**CS251**

**Software Engineering I**

Project Name

Software Design Specifications

Version X.X

Team Names and Emails

Month & Year

Contents

[Instructions [To be removed] 3](#_Toc72827671)

[Team 3](#_Toc72827672)

[Document Purpose and Audience 3](#_Toc72827673)

[System Models 3](#_Toc72827674)

[I. Class Diagram(s) 3](#_Toc72827675)

[II. Class Descriptions 5](#_Toc72827676)

[III. Sequence diagrams 5](#_Toc72827677)

[Class - Sequence Usage Table 6](#_Toc72827678)

[IV. State Diagram 7](#_Toc72827679)

[Tools 7](#_Toc72827680)

[Ownership Report 7](#_Toc72827681)

# Team

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Email** | **Mobile** |
| 20190590 | Nour amr nabil |  | 01116213211 |
| 20190107 | El-Sayed mostafa elsayed |  | 01005621787 |
| 20180165 | Alaa Maged Abdel\_aziz |  |  |

# Document Purpose and Audience

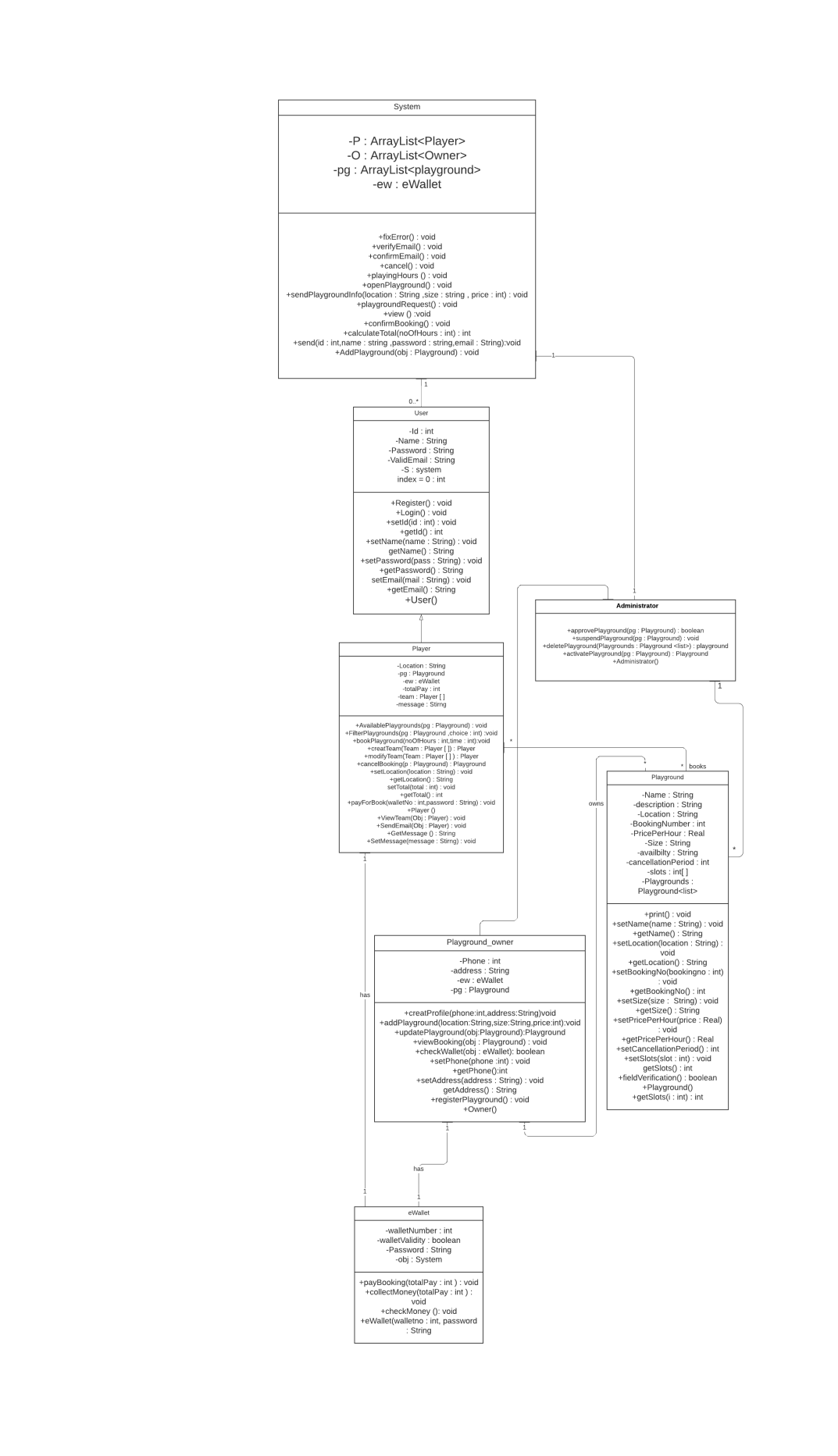
This document describes a booking system for football playgrounds

It is expected that both football playgrounds owners & administrators and players who are interested in booking football playgrounds will read this document

Those who do not care about booking football playgrounds, this document will not benefit them in anything, as it is directed to the owners of football playgrounds and players who are only interested in booking football playgrounds

# System Models

## I. Class Diagram(s)



1. **Class Descriptions**

|  |  |  |
| --- | --- | --- |
| **Class ID** | **Class Name** | **Description & Responsibility** |
| 1 | System | **Attributes :**   1. P : Arraylist<Player> 2. O : ArrayList<Owner> 3. pg : Arraylist<Playground>   **Operations :**   1. **fixError()** : void   (print out that there is an error)   1. **verifyEmail()** : void   (get email address and verify if it is correct or wrong)   1. **confirmEmail()** : void   (send confirmation message for an email)   1. **cancel()** : void (cancel creating an email) 2. **playingHours()** : void   (display all hours for each playground)   1. **openPlayground()** : void   (Allow owner to create new playground)   1. **sendPlaygroundInfo(location : String, size : String, price : Integer)** : void (collect all info about playground) 2. **playgroundRequest()** : void   (make a request to administrator to add new playground)   1. **view()** : void (display menu to player) 2. **confirmBooking()** : void   (confirm that a player booked a time slot)   1. **calculateTotal(noOfHours : Integer)** : Integer (calculates total payment for the player) 2. **send(id : Integer, name : String, password : String, email : String)** : void (sends all information about user to initiate new user) 3. **AddPlayground(playground: Playground)** : void (enables owner to add his playground to the system)   **Relations :**   1. one to one with administrator 2. one to many with users |

|  |  |  |
| --- | --- | --- |
| **Class ID** | **Class Name** | **Description & Responsibility** |
| 2 | User | **Attributes :**   1. Name : String 2. Password : String 3. ValidEmail : String 4. s : system 5. index=0 : integer   **Operations :**   1. **Register()** : void   (Create a new account for user)   1. **Login ()** : void   (login with a created account)   1. setName(name : String) : void 2. getName() : String 3. setPassword(pass : String) : void 4. getPassword() : String 5. setEmail(mail : String) : void 6. getEmail() : String 7. User()   **Relations :**  1. many to one with system |

|  |  |  |
| --- | --- | --- |
| **Class ID** | **Class Name** | **Description & Responsibility** |
| 3 | Owner | **Attributes :**   1. Phone: Integer 2. Address: String 3. ew: eWallet 4. pg: Playground   **Operations :**   1. **createProfile**(phone : Integer, address : String) : void   (enable the owner create profile to access the system and add playground)   1. **addPlayground**(location : String, size : String, price : Integer) : void   (enable the owner to add his playground to the system to be rented by players)   1. **updatePlayground**(obj : Playground) : Playground   (enable the owner to update any information from his playground)   1. **viewBooking**(obj : Playground) : void   (enable the owner to view all the bookings of his playground)   1. **checkWallet**(obj : eWallet) : Boolean   (enable the owner to check money on his eWallet)   1. **registerPlayground**() : void   (enable the owner to register his playground to the system to be able to put informations to the playground)   1. setPhone(phone : Integer) : void 2. getPhone() : Integer 3. setAddress(address : String) : void 4. getAddress() : String 5. Owner()   **Relations :**   1. inherits from user 2. one to many (ownes) with playground |

