بسم الله الرحمن الرحيم

CS251
Software Engineering
Software Design Specifications of GoFo project<pmteam>

Team members

Name	Id	e-mail	Phone
Abdelrahman sayed abdallah	20180152	abdelrahmansayed913@gmail.com	01020919239
Malak essam anwer	20180287	malakmeke100@gmail.com	01211276564

Directed to:

Dr.Moammed el-Ramly

1/6/2020

CS251: final project – <PM team >
Project: <Gofo >

Software Design Specification

Contents

CS251: final project – <PM team >
Project: <Gofo >

Software Design Specification

Introdution

- Our system is a booking system for football playgrounds. It serves the players and playground owners as well. An administrator oversees the overall operations of the system and ensures that no fraud takes place. Anyone can register himself on the system and create a profile. He can see the playgrounds near to him or in a specific area or all of them.

Team

ID	Name	Email	Mobile
20180152	Abdelrahman sayed Abdallah	Abdelrahmansayed913@gmail.com	01020919239
20180287	Malak essam anwer	malakmeke100@gmail.com	01211276564

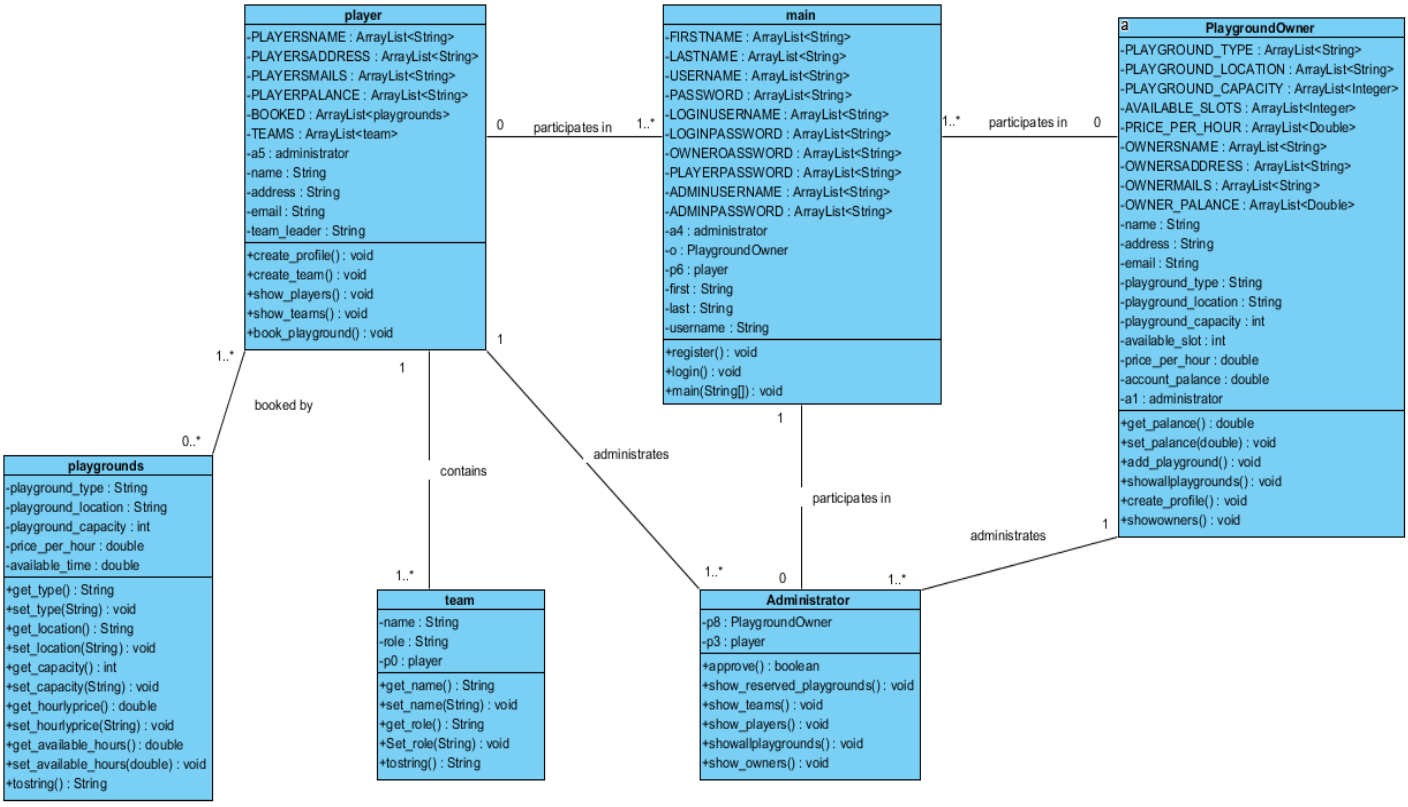
Document Purpose and Audience

- **Our purpose is make life easy for people who want to play football .and we hope footballers like our application as its purpose is to safe time , money and effort.**
- **Our audience is youth.**

CS251: final project – <PM team >
Project: <Gofo >

Software Design Specification

I. Class Diagram(s)



Software Design Specification

II. Class Descriptions

Class ID	Class Name	Description & Responsibility
1.1	User	This class is responsible for logging in or signing up by inserting password and user name by the function login() or sign up().and the entered data is sent to a database system to verify the data and send a response message to the user that the data is accepted or not . if accepted the user will move into the screen with which he associated according the entered data . if not the database system will reject it and ask the user will be asked to reenter data .and this is what the user class is responsible for. And finally the components of the user class : (1)attributes: -username ->string -password -> int (2)methods +login() ->void sign up() >void

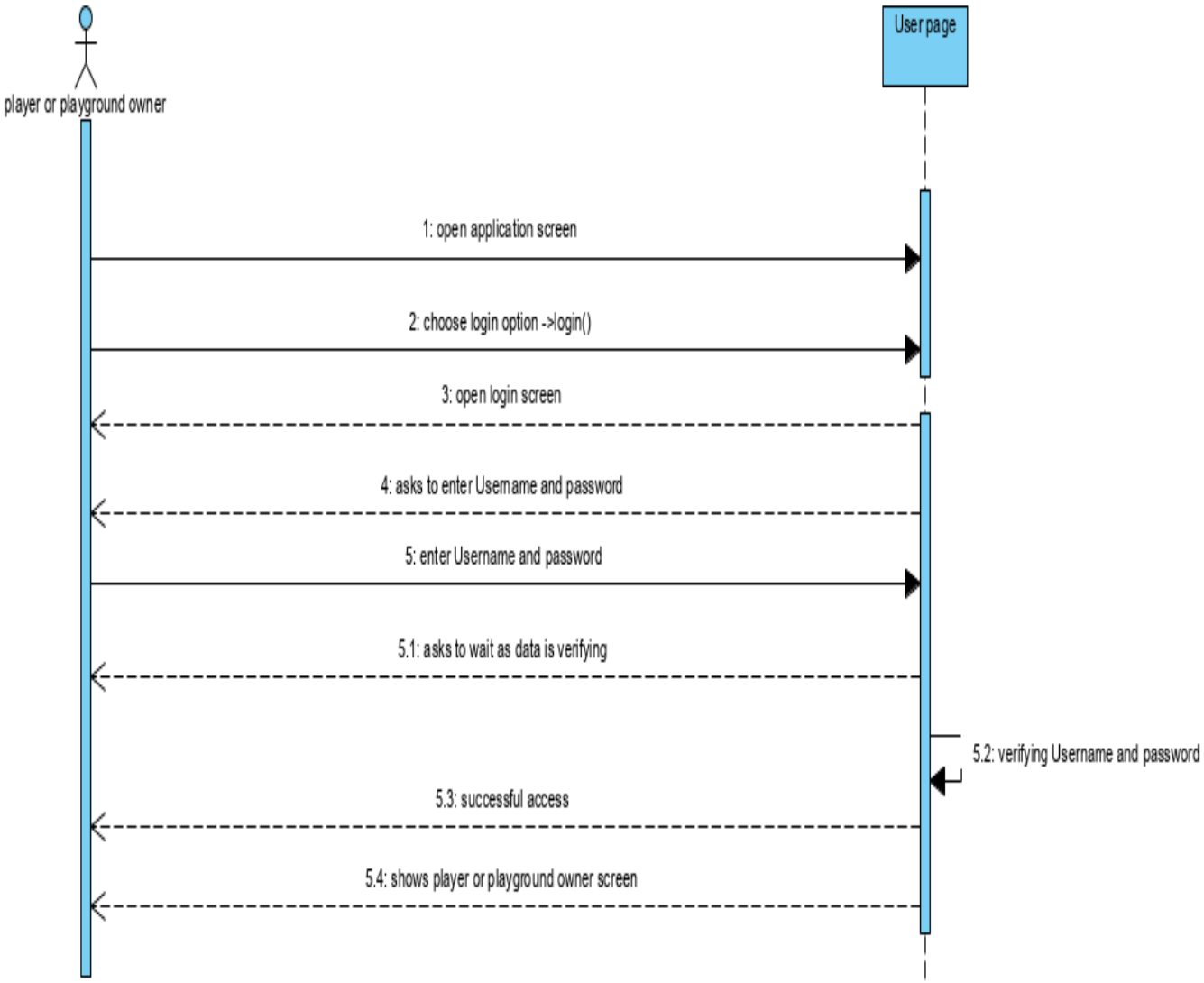
Class ID	Class Name	Description & Responsibility
1.2	Player	And the main target of this class is achieve anything that the user can do. Her lets talk in more details . Class components: (1)attributes: -player name: used to edit player information -player age: used to edit player information -player role: used to edit player information -emails[]: used in send invitation function and its target is to store players e-mails. -team members[]:used in createteam() function to store team members name -team leader: used in createteam() function to store the leader name. -time until reservation hour:used in cancel booking function -adm: used in booking a playground to show the unreserved plaugrounds. -playground size: used in booking playground to allow the player to choose the capacity of the playground. -playground type: used in booking playground to allow the player to choose the playground type. (2)methods: -booking playground():used to book playground by using object from class administrator and call the method playground list() and choose one to reserve - send invitation(): used to send invitation to friends on the system by using the array e-mails[] and take e-mails from it then send the invitation via e-mail -create team():used to create team by the attributes team members[] and team leader so the user store team members and team leader in the array then they will be stored on the database system - cancel booking():its role is to check if the time until the reservation hour is more than one hour ,if so the information of the booked hour will be deleted from the database -edit player information(): used to store player name,age and role.

