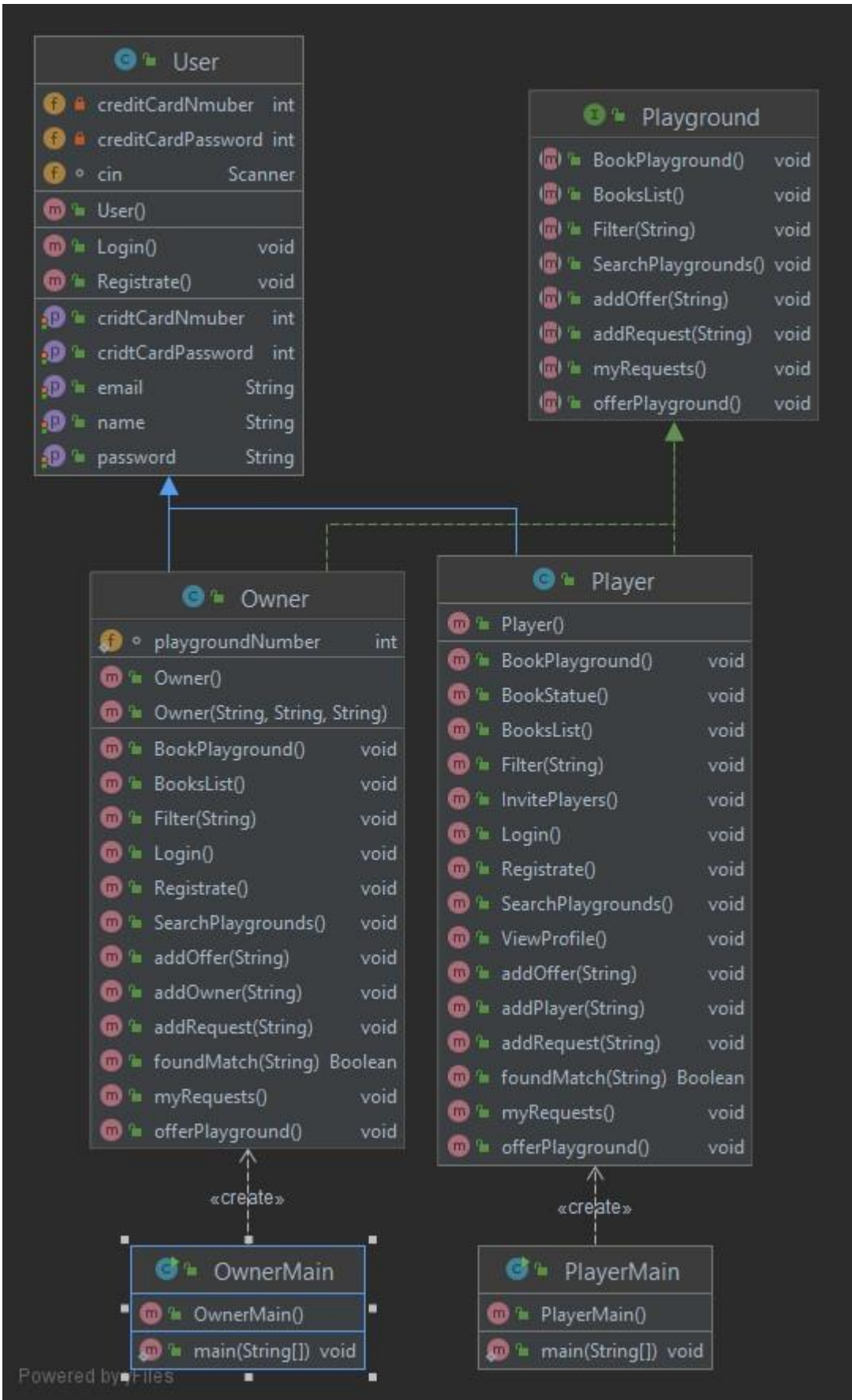
This document is for the project manager and customer to know the details will be applied in his application.