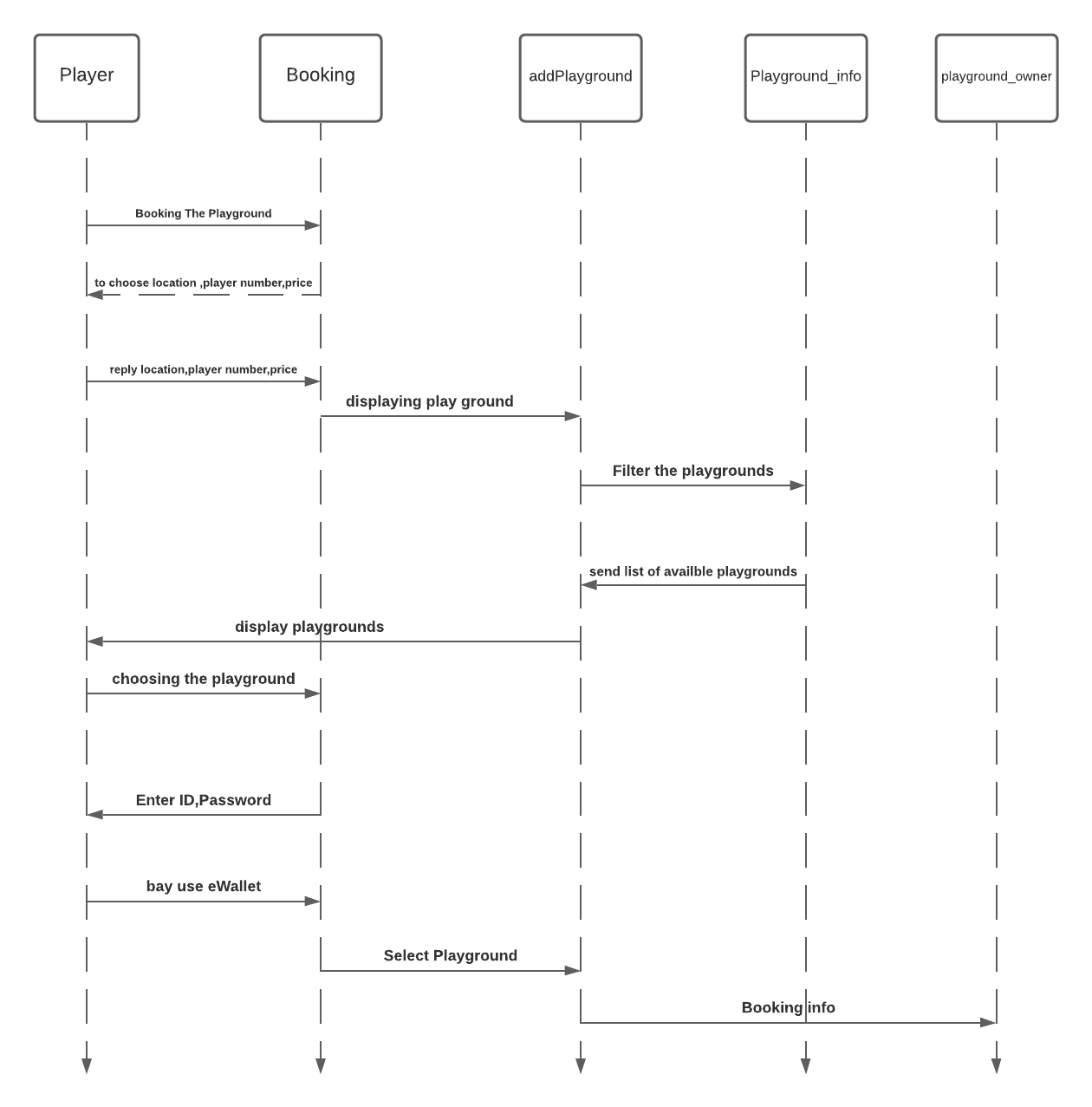
|  |  |  |
| --- | --- | --- |
| **Class ID** | **Class Name** | **Description & Responsibility** |
| 4 | Player | **Attributes :**   1. Location : String 2. pg : Playground 3. totalPay : Integer 4. Team : Player[ ] 5. message : string   **Operations :**   1. **AvailablePlaygrounds**(pg : Playground) : void   (enable the player to see all the playgrounds in the system that are available)   1. **FilterPlaygrounds**(pg : Playground, choice : Integer) : void   (enable the player to see an appropriate playgrounds according to the point that the player select)   1. **bookPlayground**(noOfHours : Integer, time : integer) : void   (enable the player to book playground with the no. of hours that he wants if available)   1. **createTeam**(Team : Player[]) : Player   (enable the player to create his team with his friends by inviting them)   1. **modifyTeam**(Team : Player[]) : Player   (enable the player to delete or add one or more to his team)   1. **cancelBooking**(p : Playground) : Playground   (enable the player cancel the booking only if on the cancelation period)   1. **payForBook**(walletNo : Integer, password : String) : void   (enable eWallet to withdraw the total pay of the booking from his wallet)   1. ViewTeam(Obj : Player) : void   (Print out all team members and their info)   1. **SendEmail(Obj : player)** : void (sends message to all team members) 2. setLocation(location : String) : void 3. getLocation() : String 4. setTotal(total : Integer) : void 5. getTotal() : Integer 6. Player() 7. GetMessage() : String 8. SetMessage(message : String) : void   **Relations :**   1. inherits from user 2. many to many (book) with playground |

|  |  |  |
| --- | --- | --- |
| **Class ID** | **Class Name** | **Description & Responsibility** |
| 5 | Administrator | **Operations :**   1. **approvePlayground**(pg : Playground) : Boolean   (enable the administrator to approve the playground after checking its information)   1. **suspendPlayground**(pg : Playground) : void   (enable the administrator to suspend the playground if there is a problem made by the owner or any incorrect information in the playground)   1. **deletePlayground**(Playgrounds : Playground<list>) : Playground   (enable the administrator to delete the playground if the owner made more than one problem with players)   1. **activatePlayground**(pg : Playground) : Playground   (enable the administrator activate any playground that suspended before)   1. Administrator()   **Relations :**   1. inherits from user 2. one to one with system 3. one to many (manager) with playground |

|  |  |  |
| --- | --- | --- |
| **Class ID** | **Class Name** | **Description & Responsibility** |
| 6 | Playground | **Attributes :**   1. Name : String 2. description : String 3. Location : String 4. BookingNumber : Integer 5. PricePerHour : Real 6. Size : String 7. availability : String 8. cancellationPeriod : Integer 9. slots : Integer[] 10. Playgrounds : Playground<list>   **Operations :**   1. **print()** : void   (print all information about playground)   1. **fieldVerification()** : Boolean (make sure that all fields are filled) 2. setName(name : String) : void 3. getName() : String 4. setLocation(location : String) : void 5. getLocation() : String 6. setBookingNo(bookingno : Integer) : void 7. getBookingNo() : Integer 8. setSize(size : String) : void 9. getSize() : String 10. setPricePerHour(price : Real) : void 11. getPricePerHour() : Real 12. setCancellationPeriod(canc : Integer) : void 13. getCancellationPeriod() : Integer 14. setSlots(slot : Integer) : void 15. getSlots() : Integer 16. getSlots(I : integer) : integer 17. Playground()   **Relations :**   1. many to one (managed by ) with administrator 2. many to many (booked by ) with player 3. many to one (owned by) with owner |

|  |  |  |
| --- | --- | --- |
| **Class ID** | **Class Name** | **Description & Responsibility** |
| 7 | eWallet | **Attributes :**   1. walletNumber : Integer 2. walletValidity : Boolean 3. Password : String 4. obj : System   **Operations :**   1. **payBooking(totalPay : Integer)** : void (takes money from player's ewallet) 2. **collectMoney(totalPay : Integer)** : void (adds money token from player to owner's ewallet) 3. **checkMoney()** : void   (print how much money are there in the e wallet)   1. eWallet(walletno : Integer, password : String)   **Relations :**   1. one to one with owner 2. one to one with player |