Class ID	Class Name	Description & Responsibility
1.3	Playground owner	This class responsibility is to allow playground owner to create profile on the system,add playground with its details(capacity,type,location,price per hour),update the playground information and check the money in his electronic wallet. Class attributes: (1) Name (2) Address (3) E-mail (4) Age And this attributes used to create profile (5) playground size (6) playground type (7) playground location (8) and this attributes used to specify the playground information (9) deposit value (10) withdraw value (11)total money and this attributes used to compute the money in the wallet (12) adm:used to add the playground to the playground list in the administrator class. Class methods: (1) Create profile():used to create personal profile for the owner and store the entered data in the database system (2) Add playground(): this function is responsible for initializing a new playground with its details(capacity,type,location,price per hour) and it will be added to the playground list in the administrator class. (3) Update playground information():responsible for updating playground data (4) Check wallet():responsible for computing the total money in the electronic wallet by subtracting the withdraw value from the deposit value and store the result in the total money attribute.

Class ID	Class Name	Description & Responsibility
1.4	Administrator	This class is responsible for controlling the players and playground owners ,organizing hours and specifying the hours that will be played and the hours wont be played and also behave carefully with any infraction steps or behaviours . Class attributes: (1) Name (2) Address (3) E-mail And this attributes used to create profile (4) Playground quality:used to specify the playground quality (5) Time until reservation hour:used in deleting playgrounds Class methods: (1) Create profile():used to create personal profile for the owner and store the entered data in the database system (2) approve playground():used to approve playgrounds by checking the quality attribute if it is true the playground will be approved and get inside the playgrounds list,if it false the playground will be rejected (3) activate playground(): here all we will do is to call the function approve playground() and if it returns true the playground will be active and if not it wont be active. (4) playground list():view all reserved playgrounds with their time. (5)suspend playground():used to suspend playground when checking the playground list and if found any playground more than one time with the same reservation time it will be suspend. (6)update playgrounds():used to update playground status if it is played or suspended or deleted etc. (7)delete playground():it is responsible for store player opinion about the playground and if the player complained then the playground will be deleted (8) view unreserved playgrounds():all what this function do is to call playgrounds list() and check playground is reserved or not if not it will be shown on the screen.