CS251: Phase 2 – **Creato Software Solution**  
Project: **GOFO**  
**Software Design Specification**

**System Models**

**I. Class Diagram(s)**





CS251: Phase 2 – **Creata Software Solution**  
Project: **GOFO**  
**Software Design Specification**

**II. Class Descriptions**

Class ID	Class Name	Description & Responsibility
1.	User	<p>It's a class contains the mutual attributes between the users such name, password, email, registration, and login.</p> <p><b>Responsibility:</b></p> <ul style="list-style-type: none"><li>• it allow all users types to extends from him to get the common features.</li><li>• it has<ul style="list-style-type: none"><li>a) 5 attributes common between the users such as <b>name, email, password, credit card number and its password,</b></li><li>b) It has <b>setters</b> and <b>getters</b> for these attributes.</li><li>c) It has 2 functions one for <b>Registrate()</b> that is for the users that open the app for the first time and the other for <b>Login()</b> that is for the an older user.</li></ul></li></ul>
2.	Playground	<p>It is an interface that contains an abstract function <b>Responsibility:</b></p> <ul style="list-style-type: none"><li>• It allow all users types to extends from him to get the common features.</li><li>• It has 8 functions that is abstracted which is declared only such as <b>BookPlayground(), offerPlayground() ,addOffer(), myRequests(), BookList(), Filter(), SearchPlaygrounds(), addRequest()</b></li></ul>
3.	Owner	<p>It is a class that creates an owner account if this user owns a playground.</p> <p><b>Responsibility:</b></p> <ul style="list-style-type: none"><li>• it allows the owner user to extends from User class and Playground interface to get the common features.</li><li>• it has<ul style="list-style-type: none"><li>a) 1 attribute which is playgroundNumber and it is static so that it could be accessed all over the class</li></ul></li></ul>



CS251: Phase 2 – **Creata Software Solution**  
Project: **GOFO**  
**Software Design Specification**

		<p>b) 9 functions common between the User class and Playground interface</p> <ul style="list-style-type: none"><li>• <b>Registrate():</b> This function allows the user to create a new owner account.</li><li>• <b>Login():</b> This function allows the user to enter his information so that he could enter his profile.</li><li>• <b>BookPlayground():</b> This function is not allowed for this user, it prints an invalid message for the user.</li><li>• <b>offerPlayground():</b> This function allows the owner to add an offer for his playground.</li><li>• <b>addOffer():</b> This function add the offer and saves it in the offer file.</li><li>• <b>myRequests():</b> It has an empty implementation because there is no requests for the owner.</li><li>• <b>Filter():</b>It has an empty implementation because there is no filtration for requests for the owner.</li><li>• <b>SearchPlaygrounds():</b>This function is not allowed for this user, it prints an invalid message for the user</li><li>• <b>addRequest():</b>This function is not allowed for this user, it prints an invalid message for the user.</li></ul> <p>c) 2 functions that is not common such as</p> <ul style="list-style-type: none"><li>• <b>addOwner():</b> This function saves the data of the new owner in the owner’s file.</li><li>• <b>foundMatch():</b> This function make sure that the user’s information are correct so that he could login successfully</li><li>• It contains a default constructor that allows the user either to</li></ul> <p>Registrate if he doesn’t have an account or Login if he has an account, also it contains a parameterized constructor that saves the data of the user such as <b>name</b>, <b>password</b> and <b>email</b>.</p>
--	--	---



CS251: Phase 2 – **Creata Software Solution**  
Project: **GOFO**  
**Software Design Specification**