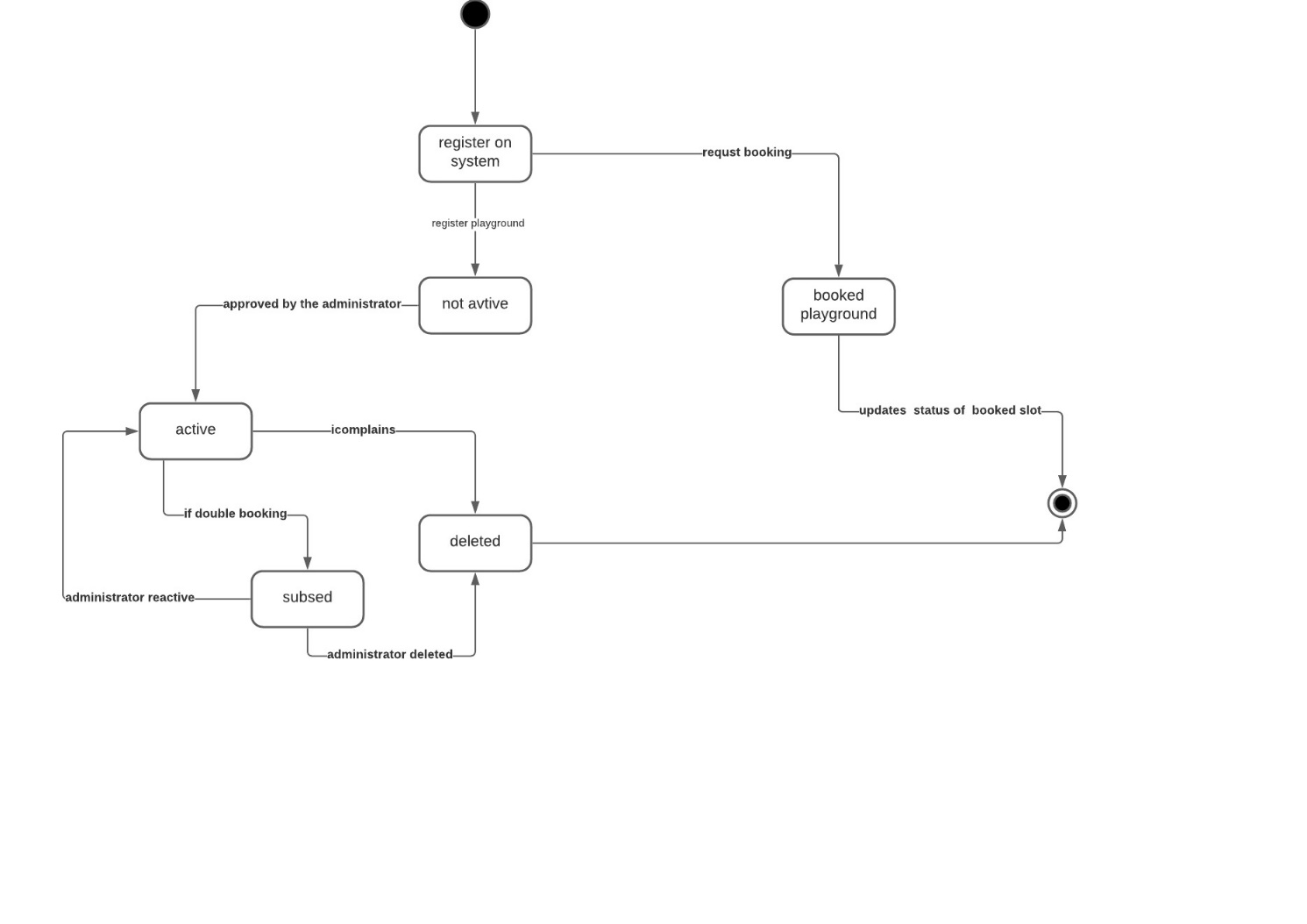
## III. Sequence diagrams

****

Class - Sequence Usage Table

|  |  |
| --- | --- |
| Class player  Class booking | 1.booking playground()  2.choose location ()  3.choose player number()  4.choose location ()  5.Reply location()  6.Reply player number ()  7.Reply price ()  8.choosing play ground()  9.Enter id (id)  10.Enter password(password)  11.bay use ewallet() |
| Class booking  Class add Playground | 1.displaying play ground()  2.select play ground () |
| Class player  Class add play ground | 1. display play ground() |
| Class Playground info  Class add playground | 1.filter play ground ()  2.send list of available play ground () |
| Class add playground  Class playground owner | 1. Booking info () |

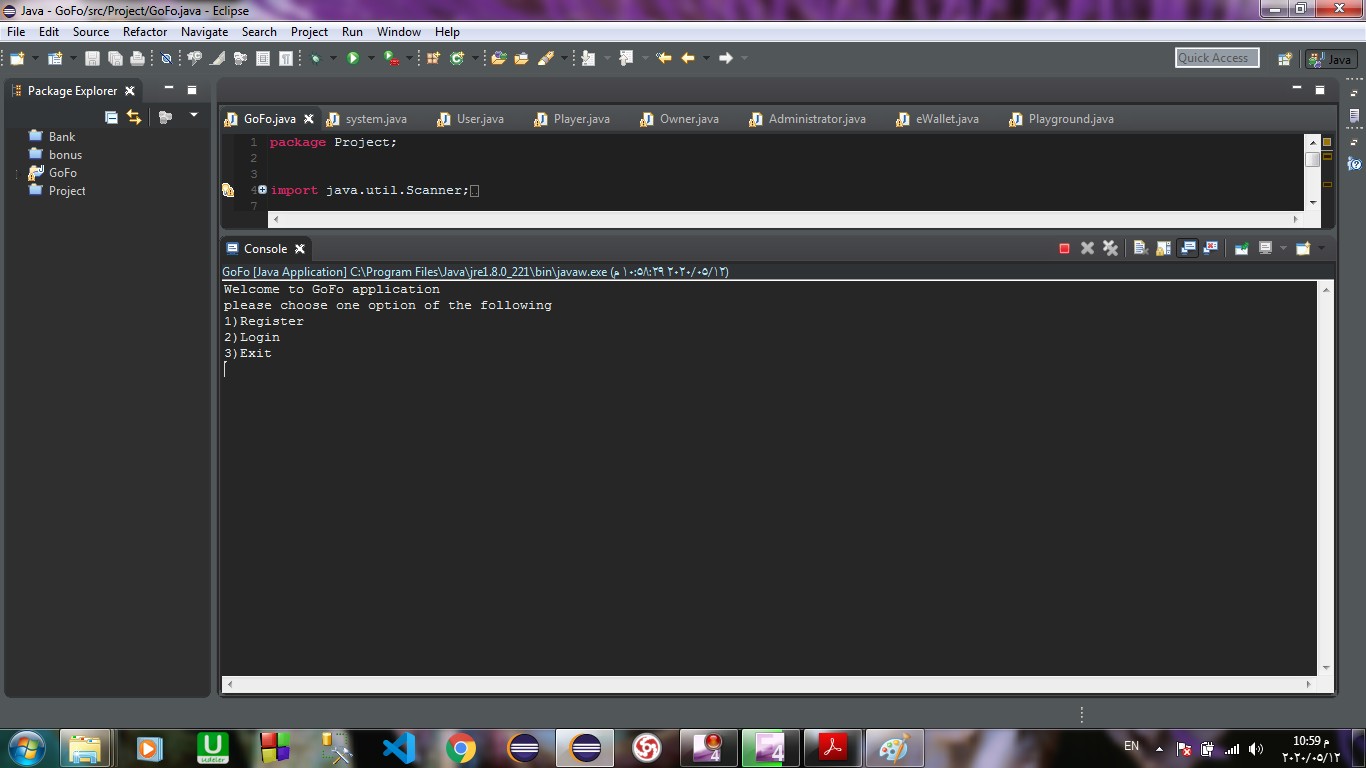
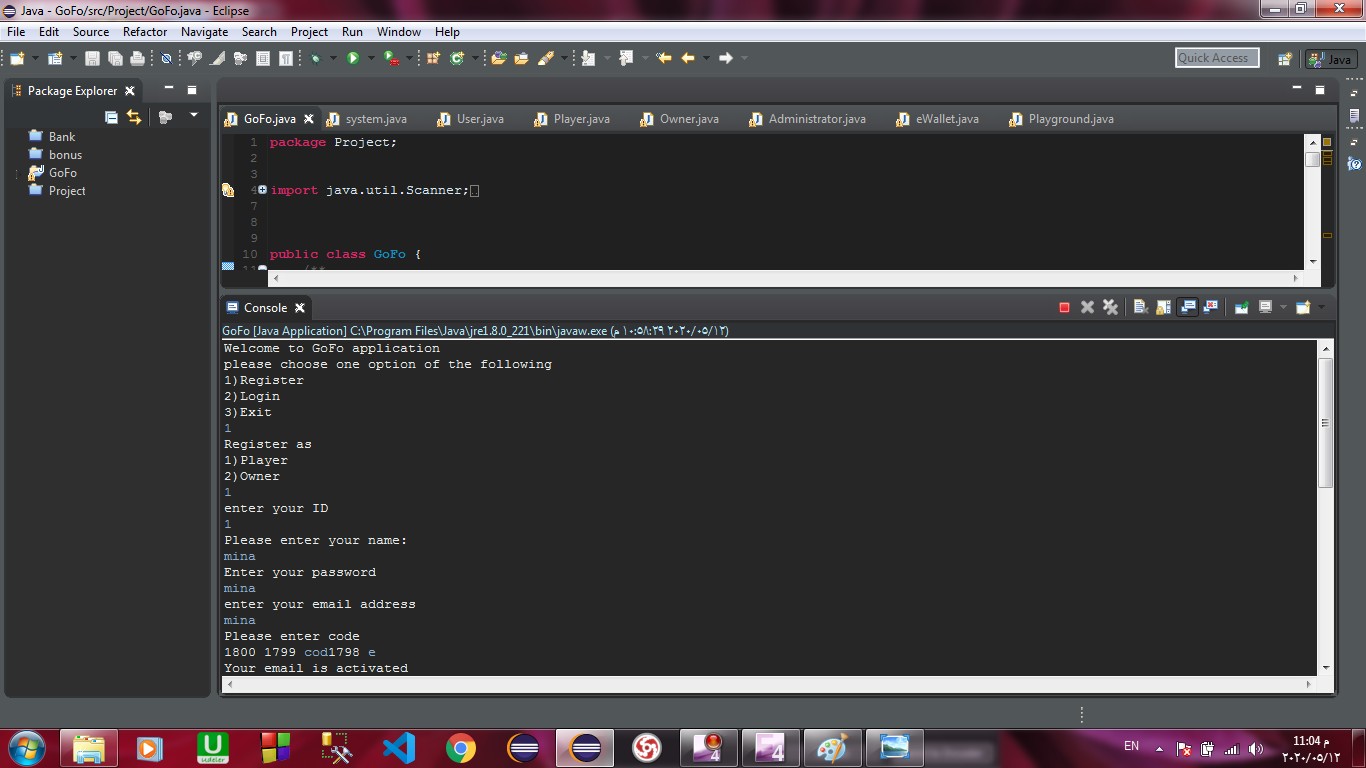
### Class - Sequence Usage Table