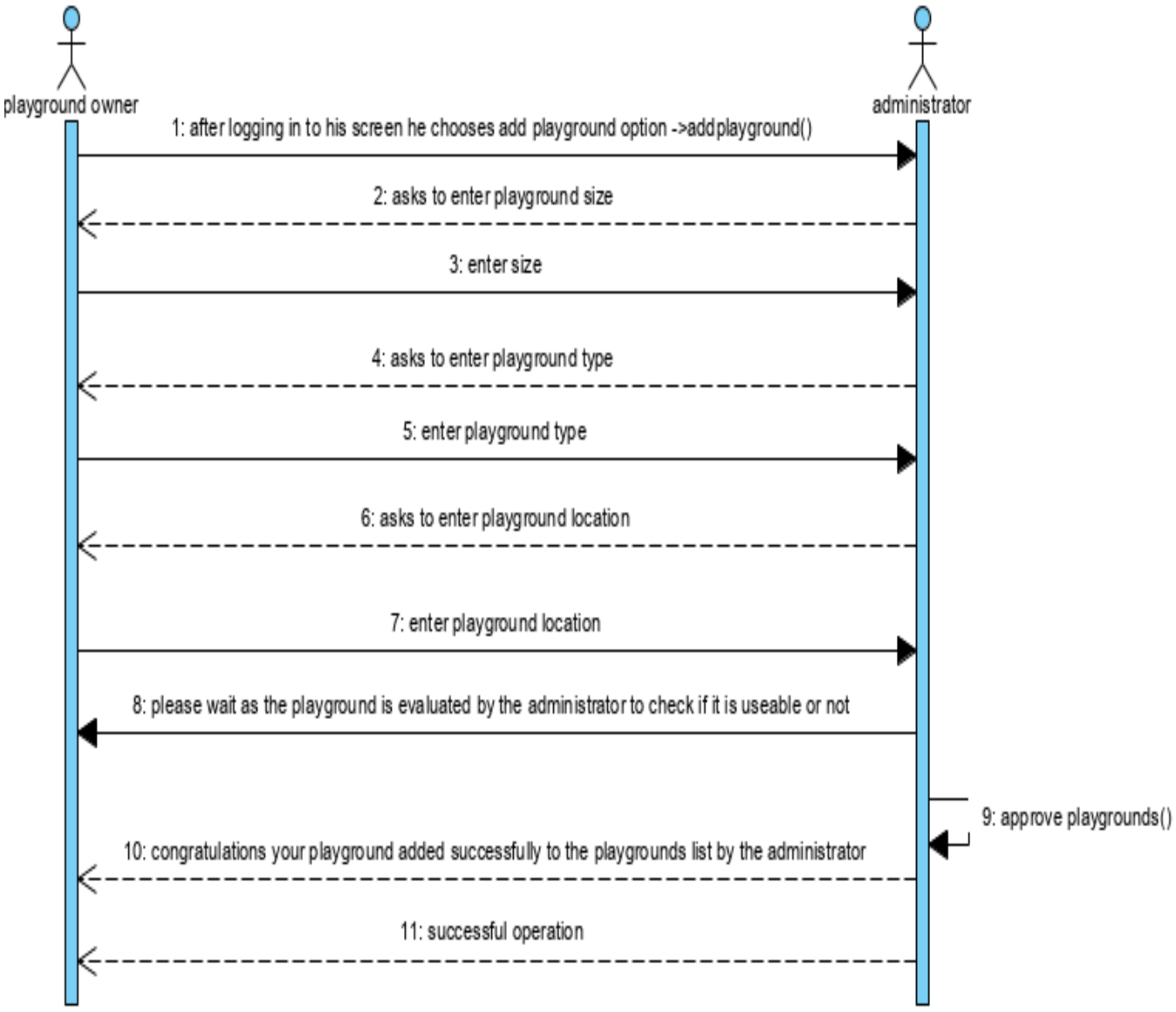
Software Design Specification

Sequence diagram for registering on the system



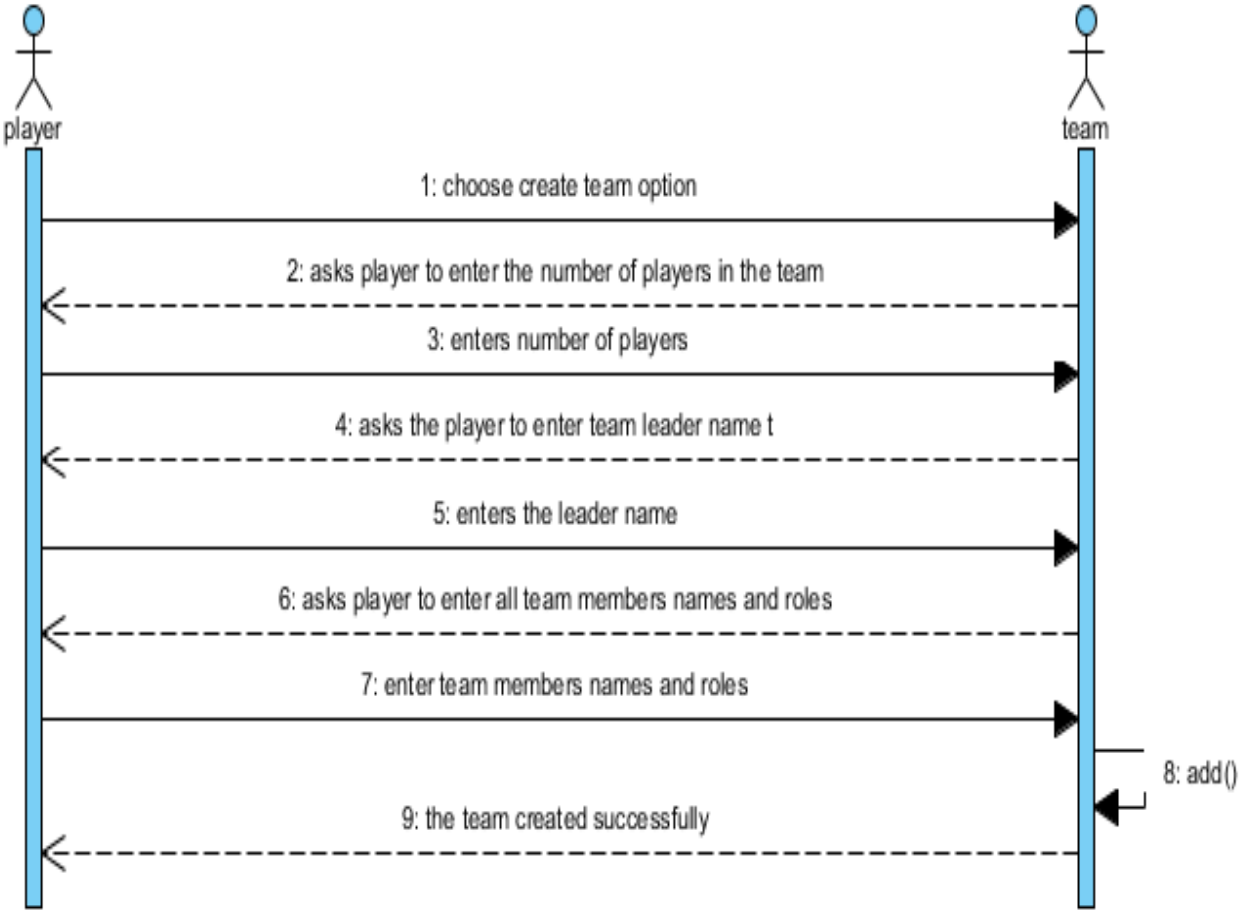
Software Design Specification

Sequence diagram for adding a playground on the system



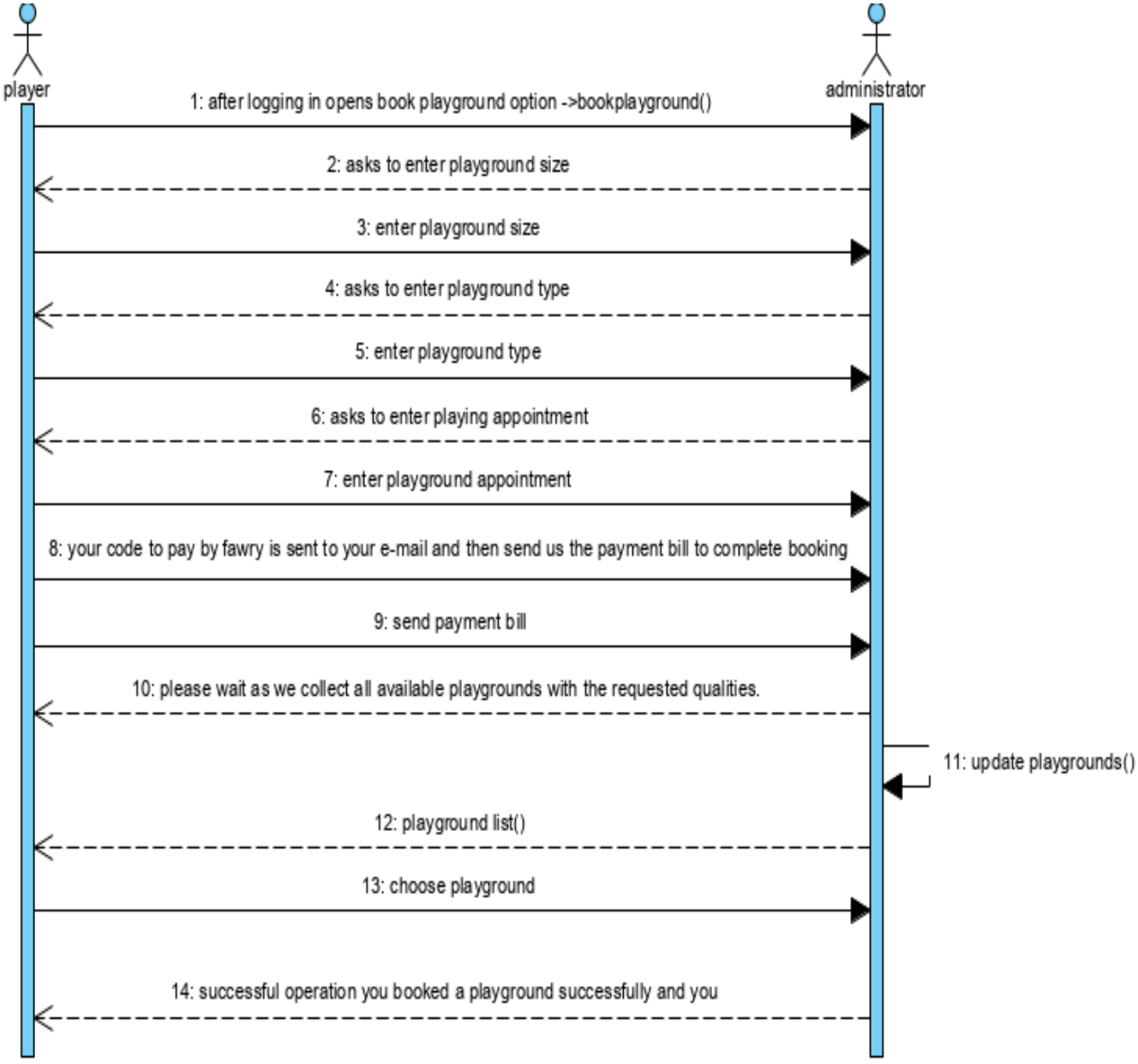
Software Design Specification

Sequence diagram for creating team on the system



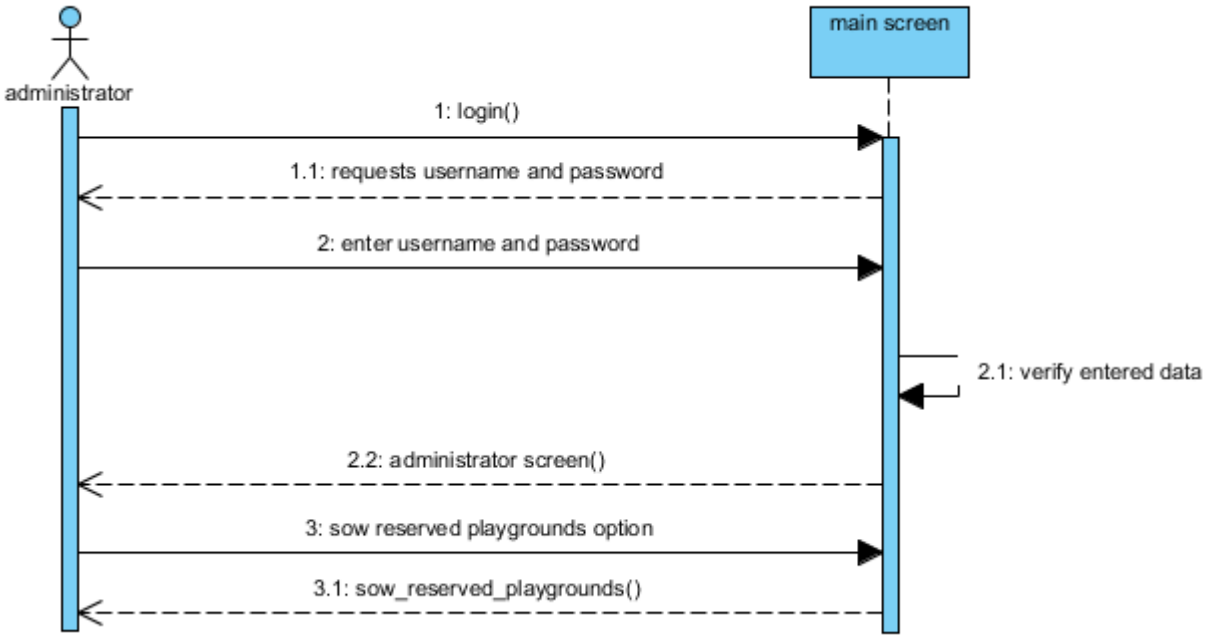
Software Design Specification

Sequence diagram for booking a playground on the system



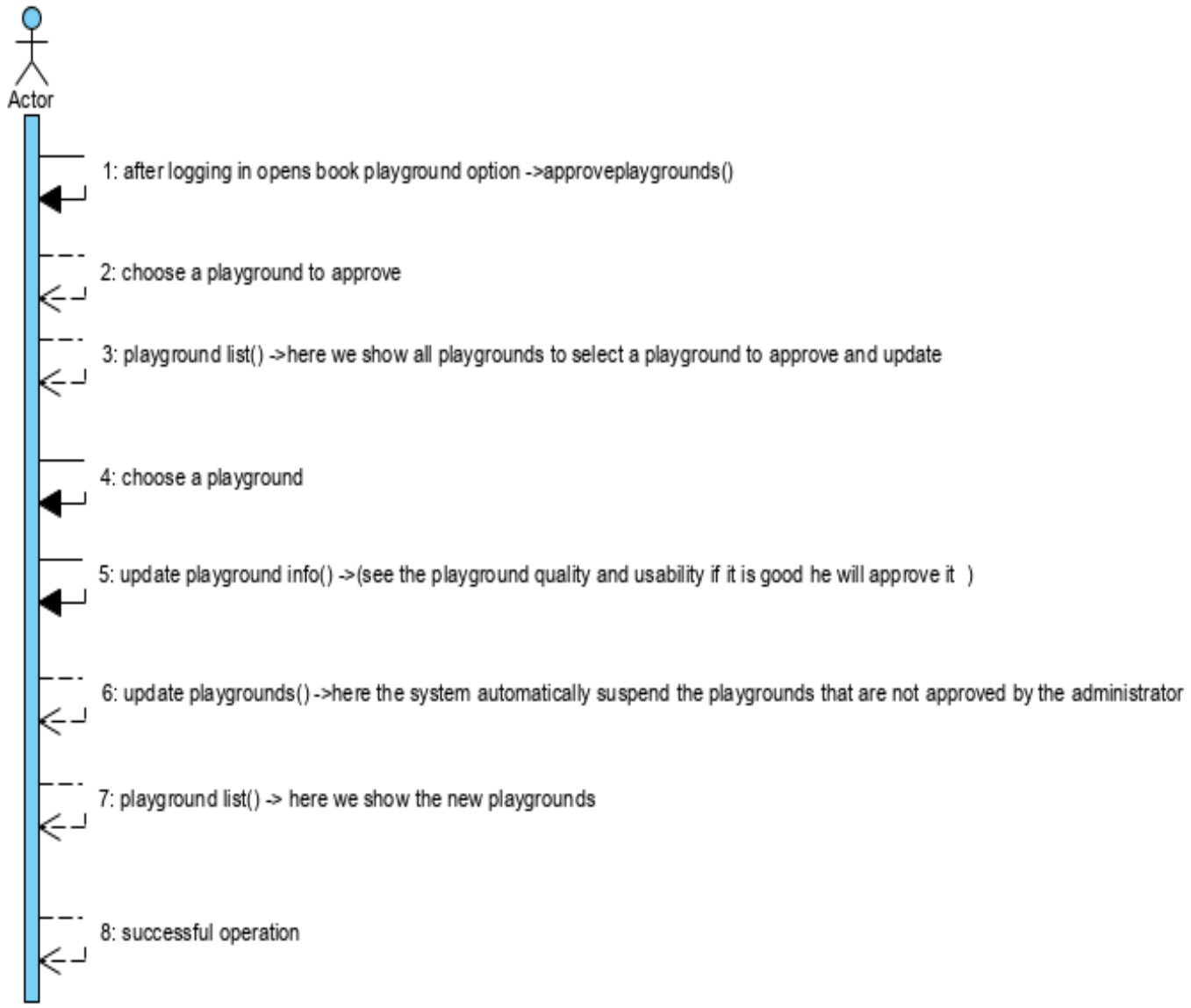
Software Design Specification

Sequence diagram for viewing playing hours on the system



Software Design Specification

Sequence diagram for approving a playground on the system Here the actor is administrator



Software Design Specification

Class - Sequence Usage Table

Sequence Diagram	Classes Used	All Methods Used
1- register on the system	Player or playground owner	Login()
2- add playground	• Playground owner • Administrator	• addplayground() • ApprovePlayground()
3-booking playground	• Player • administrator	• bookingPlayground() • updatePlayground() • playgroundList()
4-approvePlayground	• administrator	• approvePlayground() • updatePlayground() • playgroundlist() • updatePlaygroundInformation()
5-create team	• team • player	• add()
6-viewing playing hours	• administrator	• login() • show_reserved_playgrounds()