4.	Player	<p>It is a class that creates an account if the user is a player and wants to play and search for a playground.</p> <p><b>Responsibility:</b></p> <ul style="list-style-type: none"><li>• it allows the player user to extends from User class and Playground interface to get the common features.</li><li>• it has<ol style="list-style-type: none"><li>a) 10 functions common between the User class and Playground interface.</li></ol></li><li>• <b>Registrate():</b> This function allows the user to create a new account for him.</li><li>• <b>Login():</b> This function allows the user to enter his information so that he could enter his profile.</li><li>• <b>BookPlayground():</b> This function allows the player to choose the offer he wants so that he could book a playground.</li><li>• <b>offerPlayground():</b>This function is not allowed for this user, it prints an invalid message for the user.</li><li>• <b>addOffer():</b>This function is not allowed for this user, it prints an invalid message for the user</li><li>• <b>myRequests():</b> it displays the requests for booking playgrounds for the player.</li><li>• <b>BookList():</b> This function saves the confirmed books in the owner’s offer so that that no conflict occurs.</li><li>• <b>Filter():</b>It has an empty implementation because there is no filtration for requests for the player.</li><li>• <b>SearchPlaygrounds():</b> for searching playgrounds in specific location</li><li>• <b>addRequest():</b>This function saves the requests of the player in the request file.</li></ul> <p>b) 4 functions that is not common such as,</p> <p><input type="checkbox"/> <b>foundMatch():</b>This function make sure that the user’s information are correct so that he could login successfully</p> <ul style="list-style-type: none"><li>• <b>InvitePlayers():</b> This function takes the emails of the other players to send them the invitation</li><li>• <b>ViewProfile():</b> This function displays the information of the user including his bookings and requests.</li></ul>
----	--------	---