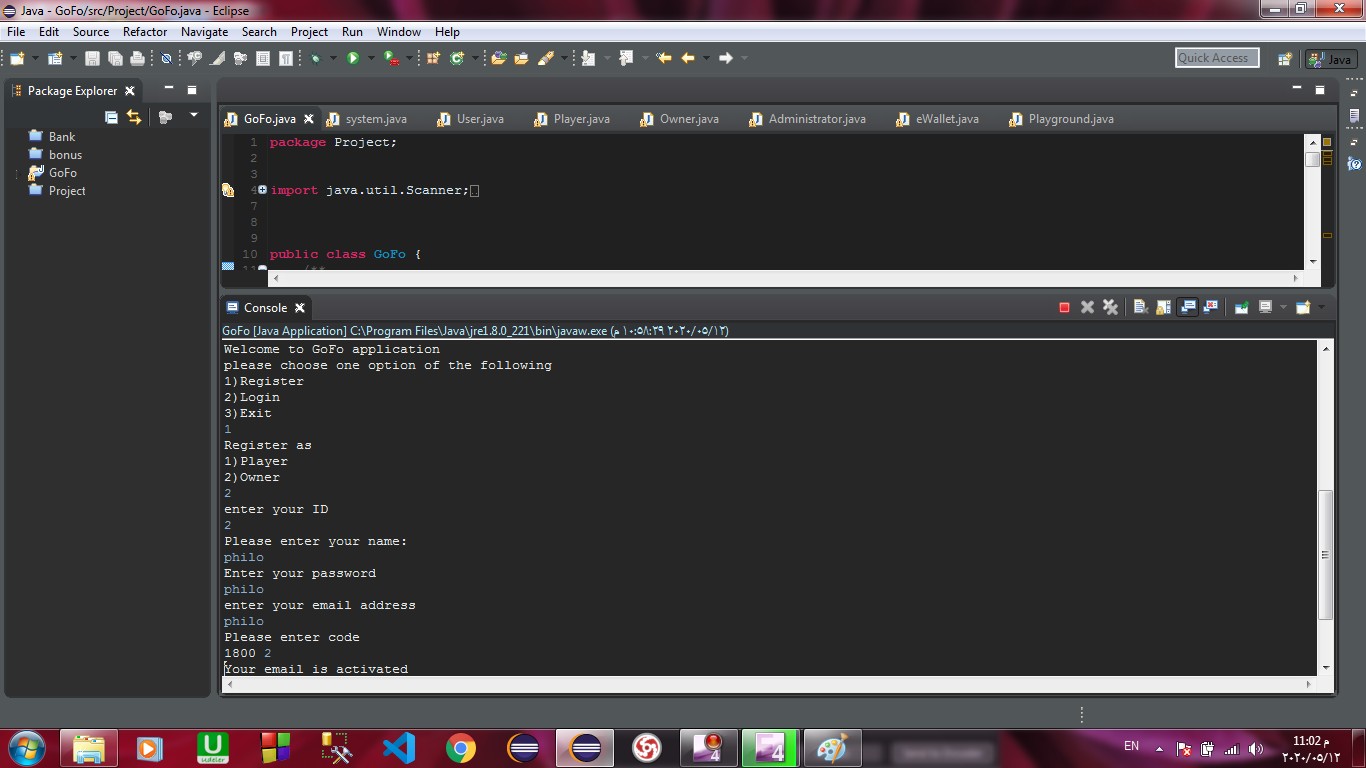


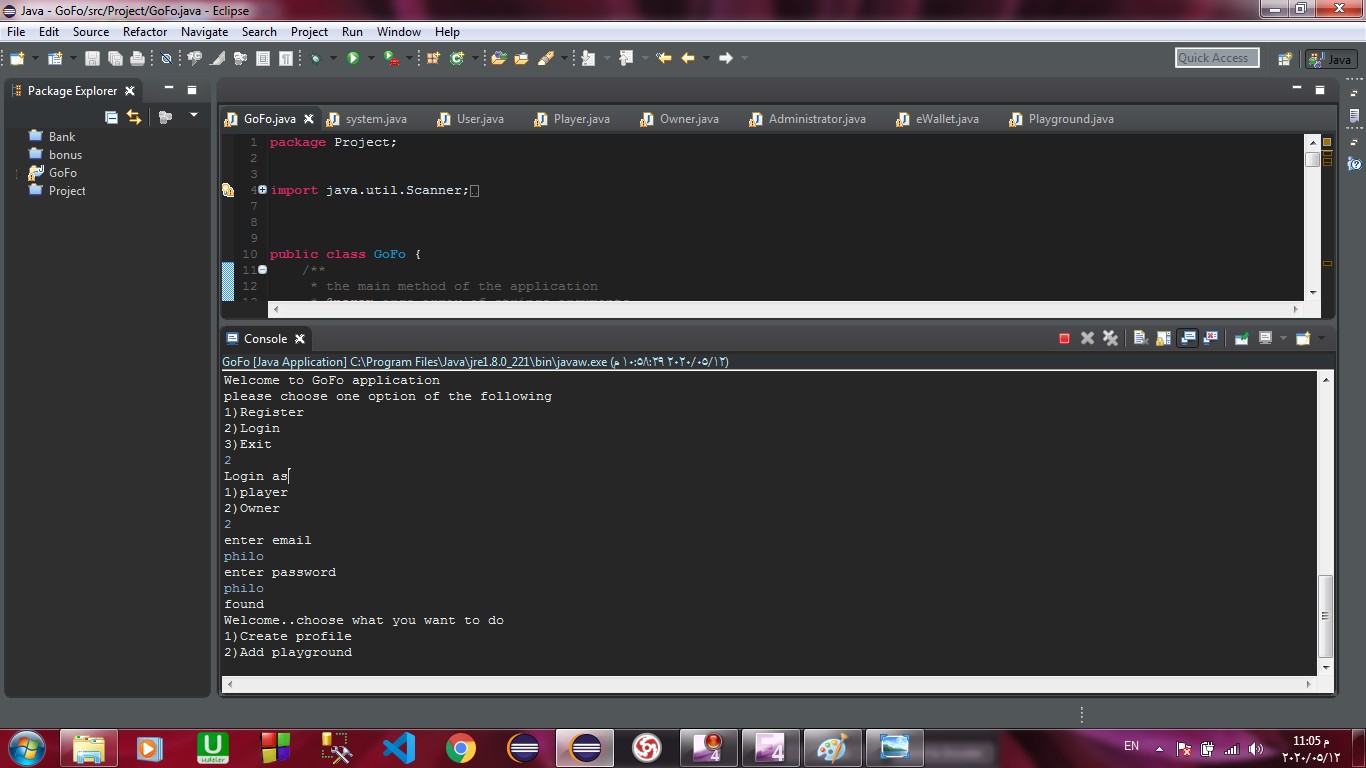
## IV. State Diagram

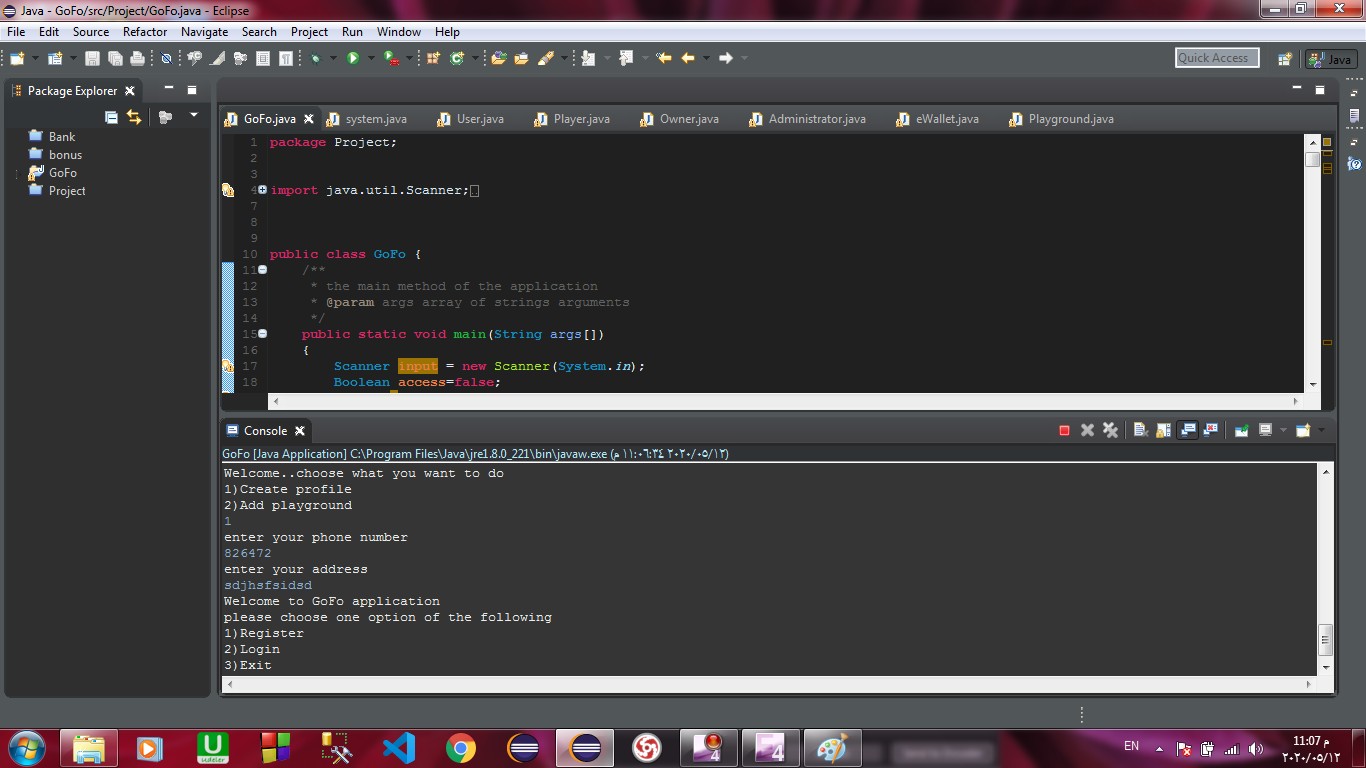