Software Design Specification

Implementation classes code&test

i. *playground owner class:*

```
package play;
import java.util.*;
public class PlaygroundOwner {
    static String name,address,email,playground_type,playground_location;
    static int playground_capacty,available_slot;
    static double price_per_hour,account_palance;
    static ArrayList<String>PLAYGROUND_TYPE=new ArrayList< String>();
    static ArrayList<String>PLAYGROUND_LOCATION=new ArrayList< String>();
    static ArrayList<Integer>AVAILABLE_SLOTS=new ArrayList< Integer>();
    static ArrayList<Double>PRICE_PER_HOUR=new ArrayList<Double>();
    static ArrayList<Integer>PLAYGROUND_CAPACITY=new ArrayList< Integer>();
    static ArrayList<String>OWNERSNAME=new ArrayList< String>();
    static ArrayList<String>OWNERSADDRESS=new ArrayList< String>();
    static ArrayList<String>OWNERSEMAILS=new ArrayList< String>();
    static ArrayList<Double>OWNER_PALANCE=new ArrayList<Double>();


    static administrator a1=new administrator();

    public static void set_palance(double palance)
    {
    account_palance=palance;
    }
    public static double get_palance()
    {
    return account_palance;
    }
```

```
public static void AddPlyground(){

    PlaygroundOwner o4=new PlaygroundOwner();
    System.out.println("enter your name:");
    Scanner nam=new Scanner(System.in);
    name=nam.nextLine();
    System.out.println("enter your address:");
    Scanner add=new Scanner(System.in);
    address=add.nextLine();
    System.out.println("enter your email:");
    Scanner mail=new Scanner(System.in);
    email=mail.nextLine();
    System.out.println("enter plyground type:");
    Scanner input=new Scanner(System.in);
    playground_type=input.nextLine();
    System.out.println("enter plyground location:");
    Scanner loc=new Scanner(System.in);
    playground_location=loc.nextLine();
    System.out.println("enter plyground capacity:");
    Scanner cap=new Scanner(System.in);
    playground_capacty=cap.nextInt();
    System.out.println("enter the price per hour:");
    Scanner p=new Scanner(System.in);
    price_per_hour=p.nextDouble();
    System.out.println("enter the available time slot:");
    Scanner A=new Scanner(System.in);
    available_slot=A.nextInt();
    if(a1.approve()==true){
    OWNERSNAME.add(name);
    OWNERSADDRESS.add(address);
    OWNERSEMAILS.add(email);
    PLAYGROUND_TYPE.add(playground_type);
    PLAYGROUND_LOCATION.add(playground_location);
    PLAYGROUND_CAPACITY.add(playground_capacty);
    AVAILABLE_SLOTS.add(available_slot);
    PRICE_PER_HOUR.add(price_per_hour);}
```

```
  if(a1.approve()==true){
  OWNERSNAME.add(name);
  OWNERSADDRESS.add(address);
  OWNERSEMAILS.add(email);
  PLAYGROUND_TYPE.add(playground_type);
  PLAYGROUND_LOCATION.add(playground_location);
  PLAYGROUND_CAPACITY.add(playground_capacty);
  AVAILABLE_SLOTS.add(available_slot);
  PRICE_PER_HOUR.add(price_per_hour);}
  else{
      System.out.println("SORRY WE DON'T SUPPROT THIS kIND OF PLAYGROUNDS YOU SHOULD FOLLOW OUR INSTRUCTIONS TO ADD A PLAYGROUND ON OUR SYSTEM AND OUR CONDITIONS ARE "
              + ":[1] playground_type=(GRASS OR ASPHALT) [2]price_per_hour<=150.0 [3]playground_capacty<=7");
  }



blic static void showallplaygrounds(){
  for(int i=0;i<OWNERSNAME.size();i++){
      System.out.println("playground type:"+" "+PLAYGROUND_TYPE.get(i)+"  "+"playground location:"+" "+PLAYGROUND_LOCATION.get(i)+
              "  "+"playground capacity:"+" "+PLAYGROUND_CAPACITY.get(i)+"  "+"available slot:"+" "+AVAILABLE_SLOTS.get(i)+"  "+"hourly price:"+PRICE_PER_HOUR.get(i)+
              "ownername:"+" "+OWNERSNAME.get(i) +"  "+"owneraddress"+" "+ OWNERSADDRESS.get(i) +"  "+"owneremail"+" "+OWNERSEMAILS.get(i));
  }




public static void showallplaygrounds(){
    for(int i=0;i<OWNERSNAME.size();i++){
        System.out.println("playground type:"+" "+PLAYGROUND_TYPE.get(i)+"  "+"playground location:"+" "+PLAYGROUND_LOCATION.get(i)+
                "  "+"playground capacity:"+" "+PLAYGROUND_CAPACITY.get(i)+"  "+"available slot:"+" "+AVAILABLE_SLOTS.get(i)+"  "+"hourly price:"+PRICE_PER_HOUR.get(i)+
                "ownername:"+" "+OWNERSNAME.get(i) +"  "+"owneraddress"+" "+ OWNERSADDRESS.get(i) +"  "+"owneremail"+" "+OWNERSEMAILS.get(i));
    }

}

public static void create_profile(){

    System.out.println("enter your name:");
    Scanner nam=new Scanner(System.in);
    name=nam.nextLine();
    System.out.println("enter your address:");
    Scanner add=new Scanner(System.in);
    address=add.nextLine();
    System.out.println("enter your email:");
    Scanner mail=new Scanner(System.in);
    email=mail.nextLine();
    OWNERSNAME.add(name);
    OWNERSADDRESS.add(address);
    OWNERSEMAILS.add(email);
}
public static void showowners(){
    for(int i=0;i<OWNERSNAME.size();i++){
        System.out.println("ownername:"+" "+OWNERSNAME.get(i)+"  "+"owneraddress"+" "+ OWNERSADDRESS.get(i) +"  "+"owneremail"+" "+OWNERSEMAILS.get(i)
                +"  "+"ownerpalance:"+OWNER_PALANCE.get(i));
    }
}

}
```

Software Design Specification

ii. player class

```
package play;

import java.util.ArrayList;

public class player {

    static ArrayList<team>TEAMS=new ArrayList< team>();
    static ArrayList<playgrounds>BOOKED=new ArrayList();
    static ArrayList<String>PLAYERSNAME=new ArrayList< String>();
    static ArrayList<String>PLAYERSADDRESS=new ArrayList< String>();
    static ArrayList<String>PLAYERSEMAILS=new ArrayList< String>();
    static ArrayList<Double>PLAYERPALANCE=new ArrayList< Double>();

    static administrator a5=new administrator();
    static String name,address,email,teamleader;
    public static void create_profile(){

        System.out.println("enter your name:");
        Scanner nam=new Scanner(System.in);
        name=nam.nextLine();
        System.out.println("enter your address:");
        Scanner add=new Scanner(System.in);
        address=add.nextLine();
        System.out.println("enter your email:");
        Scanner mail=new Scanner(System.in);
        email=mail.nextLine();
        PLAYERSNAME.add(name);
        PLAYERSADDRESS.add(address);
        PLAYERSEMAILS.add(email);
    }
```

```
public static void create_team(){
    int no;
    player p4=new player();
    String nam,rol;
    System.out.println("enter number of players in your team:");
    Scanner num=new Scanner(System.in);
    no=num.nextInt();
    System.out.println("enter your name:");
    Scanner creator=new Scanner(System.in);
    teamleader=creator.nextLine();
    for(int i=0;i<no;i++){
        System.out.println("enter name of player #"+(i+1)+" in your team:");
        Scanner NAME=new Scanner(System.in);
        nam=NAME.nextLine();
        System.out.println("enter role of player #"+(i+1)+" in your team:");
        Scanner ROLE=new Scanner(System.in);
        rol=ROLE.nextLine();
        team t1=new team(nam,rol,p4);
        TEAMS.add(t1);

    }

}

public static void show_players(){
    for(int i=0;i<PLAYERSNAME.size();i++){
        System.out.println("playername:"+" "+PLAYERSNAME.get(i) +"  "+"playeraddress"+" "+ PLAYERSADDRESS.get(i) +"  "+
    "playeremail"+" "+PLAYERSEMAILS.get(i)+"  "+"player palance:"+PLAYERPALANCE.get(i));
    }

}
public static void show_teams(){
    System.out.println("TEAM LEADER AND CREATOR IS:"+" "+teamleader);
    for(int i=0;i<TEAMS.size();i++){
        System.out.println(TEAMS.get(i).tostring() );
    }

}
```

```
public static void book_playground()
{       PlaygroundOwner p5=new PlaygroundOwner();
    player p3=new player();
    String playground_type,location;
    int playground_capacity,slot;
    double price_per_hour,total,palance=500.0;
    System.out.println("enter the type of the playground you want to book(grass or asphalt");
    Scanner TYPE=new Scanner(System.in);
    playground_type=TYPE.nextLine();
    System.out.println("enter the location of the playground you want to book");
    Scanner LOCATION=new Scanner(System.in);
    location=LOCATION.nextLine();
    System.out.println("enter the capacity of the playground you want to book");
    Scanner CAPACITY=new Scanner(System.in);
    playground_capacity=CAPACITY.nextInt();
    System.out.println("enter the time slot that you wnt to play in the playground you want to book");
    Scanner SLOT=new Scanner(System.in);
    slot=SLOT.nextInt();
    System.out.println("enter the hourly price of the playground you want to book");
    Scanner HOURLY_PRICE=new Scanner(System.in);
    price_per_hour=HOURLY_PRICE.nextInt();
    total=palance-price_per_hour;
    p5.set_palance(price_per_hour);
    p5.OWNER_PALANCE.add(price_per_hour);
    PLAYERPALANCE.add(total);
    boolean check1=p5.PLAYGROUND_TYPE.contains(playground_type);
    boolean check2=p5.PLAYGROUND_LOCATION.contains(location);
    boolean check3=p5.PLAYGROUND_CAPACITY.contains( playground_capacity);
    boolean check4=p5.PRICE_PER_HOUR.contains(price_per_hour);
    boolean check5=p5.AVAILABLE_SLOTS.contains(slot);
    boolean check1=p5.PLAYGROUND_TYPE.contains(playground_type);
    boolean check2=p5.PLAYGROUND_LOCATION.contains(location);
    boolean check3=p5.PLAYGROUND_CAPACITY.contains( playground_capacity);
    boolean check4=p5.PRICE_PER_HOUR.contains(price_per_hour);
    boolean check5=p5.AVAILABLE_SLOTS.contains(slot);
    if(check1==true &&  check2==true && check3==true && check4==true && check5==true)
    {
        System.out.println("the suitable playgrounds to the entered specifications:");
        for(int i=0;i<1;i++){
            System.out.println("type:"+" "+playground_type+" " +"location:"+" "+ location +" "+"capacity:"+" "+
        playground_capacity +" "+"hourly price:"+" "+price_per_hour+" "+"available time slot"+" "+slot+"  "+"your palance became:"+PLAYERPALANCE.get(i));

        }
        playgrounds p0=new playgrounds(p5.playground_type,p5.playground_location,p5.playground_capacty,p5.price_per_hour,p5.available_slot);
        BOOKED.add(p0);
    }
    else{System.out.println("SORRY NO PLAYGROUND MATCHES WITH THE ENTERED SPECIFICATIONS!");}


}
//public static void main(String []args){
//  create_team();
//  show_teams();

//  }
}
```