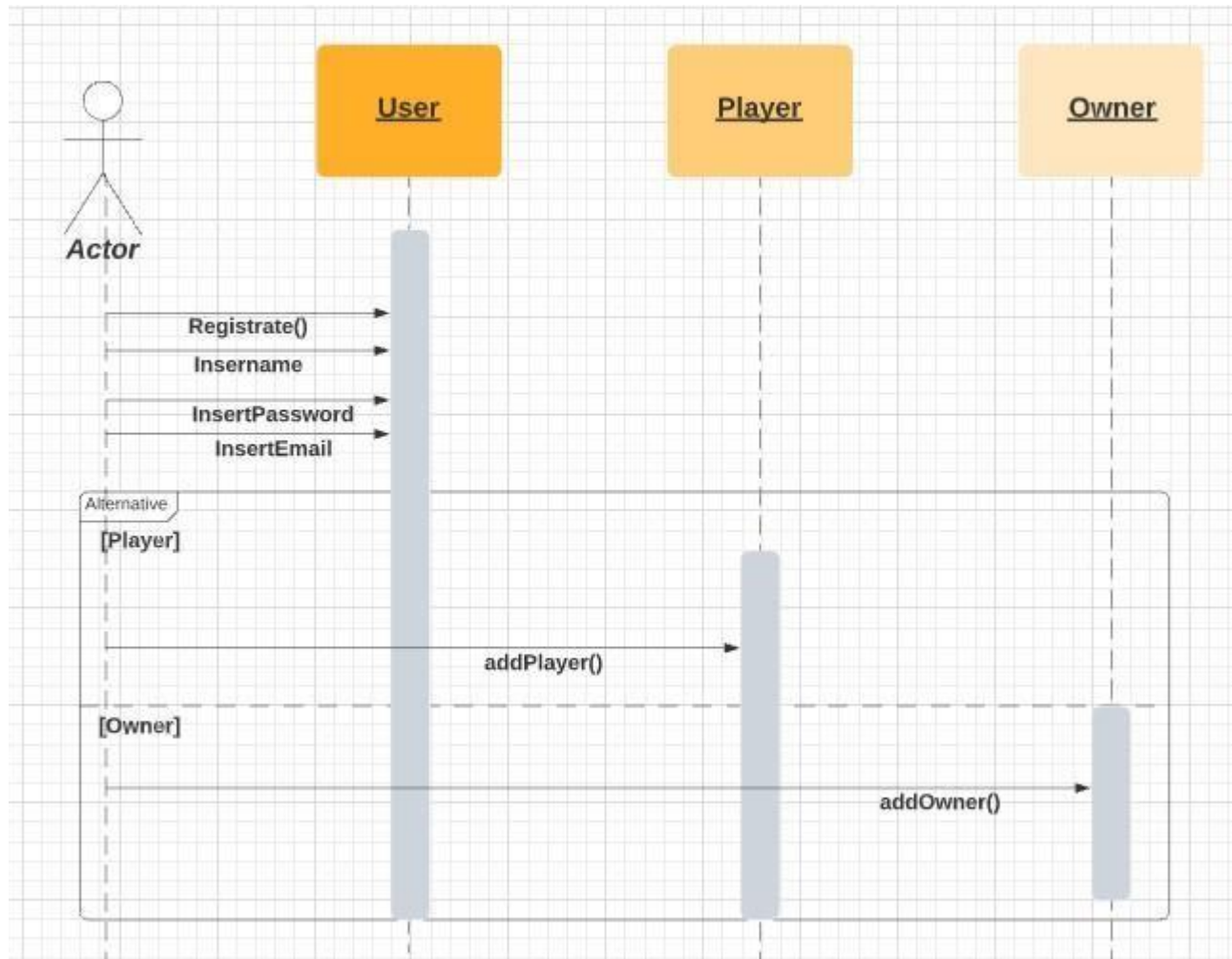


CS251: Phase 2 – **Creata Software Solution**  
Project: **GOFO**  
**Software Design Specification**

		<ul style="list-style-type: none"><li>• <b>addPlayer():</b> This function saves the data of the new player in the player's file</li><li>• It contains a default constructor that allows the user either to Register if he doesn't have an account or Login if he has an account.</li></ul>
--	--	--