# 

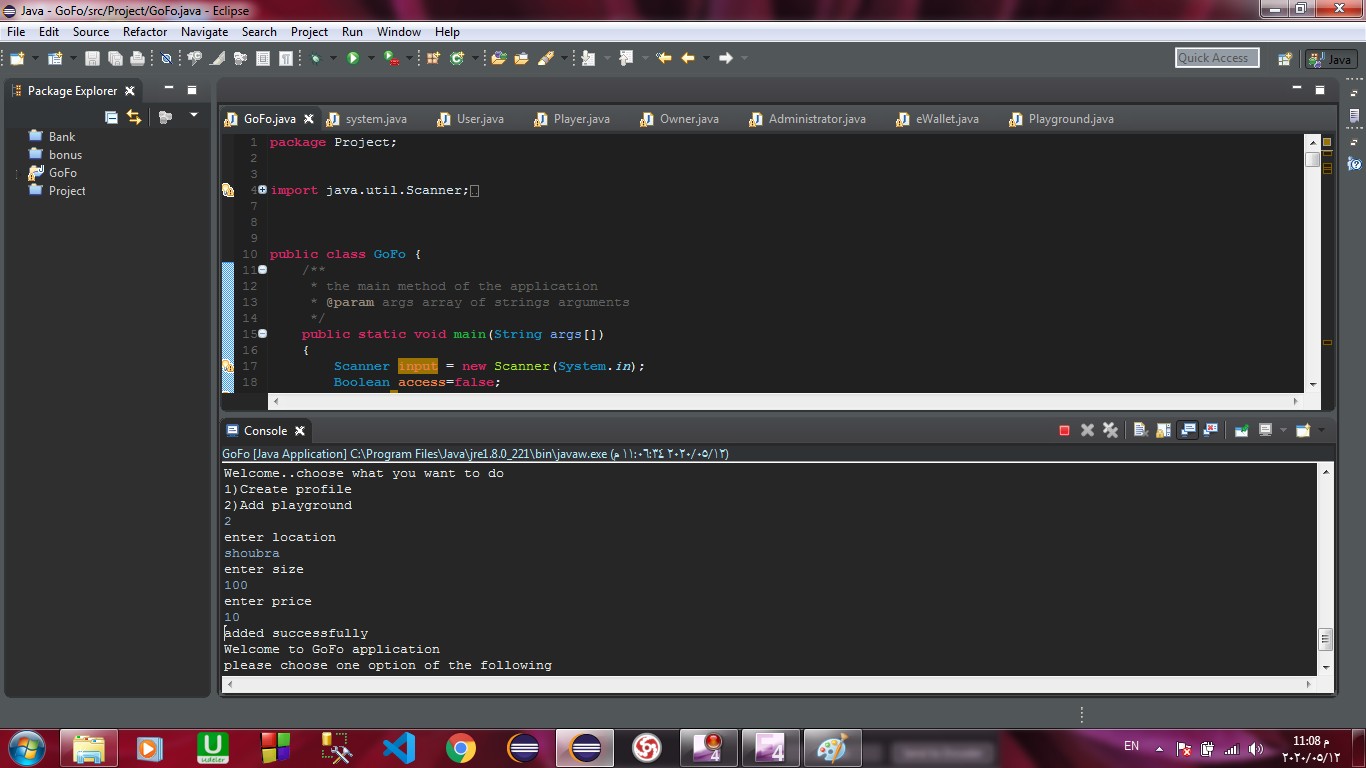
**Screenshots**

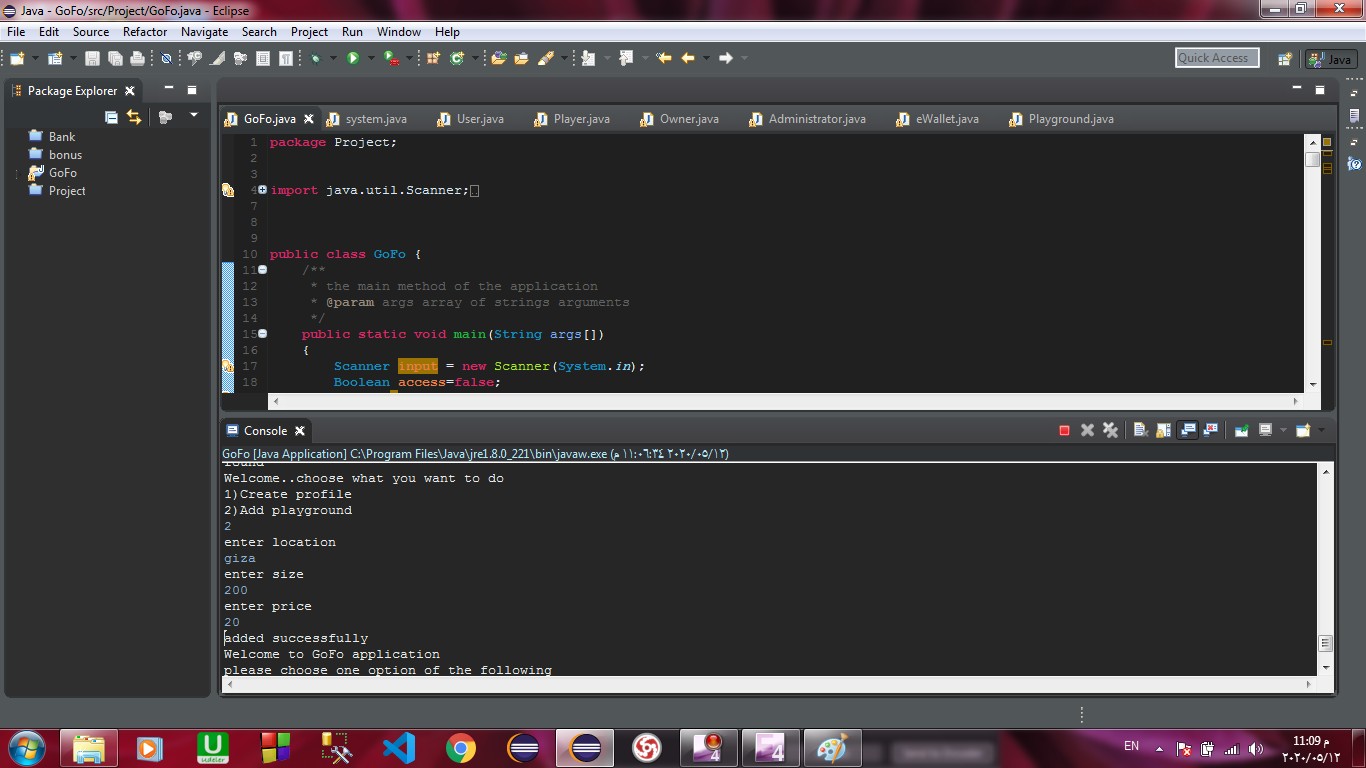
1. **Main screen**
2. **Registered as player**
3. **Registered as owner**