Software Design Specification

iii. administrator class

```
package play;

import java.util.ArrayList;

public class administrator {



    public static PlaygroundOwner p8=new PlaygroundOwner();
    public static player p3=new player();
    public static boolean approve(){
        if((p8.playground_type=="grass" && p8.playground_type=="asphalt") && p8.price_per_hour<=150.0 && p8.playground_capacty<=7)
        {
return true ;   }
        else{
return false;       }

    }
    public static void show_reserved_playgrounds(){
        if(p3.BOOKED.isEmpty()){
            System.out.println("sorry there is no playgrounds booked yet to be shown");
        }
        else{
        for(int i=0;i<p3.BOOKED.size();i++){
            System.out.println("type:"+p3.BOOKED.get(i).tostring());
        }
        }
    }

public static void show_teams(){
    System.out.println("TEAM LEADER AND CREATOR IS:"+" "+p3.teamleader);
    for(int i=0;i<p3.TEAMS.size();i++){
        System.out.println(p3.TEAMS.get(i).tostring() );
    }

}
```

```
public static void show_teams(){
    System.out.println("TEAM LEADER AND CREATOR IS:"+" "+p3.teamLeader);
    for(int i=0;i<p3.TEAMS.size();i++){
        System.out.println(p3.TEAMS.get(i).tostring() );
    }

}
public static void show_players(){
    for(int i=0;i<p3.PLAYERSNAME.size();i++){
        System.out.println("playername:"+" "+p3.PLAYERSNAME.get(i) +"  "+"playeraddress"+" "+ p3.PLAYERSADDRESS.get(i)
                +"  "+"playeremail"+" "+p3.PLAYERSEMAILS.get(i)+"  "+"player palance:"+p3.PLAYERPALANCE.get(i));
    }
}
    public static void showallplaygrounds(){
        for(int i=0;i<p8.OWNERSNAME.size();i++){
            System.out.println("playground type:"+" "+p8.PLAYGROUND_TYPE.get(i)+"  "+"playground location:"+
        " "+p8.PLAYGROUND_LOCATION.get(i)+"  "+"playground capacity:"+" "+p8.PLAYGROUND_CAPACITY.get(i)+"  "+"available slot:"+
                    " "+p8.AVAILABLE_SLOTS.get(i)+"  "+"hourly price:"+p8.PRICE_PER_HOUR.get(i)+"ownername:"+" "+p8.OWNERSNAME.get(i)
                   +"  "+"owneraddress"+" "+ p8.OWNERSADDRESS.get(i) +"  "+"owneremail"+" "+p8.OWNERSEMAILS.get(i));
        }
    }

    public static void show_owners(){
        for(int i=0;i<p8.OWNERSNAME.size();i++){
            System.out.println("ownername:"+" "+p8.OWNERSNAME.get(i) +"  "+"owneraddress"+" "+ p8.OWNERSADDRESS.get(i) +"  "+"owneremail"+" "+
        p8.OWNERSEMAILS.get(i)+"  "+"ownerpalance:"+p8.OWNER_PALANCE.get(i));
        }
    }

}
```

Software Design Specification

iv. playgrounds class

```
package play;

public class playgrounds {
    static String playground_type,playground_location;
     static int playground_capacity;
    static double price_per_hour,available_time;
    public playgrounds( String playground_type,String playground_location,int playground_capacity,double price_per_hour,double available_time){
        this.playground_type=playground_type;
        this.playground_capacity=playground_capacity;
        this.playground_location=playground_location;
        this.price_per_hour=price_per_hour;
        this.available_time=available_time;
    }
    public static void set_type(String type){
        playground_type=type;
    }
    public static void set_location(String location){
        playground_location=location;
    }
    public static void set_capacity(int capacity){
        playground_capacity=capacity;
    }
    public static void set_hourlyPrice(double hourly_price){
        price_per_hour=hourly_price;
    }
    public static void set_available_hours(double available){
        available_time=available;
    }

    public String get_type(){
        return playground_type;
    }
    public String get_location(){
        return playground_location;
    }
    public int get_capacity(){
        return playground_capacity;
    }
public String get_type(){
    return playground_type;
}
public String get_location(){
    return playground_location;
}
public int get_capacity(){
    return playground_capacity;
}
public double get_hourlyprice(){
    return price_per_hour;
}
public String tostring()
{
return "type:"+" "+playground_type+" " +"location:"+" "+ playground_location +" "+"capacity:"+" "+ playground_capacity +" "+
"hourly price:"+" "+price_per_hour+" "+"reserved time slot"+" "+available_time;
}
```

Software Design Specification

v. team class

```
package play;

public class team {
    String name,role;
    player p0=new player();
    public team(String name,String role,player p4){
        p0.name=p4.name;
        this.name=name;
        this.role=role;
    }

    public void set_name(String name){
        this.name=name;
    }
    public void set_role(String role){
        this.role=role;
    }
    public String get_name(){
        return name;
    }
    public String get_role(){
        return role;
    }
    public String tostring(){
        return "name:"+" "+name+"  "+"role:"+" "+role;
    }

}
```