**III. Sequence diagrams**

1. Registration: diagram for registration process for owner and player.

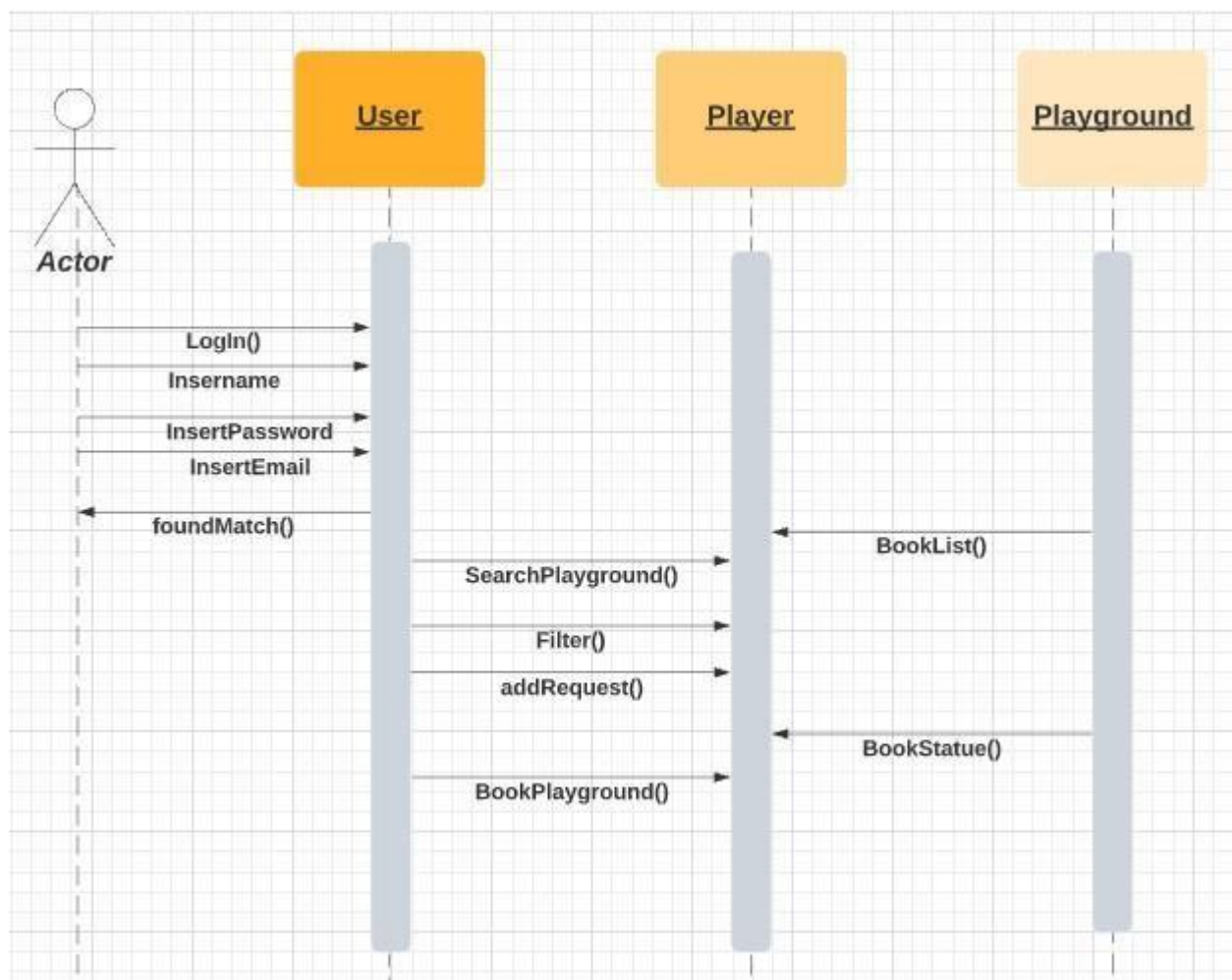






CS251: Phase 2 – **Creto Software Solution**  
Project: **GOFO**  
**Software Design Specification**