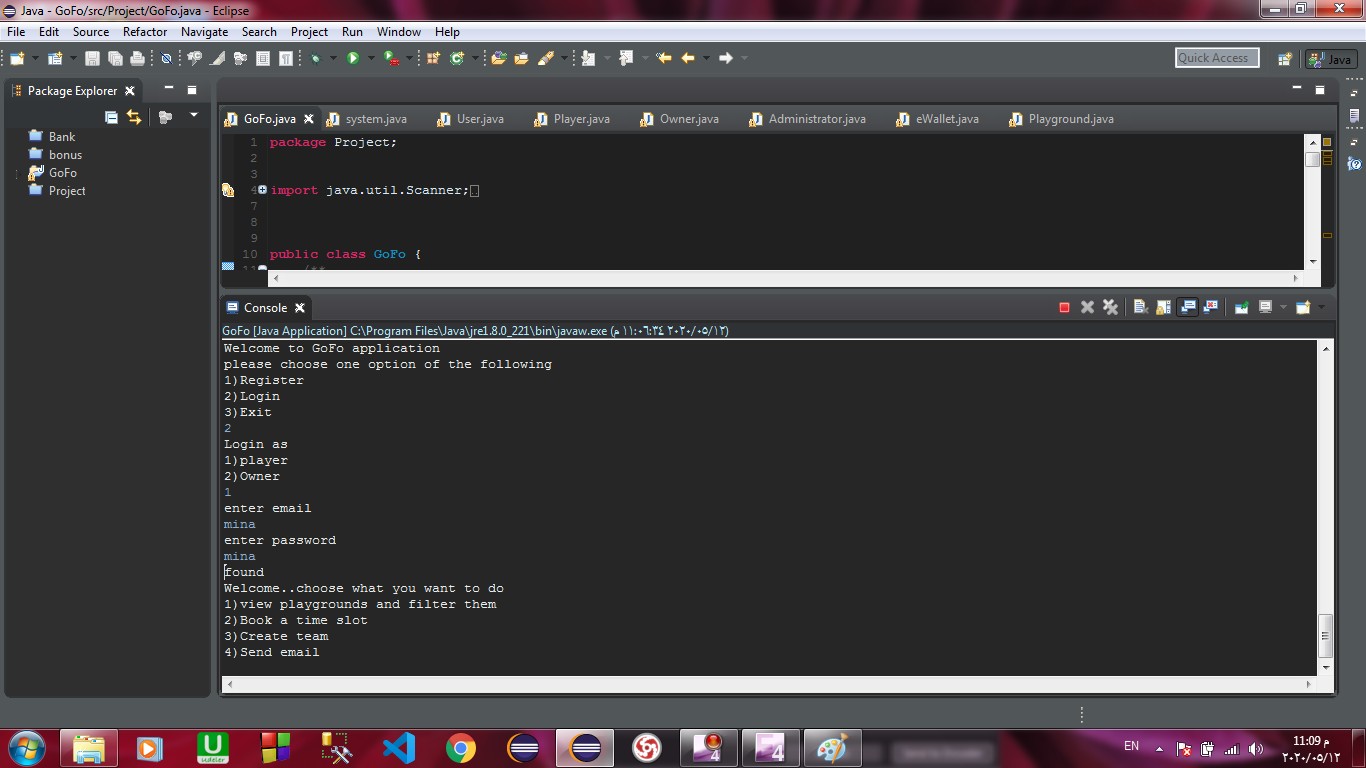


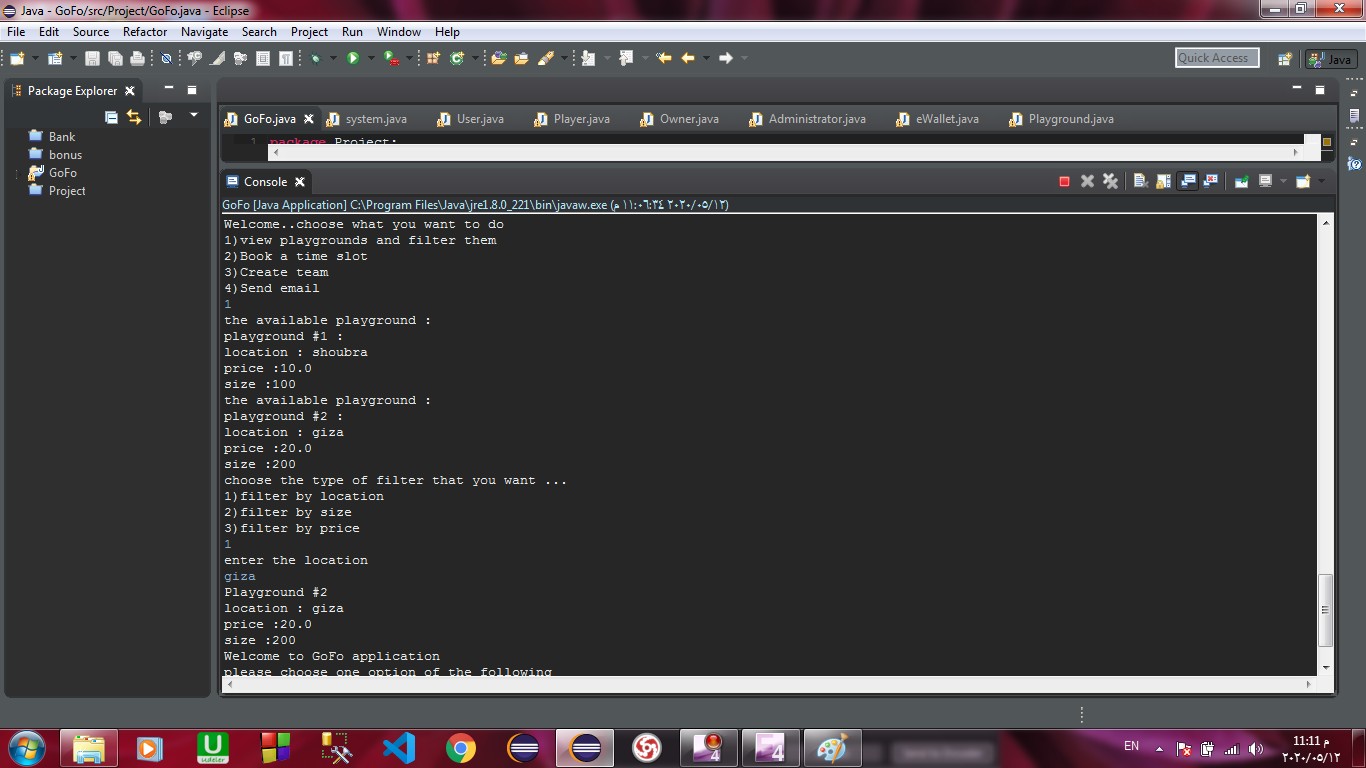
1. **Login as owner**
2. **Create profile as owner**