Software Design Specification

vi. main class

```
package play;
import java.util.*;

public class p {

    static ArrayList<String>FIRSTNAME=new ArrayList<String>();
    static ArrayList<String>LASTNAME=new ArrayList<String>();
    static ArrayList<String>USERNAME=new ArrayList<String>();
    static ArrayList<String>PASSWORD=new ArrayList<String>();
    static ArrayList<String>LOGINPASSWORD=new ArrayList<String>();
    static ArrayList<String>LOGINUSERNAME=new ArrayList<String>();
    static ArrayList<String>OWNERSPASSWORD=new ArrayList<String>();
    static ArrayList<String>PLAYERPASSWORD=new ArrayList<String>();
    static ArrayList<String>ADMINUSERNAME=new ArrayList<String>();
    static ArrayList<String>ADMINPASSWORD=new ArrayList<String>();
    static administrator a4=new administrator();
    static PlaygroundOwner o=new PlaygroundOwner();
    static player p6=new player();
    //static profile p3=new profile(o);
    static String first,last,username;
    static String password;
    public static void register()
    {
        System.out.println("please enter first name :");
        Scanner fname=new Scanner(System.in);
        first=fname.nextLine();
        System.out.println("please enter last name :");
        Scanner lname=new Scanner(System.in);
        last=lname.nextLine();
        System.out.println("please enter username :");
        Scanner uname=new Scanner(System.in);
        username=uname.nextLine();
        System.out.println("please enter password :");
        Scanner pass=new Scanner(System.in);
        password=pass.nextLine();
        FIRSTNAME.add(first);
        LASTNAME.add(last);
```

```
        FIRSTNAME.add(first);
        LASTNAME.add(last);
        USERNAME.add(username);
        PASSWORD.add(password);
        boolean check1=LOGINUSERNAME.contains(username);
        boolean check2=LOGINPASSWORD.contains(password);
        if(check1 ==true && check2==true)
        {
            System.out.println("sorry this username and password is alredy logged in ");
        }
        else
        {
            System.out.println("congratulations you had registered successfully");
            LOGINUSERNAME.add(username);
            LOGINPASSWORD.add(password);

        }


}
```

```
public static void login(){
    String adminName="podo";
    String AdminPassword="podo&mala";
    ADMINUSERNAME.add(adminName);
    ADMINPASSWORD.add(AdminPassword);
    String username,password;
    System.out.println("please enter username :");
    Scanner uname=new Scanner(System.in);
    username=uname.nextLine();
    System.out.println("please enter password :");
    Scanner pass=new Scanner(System.in);
    password=pass.nextLine();

        boolean user=LOGINUSERNAME.contains(username);
        boolean passw=LOGINPASSWORD.contains(password);
        boolean check2=ADMINUSERNAME.contains(username);
        boolean check3=ADMINPASSWORD.contains(password);
        if(user ==true && passw==true)
        {
            System.out.println("congratulations you logged in successfully !!!");


            boolean role=PLAYERPASSWORD.contains(password);
            boolean rol=OWNERSPASSWORD.contains(password);

            if(role ==true){
                int CHOICE;
                String option;
                System.out.println("you are player!!");
                do{
                    System.out.println("[1] book playground [2]create profile [3]create team [4]show teams [5]show all players");
                    Scanner input=new Scanner(System.in);
                    CHOICE=input.nextInt();
                    if(CHOICE==1){
                        p6.book_playground();
                    }
                    else if(CHOICE==2){
```

```
        else if(CHOICE==2){
            p6.create_profile();
        }
        else if(CHOICE==3){
            p6.create_team();
        }
        else if(CHOICE==4){
            p6.show_teams();
        }
        else if(CHOICE==5){
            p6.show_players();
        }
        else{
            System.out.println("wrong choice please choose a right number ");
            CHOICE=input.nextInt();
        }

        System.out.println("want to perform another option?(yes/no)");
        Scanner OPTION=new Scanner(System.in);
        option=OPTION.nextLine();
        }while(option=="yes");

}
 if(rol==true){System.out.println("you are playground owner!!");
 int enter;
 String option;
    do{
        System.out.println("[1] add playground [2]create profile [3]show all playgrounds [4] show playgrounds owners");
        Scanner input=new Scanner(System.in);
        enter=input.nextInt();
        if(enter==1){
    o.AddPlyground();

        }
        else if(enter==2){
        o.create_profile();

        }
```

```
            else if(enter==3){
o.showallplaygrounds();}

            else if(enter==4){
o.showowners();}
            else{
                System.out.println("wrong choice please choose a right number ");
                enter=input.nextInt();
            }

            System.out.println("want to perform another option?(yes/no)");
            Scanner OPTION=new Scanner(System.in);
            option=OPTION.nextLine();
            }while(option =="yes");
        }



}
 else if(check2==true && check3==true){

        System.out.println("WELCOME ADMINISTRATOR:"+adminName);
        System.out.println("[1]show the booked playgrounds [2]show owners "
                + " [3]show players [4]show teams [5]show all playgrounds ");

        Scanner inter=new Scanner(System.in);
        int optional=inter.nextInt();
        if(optional==1){
            a4.show_reserved_playgrounds();
        }
        else if(optional==2){
            a4.show_owners();
        }
        else if(optional==3){
            a4.show_players();
        }
```

```
                else if(optional==4){
                    a4.show_teams();
                }
                else if(optional==5){
                    a4.showallplaygrounds();
                }
                else{
                    System.out.println("wrong choice: CHOOSE A CORRECT NUMBER ");
                }
         }

        else
        {
            System.out.println("username or password is wrong !");

        }

    }

public static void main(String []args){
    int choose,mychoice;
    String adminName="podo";
    String AdminPassword="podo and malak";
    LOGINUSERNAME.add(adminName);
    LOGINPASSWORD.add(AdminPassword);

    while(true)
    {
        System.out.println("[1]login or [2]register");
        Scanner choice=new Scanner(System.in);
        choose=choice.nextInt();
        if(choose==1)
        {
            login();
            }
```

```
else if(choose==2){

    System.out.println("register as [1]player or [2]playground owner?");
    Scanner select=new Scanner(System.in);
    mychoice=select.nextInt();
    if(mychoice==1){
        int CHOICE;
        register();
        PLAYERPASSWORD.add(password);

        String option;
        do{
        System.out.println("[1] book playground [2]create profile "
                + "[3]create team [4]show teams [5]show all players");
        Scanner input=new Scanner(System.in);
        CHOICE=input.nextInt();
        if(CHOICE==1){
            p6.book_playground();
        }
        else if(CHOICE==2){
            p6.create_profile();
        }
        else if(CHOICE==3){
            p6.create_team();
        }
        else if(CHOICE==4){
            p6.show_teams();
        }
        else if(CHOICE==5){
            p6.show_players();
        }
        else{
            System.out.println("wrong choice please choose a right number ");
            CHOICE=input.nextInt();
        }
```

```
            }
            System.out.println("want to perform another option?(yes/no)");
            Scanner OPTION=new Scanner(System.in);
            option=OPTION.nextLine();
            }while(option=="yes");

        }
        else if(mychoice==2){
            int enter;
            String option;
            register();
            OWNERSPASSWORD.add(password);
            do{
            System.out.println("[1] add playground [2]create profile "
                    + "[3]show all playgrounds [4] show playgrounds owners");
            Scanner input=new Scanner(System.in);
            enter=input.nextInt();
            if(enter==1){
        o.AddPlyground();

            }
            else if(enter==2){
            o.create_profile();

            }

            else if(enter==3){
o.showallplaygrounds();}

            else if(enter==4){
o.showowners();}
            else{
                System.out.println("wrong choice please choose a right number ");
                enter=input.nextInt();
            }
```

```
                else if(enter==2){
                o.create_profile();

                }

                else if(enter==3){
    o.showallplaygrounds();}

                else if(enter==4){
    o.showowners();}
                else{
                    System.out.println("wrong choice please choose a right number ");
                    enter=input.nextInt();
                }
                System.out.println("want to perform another option?(yes/no)");
                Scanner OPTION=new Scanner(System.in);
                option=OPTION.nextLine();
                }while(option=="yes");
            }

        }
    }
}

}
```

