



CS251

Software Engineering I

GoFo

Software Requirements Specifications

Version 2.0

Project ID: PM-1179

Salah Mohamed	salah20487@gmail.com	20180138
Yasser Eid	yassereid247@gmail.com	20180334
Amr Ayman	amr197659@gmail.com	20180186

3/6/2020



CS251: Phase 2 – <Team Name>

Project: <Project Name>

Software Design Specification

Contents

1. Team	3
2. Document Purpose and Audience	3
3. System Models	3
I. Class Diagram(s).....	3
II. Class Descriptions	5
III. Sequence diagrams.....	5
Class - Sequence Usage Table.....	11
IV. User Interface Design	14
4. Tools	21
5. Ownership Report.....	22
6. References	22
Appendix A: Code Listing and Screen Snapshots.....	23
Authors	63



CS251: Phase 2 – <Team Name>

Project: <Project Name>

Software Design Specification

1. Team

ID	Name	Email	Mobile
20180138	Salah Mohamed Salah	Salah204087@gmail.com	01143987710
20180334	Yasser Eid	yassereid247@gmail.com	01093046291
20180186	Amr Ayman	amr197659@gmail.com	01142659554

2. Document Purpose and Audience

The class diagram is a central modeling technique that runs through nearly all objectoriented methods. This diagram describes the types of objects in the system and various kinds of static relationships which exist between them. There are three principal kinds of relationships which are important: associations (User and owner), subtypes (playground) and aggregation (a Booking and ewallet). The various OO methods all use different (and often conflicting) terminology for these concepts, this is extremely frustrating but inevitable: OO languages are just as inconsiderate. It is in this area that the UML will bring some of its greatest benefits in simplifying these different diagrams. In this section I will use the UML terms as my main terminology, and relate to other terms as I go along

3. System Models

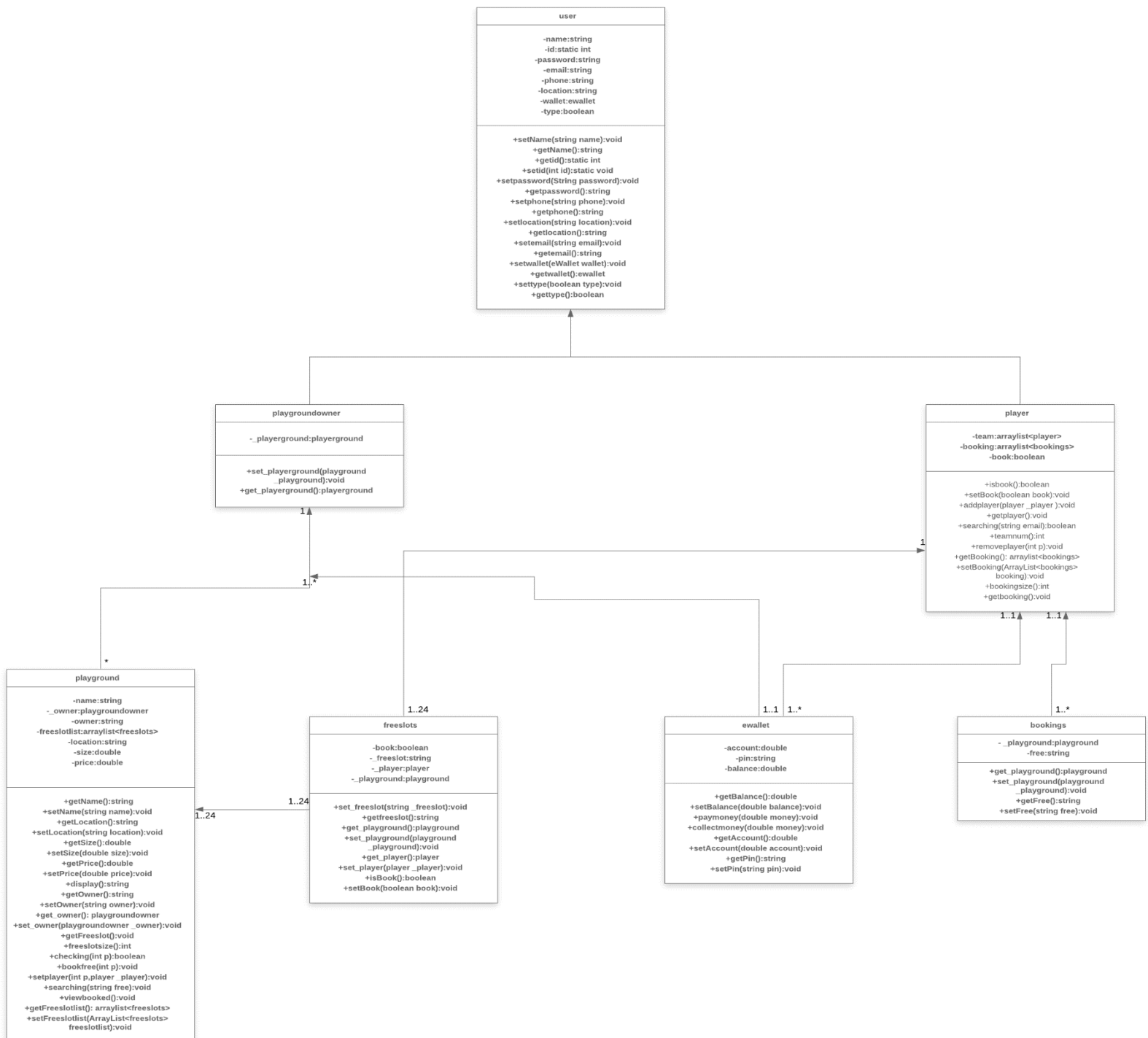
I. Class Diagram(s)