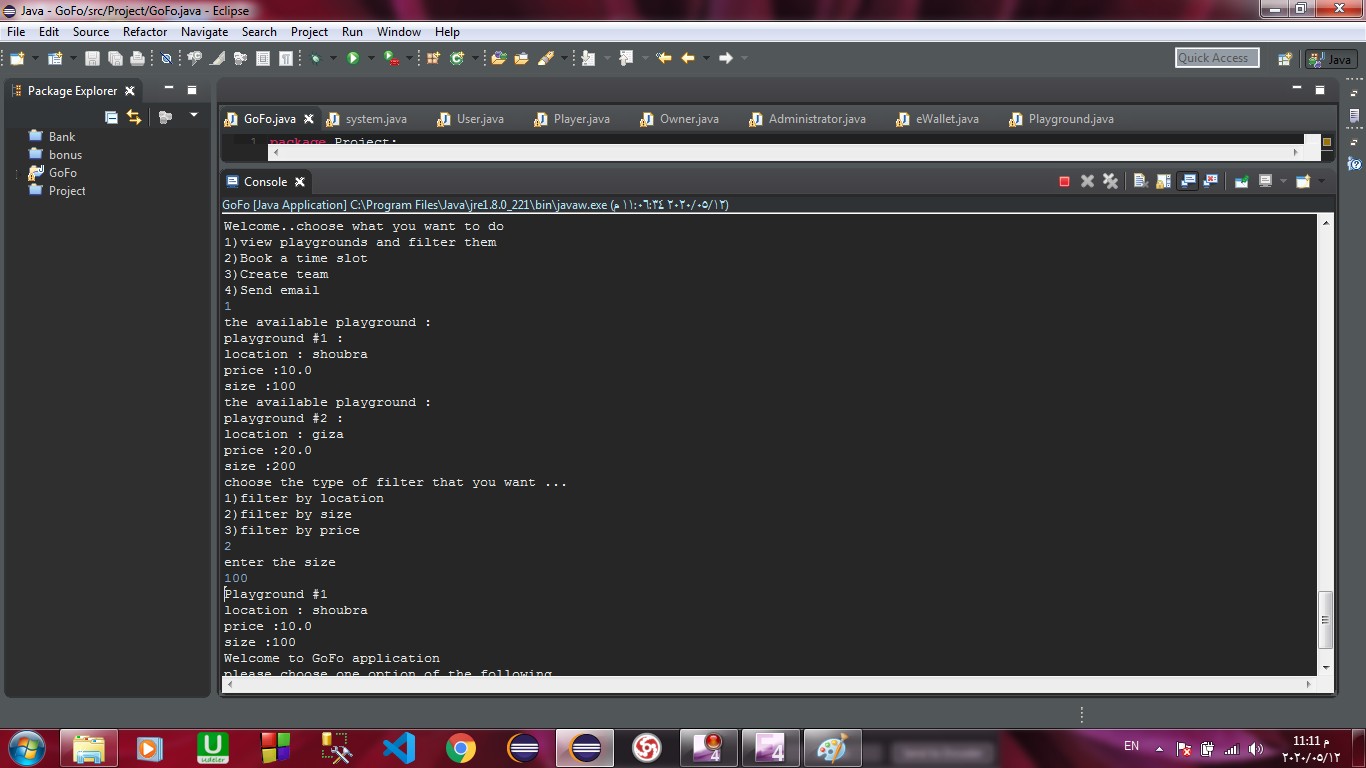


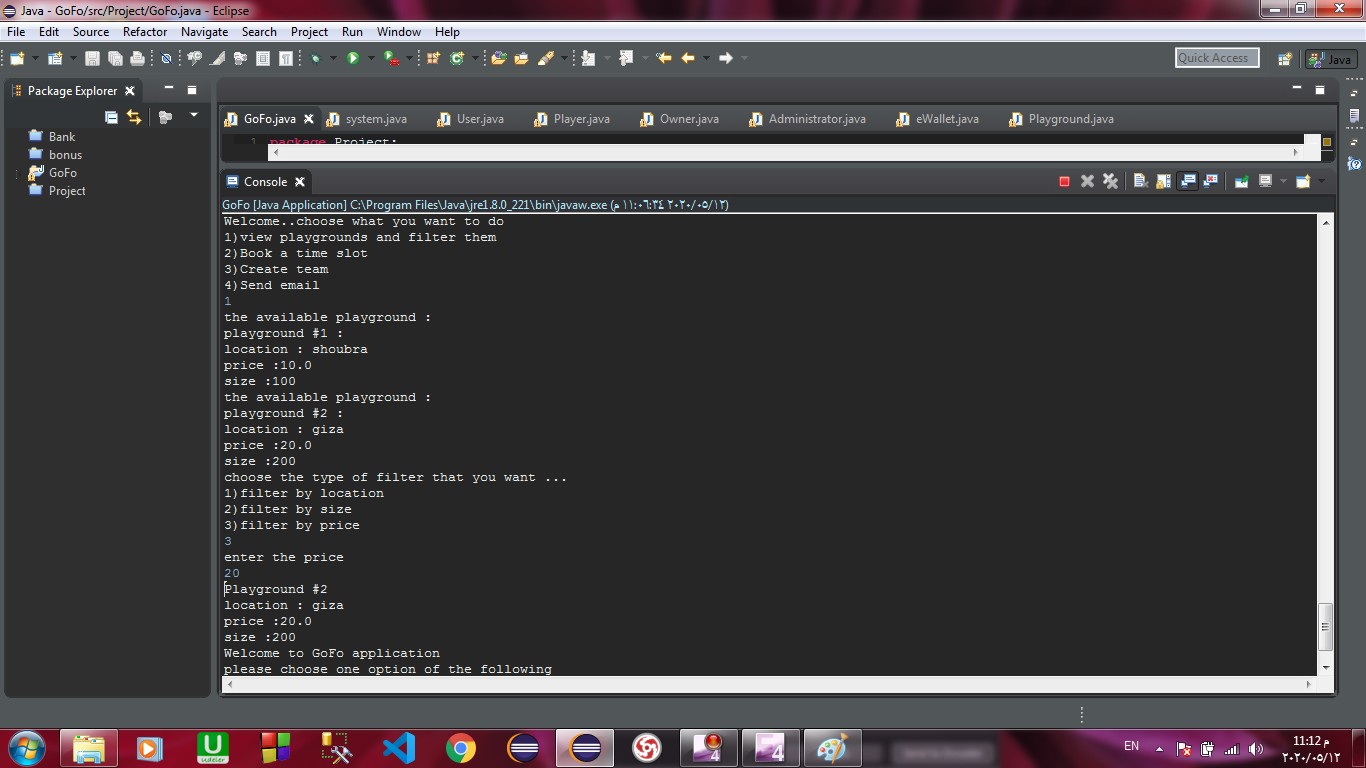
1. **Add Playground 1**
2. **Add playground 2**