Software Design Specification

Test console part

```
p [Java Application] C:\Program Files\Java\jre1.8.0_40\bin\javaw.exe (م ٦:٤٩:١٣ ٢٠٢٠/٠٥/٣٠)
[1]login or [2]register
2
register as [1]player or [2]playground owner?
2
please enter first name :
podo
please enter last name :
sayed
please enter username :
abdalla
please enter password :
ragab
congratulations you had registered successfully
[1] add playground [2]create profile [3]show all playgrounds [4] show playgrounds owners
1
enter your name:
podo
enter your address:
monib
enter your email:
podo@gmail.com
enter plyground type:
grass
enter plyground location:
monib
enter plyground capacity:
7
enter the price per hour:
100
enter the available time slot:
8
want to perform another option?(yes/no)
no
[1]login or [2]register
2
```

```
grass
enter plyground location:
monib
enter plyground capacity:
7
enter the price per hour:
100
enter the available time slot:
8
want to perform another option?(yes/no)
no
[1]login or [2]register
2
register as [1]player or [2]playground owner?
2
please enter first name :
ramy
please enter last name :
so
please enter username :
filo
please enter password :
mina
congratulations you had registered successfully
[1] add playground [2]create profile [3]show all playgrounds [4] show playgrounds owners
1
enter your name:
ramy
enter your address:
giza
enter your email:
ramy@gmail.com
enter plyground type:
asphalt
enter plyground location:
giza
```

```
[1] add playground [2]create profile [3]show all playgrounds [4] show playgrounds owners
1
enter your name:
ramy
enter your address:
giza
enter your email:
ramy@gmail.com
enter plyground type:
asphalt
enter plyground location:
giza
enter plyground capacity:
6
enter the price per hour:
120
enter the available time slot:
9
want to perform another option?(yes/no)
no
[1]login or [2]register
2
register as [1]player or [2]playground owner?
2
please enter first name :
malak
please enter last name :
essam
please enter username :
anwer
please enter password :
milo
congratulations you had registered successfully
[1] add playground [2]create profile [3]show all playgrounds [4] show playgrounds owners
1
enter your name:
```

```
enter your name:
malak
enter your address:
warrak
enter your email:
malak@gmail.com
enter plyground type:
grass
enter plyground location:
algamaa
enter plyground capacity:
7
enter the price per hour:
140
enter the available time slot:
11
want to perform another option?(yes/no)
no
[1]login or [2]register
2
register as [1]player or [2]playground owner?
1
please enter first name :
sayed
please enter last name :
sido
please enter username :
dodo
please enter password :
pida
congratulations you had registered successfully
[1] book playground [2]create profile [3]create team [4]show teams [5]show all players
1
enter the type of the playground you want to book(grass or asphalt
grass
```

```
enter the type of the playground you want to book(grass or asphalt
grass
enter the location of the playground you want to book
monib
enter the capacity of the playground you want to book
7
enter the time slot that you wnt to play in the playground you want to book
8
enter the hourly price of the playground you want to book
100
the suitable playgrounds to the entered specifications:
type: grass location: monib capacity: 7 hourly price: 100.0 available time slot 8  your palance became:400.0
want to perform another option?(yes/no)
no
[1]login or [2]register
1
please enter username :
dodo
please enter password :
pida
congratulations you logged in successfully !!!
you are player!!
[1] book playground [2]create profile [3]create team [4]show teams [5]show all players
1
enter the type of the playground you want to book(grass or asphalt
asphalt
enter the location of the playground you want to book
giza
enter the capacity of the playground you want to book
6
enter the time slot that you wnt to play in the playground you want to book
9
enter the hourly price of the playground you want to book
120
the suitable playgrounds to the entered specifications:
type: asphalt location: giza capacity: 6 hourly price: 120.0 available time slot 9  your palance became:400.0
```

```
120
the suitable playgrounds to the entered specifications:
type: asphalt location: giza capacity: 6 hourly price: 120.0 available time slot 9  your palance became:400.0
want to perform another option?(yes/no)
no
[1]login or [2]register
1
please enter username :
podo
please enter password :
podo&mala
WELCOME ADMINISTRATOR:podo
[1]show the booked playgrounds [2]show owners  [3]show players [4]show teams [5]show all playgrounds
1
playground type: grass  playground location: monib  playground capacity: 7  reserved slot: 8  hourly price:100ownername: podo  owneraddress monib  owneremail podo@gmail.com
playground type: asphalt  playground location: giza  playground capacity: 6  reserved slot: 9  hourly price:120ownername: ramy  owneraddress giza  owneremail ramy@gmail.com
[1]login or [2]register
1
please enter username :
podo
please enter password :
podo&mala
WELCOME ADMINISTRATOR:podo
[1]show the booked playgrounds [2]show owners  [3]show players [4]show teams [5]show all playgrounds
2
ownername: podo  owneraddress monib  owneremail podo@gmali.com  ownerpalance:100
ownername: ramy  owneraddress giza  owneremail ramy@gmali.com  ownerpalance:120
ownername: malak  owneraddress warrak  owneremail malak@gmali.com  ownerpalance:0
[1]login or [2]register
2
register as [1]player or [2]playground owner?
2
please enter first name :
ssd
please enter last name :
sdse
```

```
please enter first name :
ssd
please enter last name :
sdse
please enter username :
aew
please enter password :
sderws
congratulations you had registered successfully
[1] add playground [2]create profile [3]show all playgrounds [4] show playgrounds owners
6
wrong choice please choose a right number
2
want to perform another option?(yes/no)
no
[1]login or [2]register
2
register as [1]player or [2]playground owner?
1
please enter first name :
aya
please enter last name :
sayed
please enter username :
so
please enter password :
we
congratulations you had registered successfully
[1] book playground [2]create profile [3]create team [4]show teams [5]show all players
2
enter your name:
PODO
enter your address:
giza
enter your email:
aya@gmail.com
```

```
aya@gmail.com
want to perform another option?(yes/no)
no
[1]login or [2]register
1
please enter username :
so
please enter password :
we
congratulations you logged in successfully !!!
you are player!!
[1] book playground [2]create profile [3]create team [4]show teams [5]show all players
3
enter number of players in your team:
3
enter your name:
PODO
enter name of player #1 in your team:
aya
enter role of player #1 in your team:
straycar
enter name of player #2 in your team:
mina
enter role of player #2 in your team:
definder
enter name of player #3 in your team:
mala
enter role of player #3 in your team:
goaleeper
want to perform another option?(yes/no)
no
[1]login or [2]register
1
please enter username :
so
please enter password :
so
please enter password :
we
congratulations you logged in successfully !!!
you are player!!
[1] book playground [2]create profile [3]create team [4]show teams [5]show all players
4
TEAM LEADER AND CREATOR IS: PODO
name: aya  role: straycar
name: mina  role: definder
name: mala  role: goaleeper
want to perform another option?(yes/no)
no
[1]login or [2]register
1
please enter username :
podo
please enter password :
podo&mala
WELCOME ADMINISTRATOR:podo
[1]show the booked playgrounds [2]show owners  [3]show players [4]show teams [5]show all playgrounds
1
playground type: grass  playground location: monib  playground capacity: 7  reserved slot: 8  hourly price:100ownername: podo  owneraddress monib  owneremail podo@gmail.com
playground type: asphalt  playground location: giza  playground capacity: 6  reserved slot: 9  hourly price:120ownername: ramy  owneraddress giza  owneremail ramy@gmail.com
[1]login or [2]register
1
please enter username :
podo
please enter password :
podo&mala
WELCOME ADMINISTRATOR:podo
[1]show the booked playgrounds [2]show owners  [3]show players [4]show teams [5]show all playgrounds
2
ownername: podo  owneraddress monib  owneremail podo@gmali.com  ownerpalance:100
ownername: ramy  owneraddress giza  owneremail ramy@gmali.com  ownerpalance:120
ownername: malak  owneraddress warrak  owneremail malak@gmali.com  ownerpalance:0
```

```
[1]login or [2]register
1
please enter username :
podo
please enter password :
podo&mala
WELCOME ADMINISTRATOR:podo
[1]show the booked playgrounds [2]show owners  [3]show players [4]show teams [5]show all playgrounds
3
playername: sayed  playeraddress monib  playeremail sayed@gmail.com  player palance:280
playername: aya  playeraddress giza  playeremail aya@gmail.com  player palance:500
[1]login or [2]register
1
please enter username :
podo
please enter password :
podo&mala
WELCOME ADMINISTRATOR:podo
[1]show the booked playgrounds [2]show owners  [3]show players [4]show teams [5]show all playgrounds
4
TEAM LEADER AND CREATOR IS: PODO
name: aya  role: straycar
name: mina  role: definder
name: mala  role: goaleeper
[1]login or [2]register
1
please enter username :
podo
please enter password :
podo&mala
WELCOME ADMINISTRATOR:podo
[1]show the booked playgrounds [2]show owners  [3]show players [4]show teams [5]show all playgrounds
5
playground type: grass  playground location: monib  playground capacity: 7  reserved slot: 8  hourly price:100ownername: podo  owneraddress monib  owneremail podo@gmail.com
playground type: asphalt  playground location: giza  playground capacity: 6  reserved slot: 9  hourly price:120ownername: ramy  owneraddress giza  owneremail ramy@gmail.com
playground type: grass  playground location: algamaa  playground capacity: 7  available slot: 11  hourly price:140ownername: malak  owneraddress warrak  owneremail malak@gmail.com
```

Software Design Specification

- software engineer code of ethics:

case study #2

- here are some mistakes take arm company committed according to software engineer ethics principles in IEEE organization. and we will show the mistakes happened in some principles of the IEEE code of principles.

    1. Public: the company approved the system althouth it was not safe,well tested and it causes a harmful pain to the partners as they contributed with a lot amount of money hopping that the software will be finished successfully.and that cost them a huge amount of money in range 125$ million.
    2. Client and employer: here the software engineers of the company acted in a manner that is in the best of their employer but not for their partners.as they didn't explain the reliability of the system. Also they didn't report to their partners all difficulties and violations that the company team faced.

3. Product:here the company commited a huge mistake as the company team accepted to execute and build the system and they completely know that they are not qualified to build the software.
4. Management:here the company manager didn't promote the the ethical approach to the management of the system development and maintainance.

- And here is an advice from our team : if you are a SW engineer please be careful before executing or building any system or software. Because the mistakes of SW engineers only leads to a very high money loss or lead to death.

"Your health and honesty before money"

Software Design Specification

case study #7

- here are some mistakes that company committed according to software engineer ethics principles in IEEE organization. and we will show the mistakes and ethical behaviours happened in some principles of the IEEE code of principles.
    1. Public: here the employee accepted the full responsibility for his own work. Because as soon as he discovered the security vulnerability that attackers could exploit to steal user's personal data and then he told the management team.
    2. Client and employer: here we find that the employee report to his managers about the violations that he had discovered. Also he didn't choose to hush but he tried to solve the problem and found out that it can't be solved individually.
    3. Product: unfortunately companies didn't know if they are qualified to execute and build systems or not till they try to build it. Also unfortunately this company was not qualified to build this system efficiently as attackers can discover a cavity from

which they can steal customers personal information.

4. Judgement: here the employee maintained the

5. integrity and independence in his professional judgement ; as he reported the management team .
6. Management: in this principle the company manager make a great mistake and ethical crime in SW engineering code of ethics; as he ignored the employee discovery and didn't care about him. And his only care was only on earning money not more.

Thanks a lot for your time .

Dr.el-ramly

Software Design Specification

Tools :

- visual paradigm.
- Eclipse ide
- Microsoft office 2013
- Argo(UML)

Ownership report:

Item	Owners
Abdelrahman sayed Abdallah	class description , sequence diagram and sequence usage table. screenshots and writing code
Malak essam anwer	Class diagram and case studies of Ethics,screenshots and writing code

Software Design Specification

Refrences:

Software requirement specifications.

Authors :

(1)Abdelrahman sayed abdallah
(2)malak essam anwer