2. Booking: diagram for searching and filtering playgrounds.



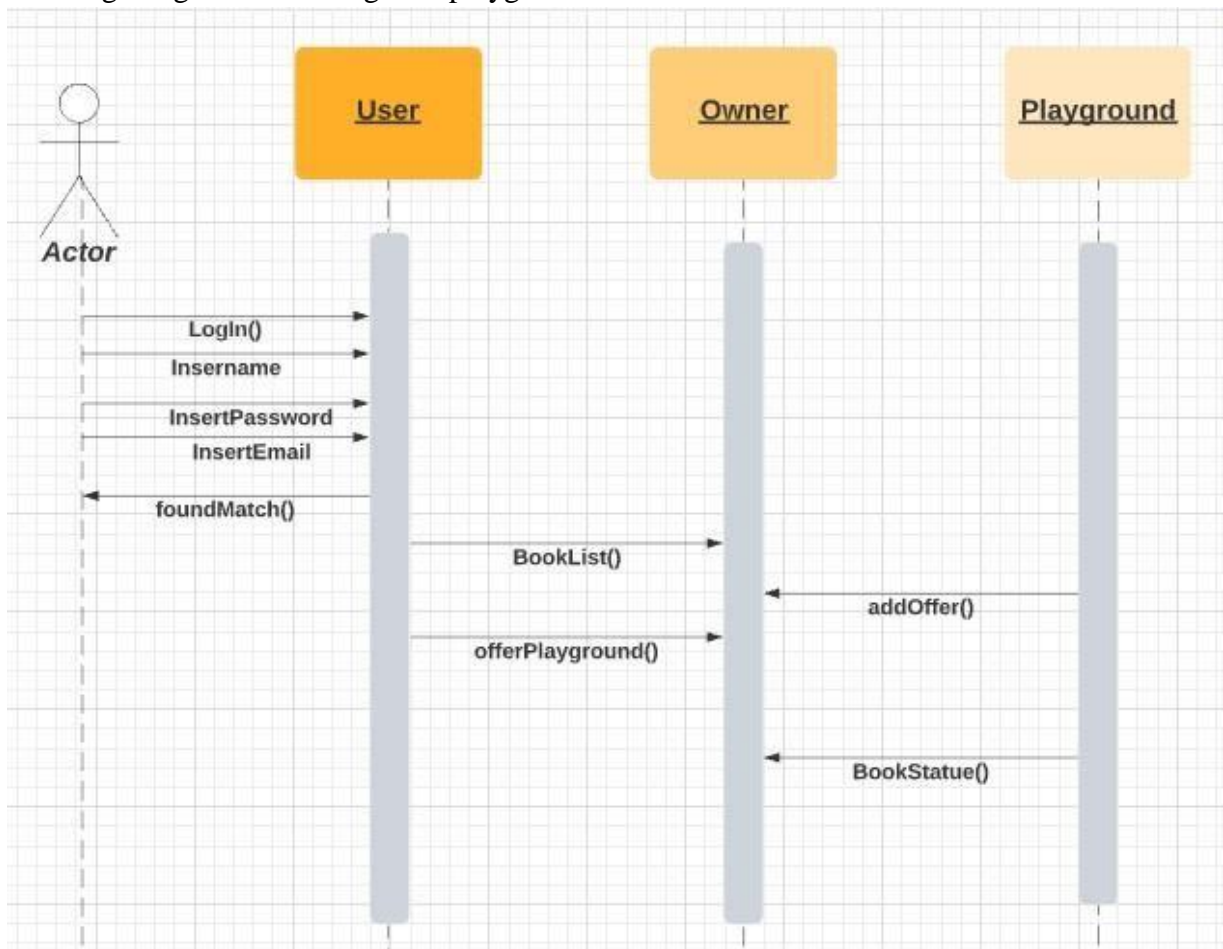


## CS251: Phase 2 – Create Software Solution

### Project: **GOFO**

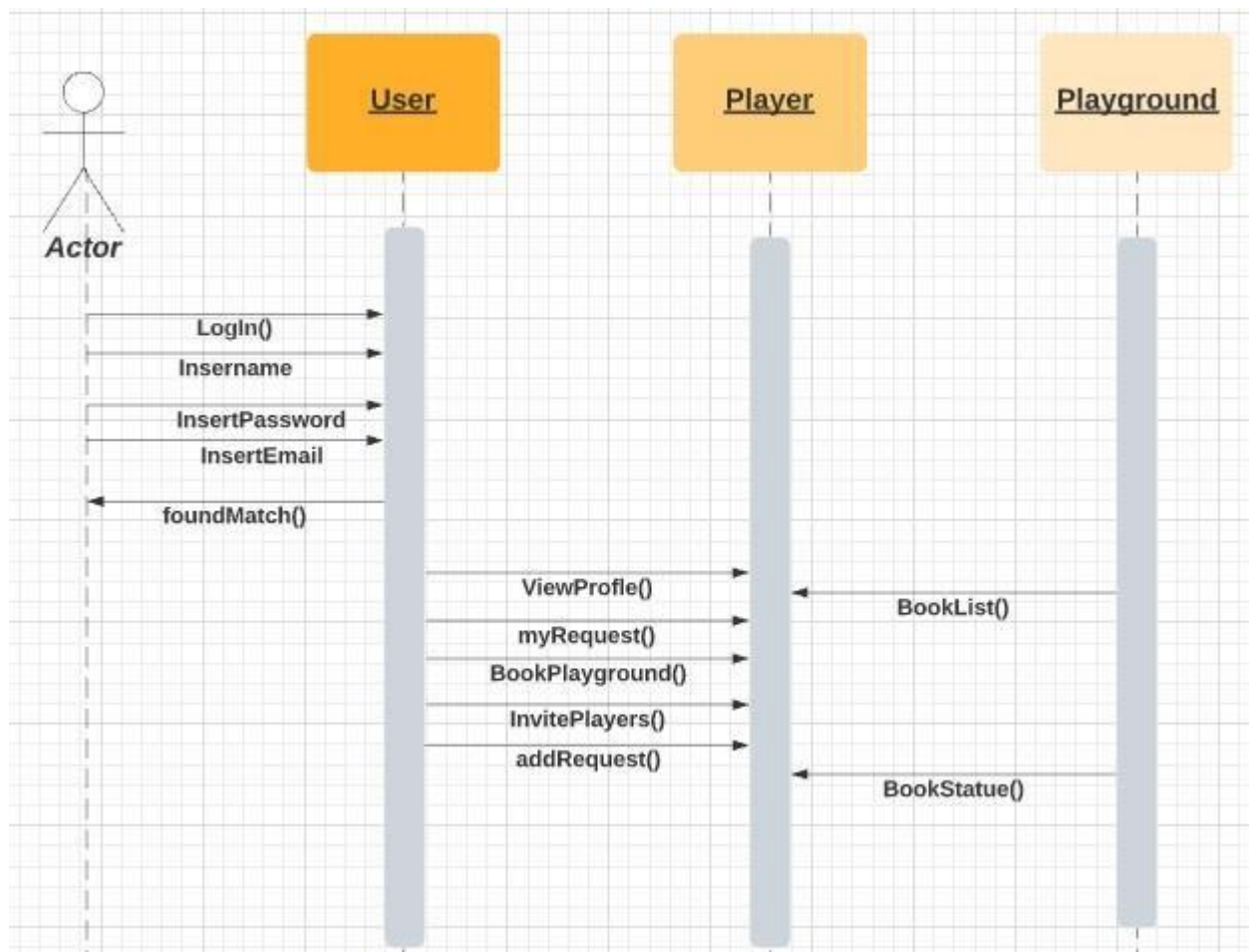
## Software Design Specification

3. Offering: diagram for adding new playgrounds offers.





4. Invite Players: diagram for inviting players to share a playground.



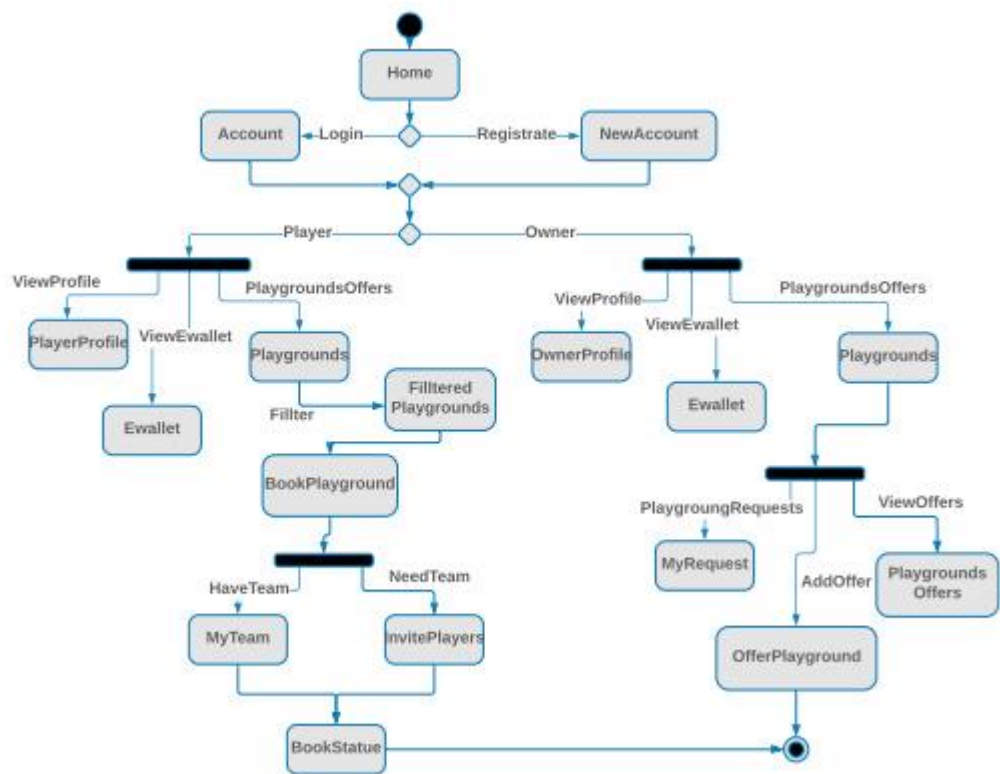


Class - Sequence Usage Table

Sequence Diagram	Classes Used	All Methods Used
1. Registration	User Player Owner	Registerate() addOwner() addPlayer()
2. Booking:	Player Playground User	LogIn() foundMatch() BookList() SearchPlayground() Filter() BookPlayground() addRequest() BookStatue()
3. Offer playground	User Owner Playground	LogIn() FoundMatch() BookStatue() offerPlayground() BookList() addOffer()
4. Invite players	Player Playground User	LogIn() FoundMatch() ViewProfile() BookPlayground() InvitePlayers() BookStatue()



State Diagram



Tools

- ArgoUML
- Lucidchart
- Intellij
- Git
- Github

Ownership Report

Item	Owners
Sequence Diagram, Sequence Usage Table, State Diagram.	Nahla Husain Ali
Class Description and responsibility, java documentation	Gowhara eid abdelghany
Class diagram, part of class description and responsibility, java code implementation.	Maryam Sayed Sadeek