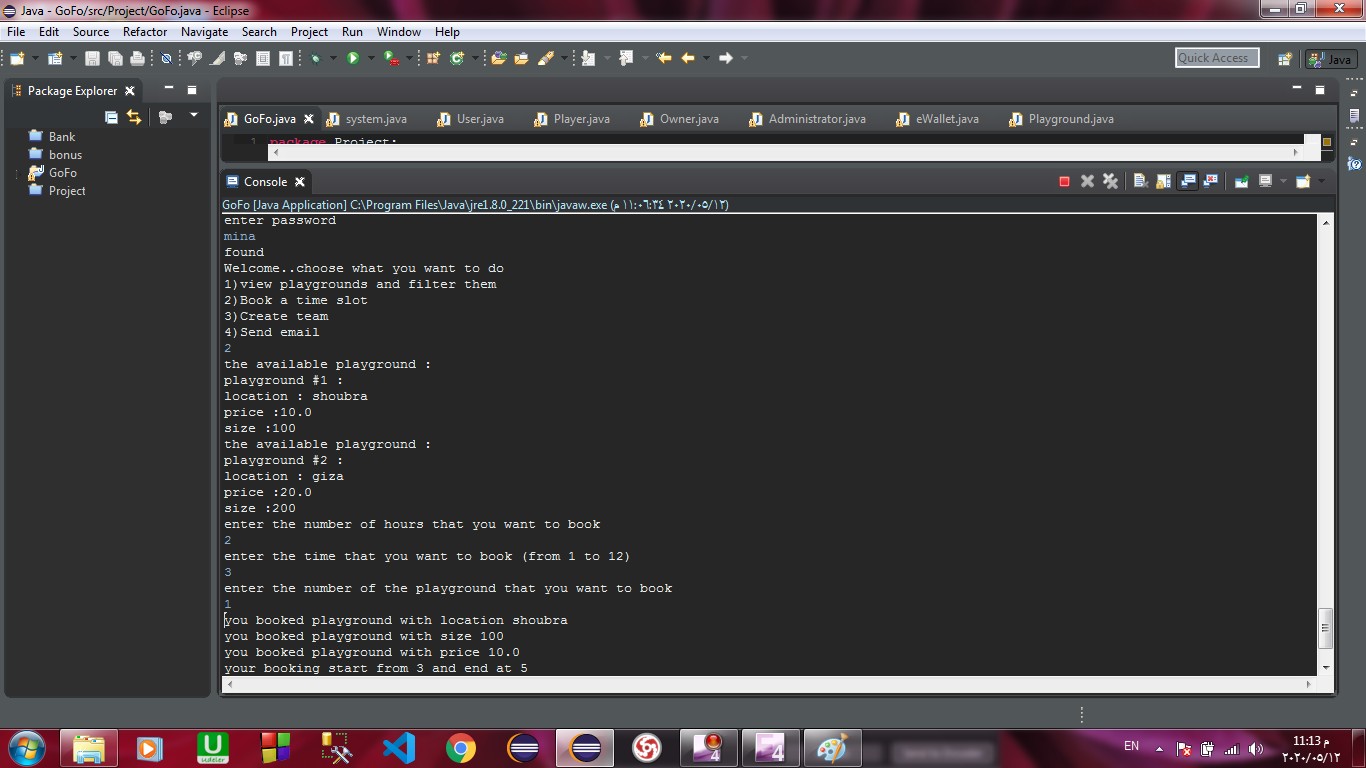


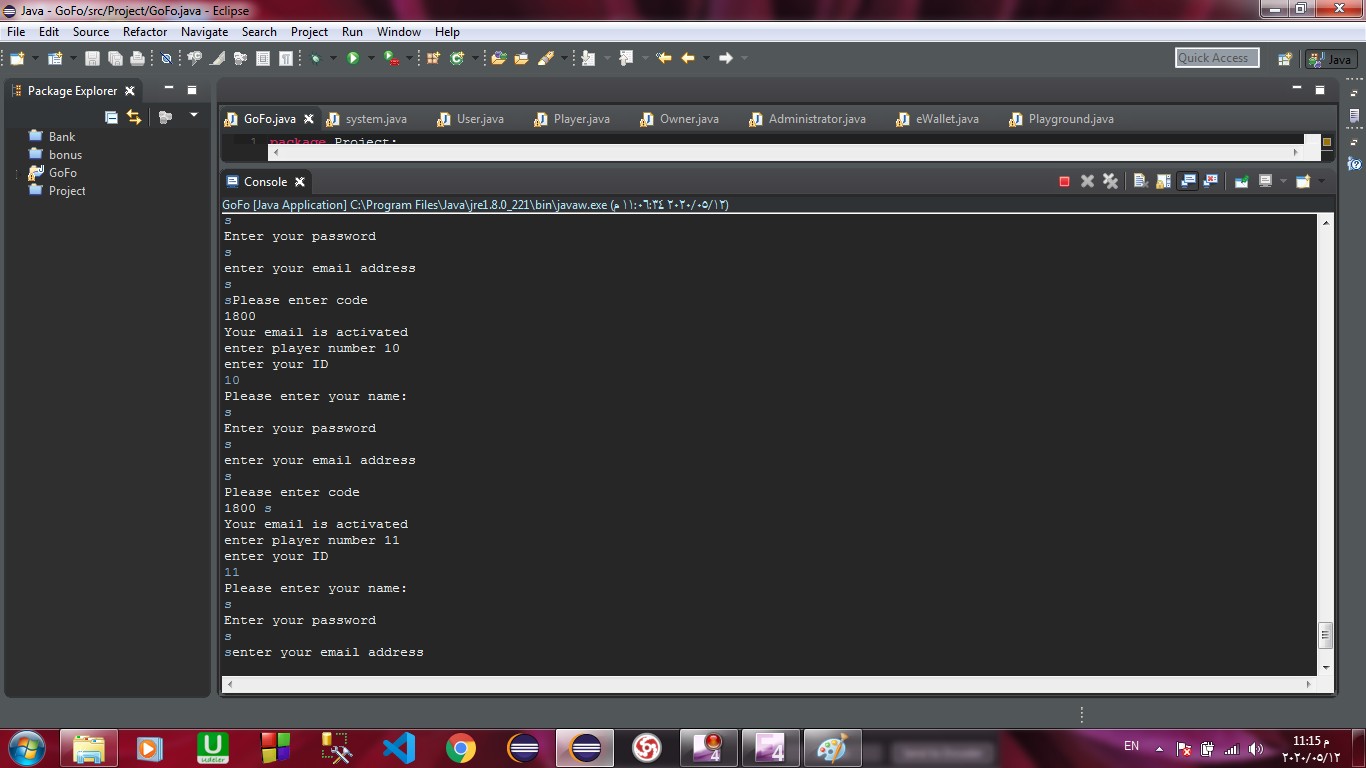
1. **Login as player**
2. **Filter by location**

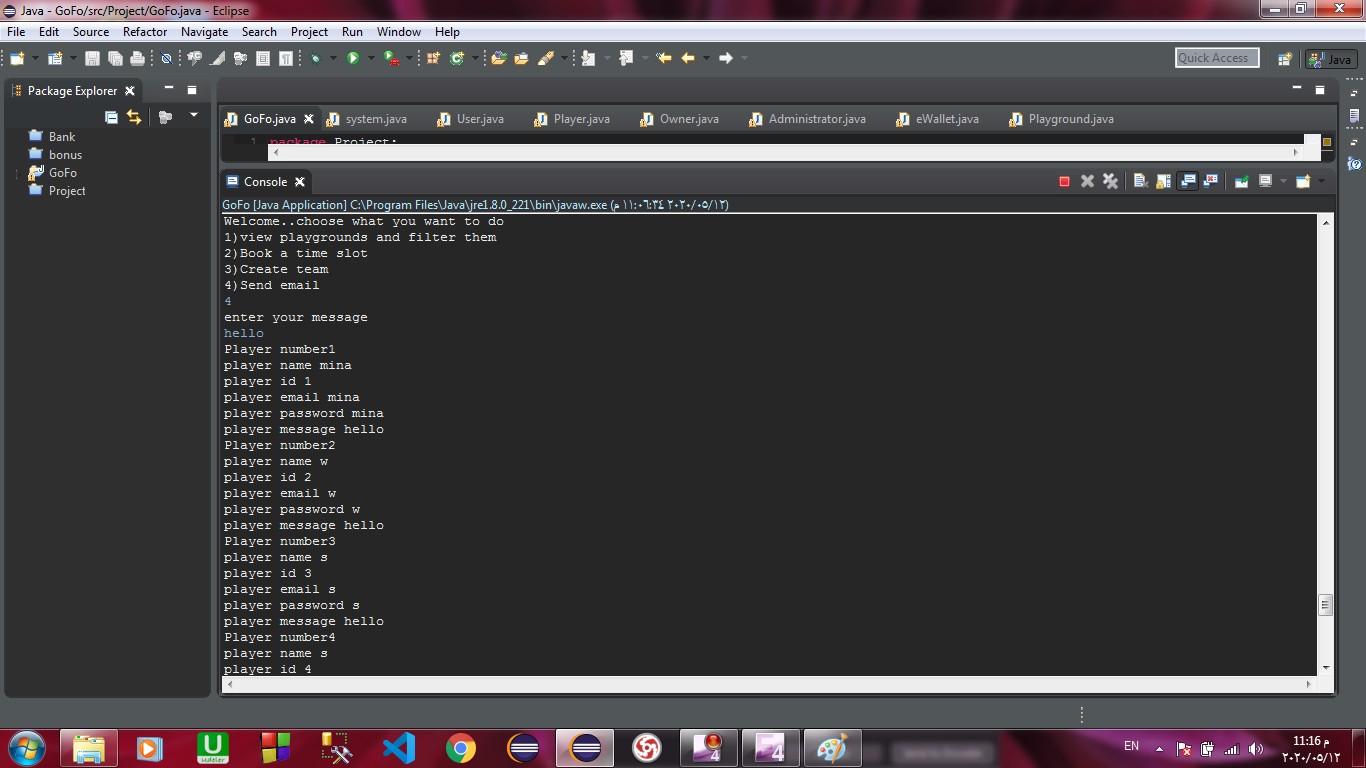


1. **Filter by size**
2. **Filter by price**



1. **Book a time slot**
2. **Create a team of 11 players**



1. **Send email**

# Tools

# lucid.app

1. **Esclipse**
2. **GitHub**

# Ownership Report

|  |  |
| --- | --- |
| **Item** | **Owners** |
| Nour amr nabil | **Part of state diagram , some classes in java** |
| El-sayed mostafa | **Part of**  **Sequence diagram and some classes in java** |
| Alaa Maged Abdel | *Part of class diagram* |

**https://github.com/Sayed2002/homework-4**