CS251: Phase 2 – <Team Name>

Project: <Project Name>

Software Design Specification





CS251: Phase 2 – <Team Name>

Project: <Project Name>

Software Design Specification

II. Class Descriptions

Class ID	Class Name	Description & Responsibility
1-	Playground owner	Allow the user to create a profile with his name, id, password, email, phone and default location. He can set and change the available hours for booking for his playground and view his bookings, add and remove playground.
2-	Player	Allow the user to create a profile with his name, id, password, email, phone and default location. He can display the playgrounds near to him or in specific location on specific dates and filter them by the hours and date. He can send invitation to his team members via email and can create his favorite team. he can cancel the booking if within cancellation period.
3-	Playground	Allow the user to register a playground with its name, location, size, price per hour and the cancellation period.
4-	e_wallet	A system allows players to pay for the bookings and playground owners to collect their money. Allowing the users to check the money in his eWallet and transfer money to another eWallet.
5-	Booking	Checking available grounds and free slots and allow the user to book a playground for one or more hours if available. Calculating the total price and paying the amount from the player eWallet to the playground owner eWallet, Updating the status of the booked slots.
6-	freeslot	Determine the time available for playing in the playgrounds and closing other times reserved

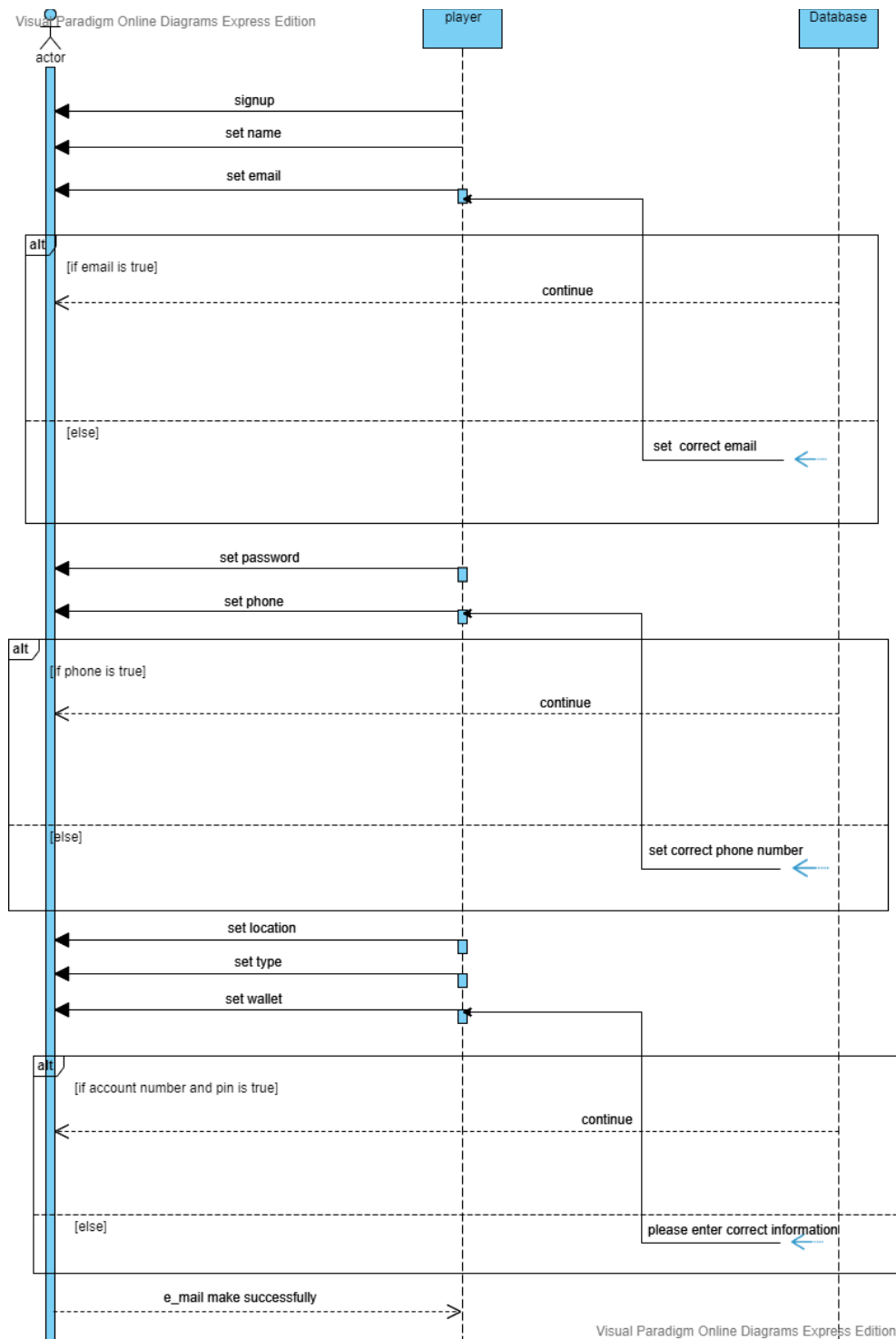
III. Sequence diagrams



CS251: Phase 2 – <Team Name>

Project: <Project Name>

Software Design Specification

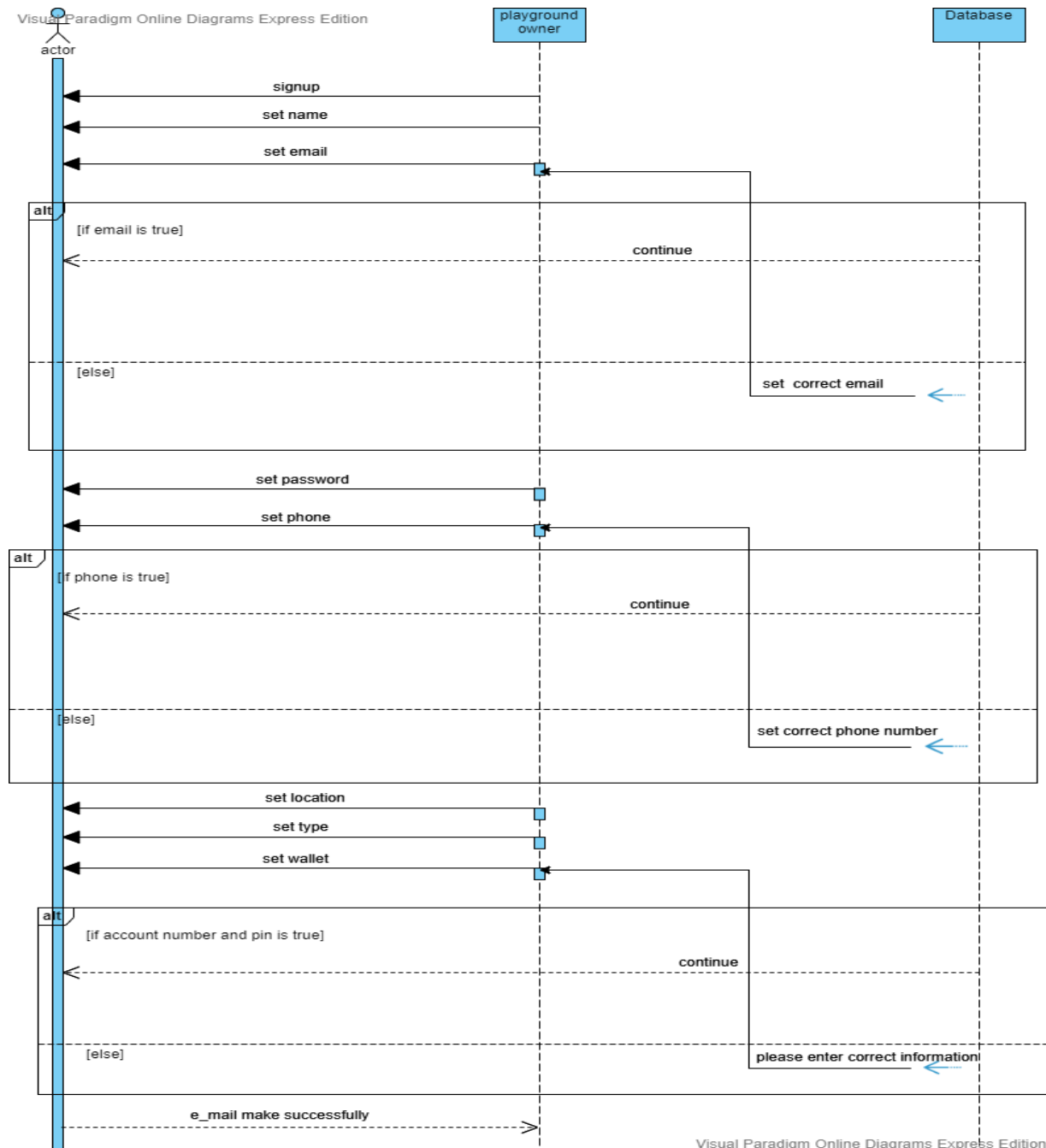




CS251: Phase 2 – <Team Name>

Project: <Project Name>

Software Design Specification

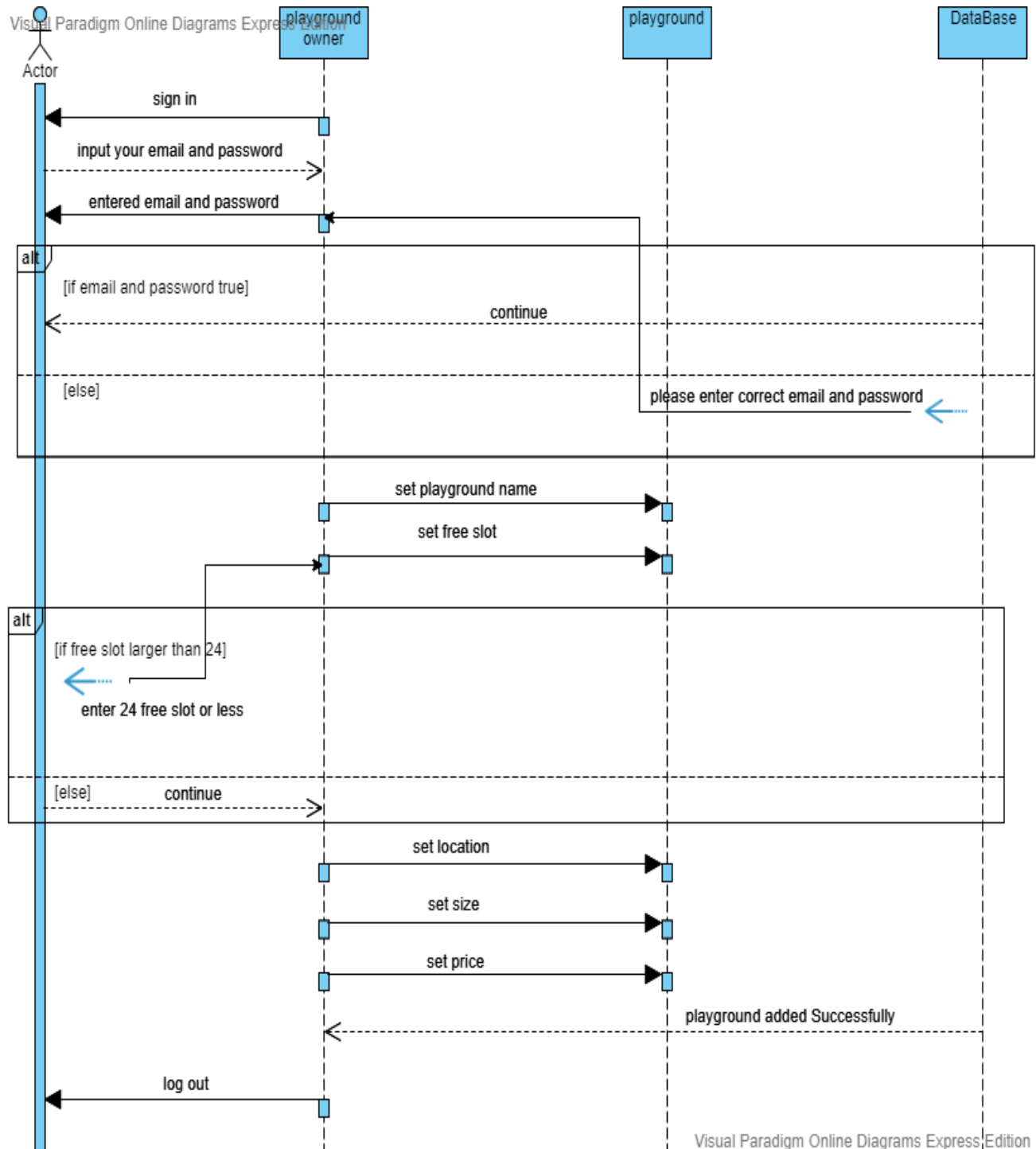




CS251: Phase 2 – <Team Name>

Project: <Project Name>

Software Design Specification

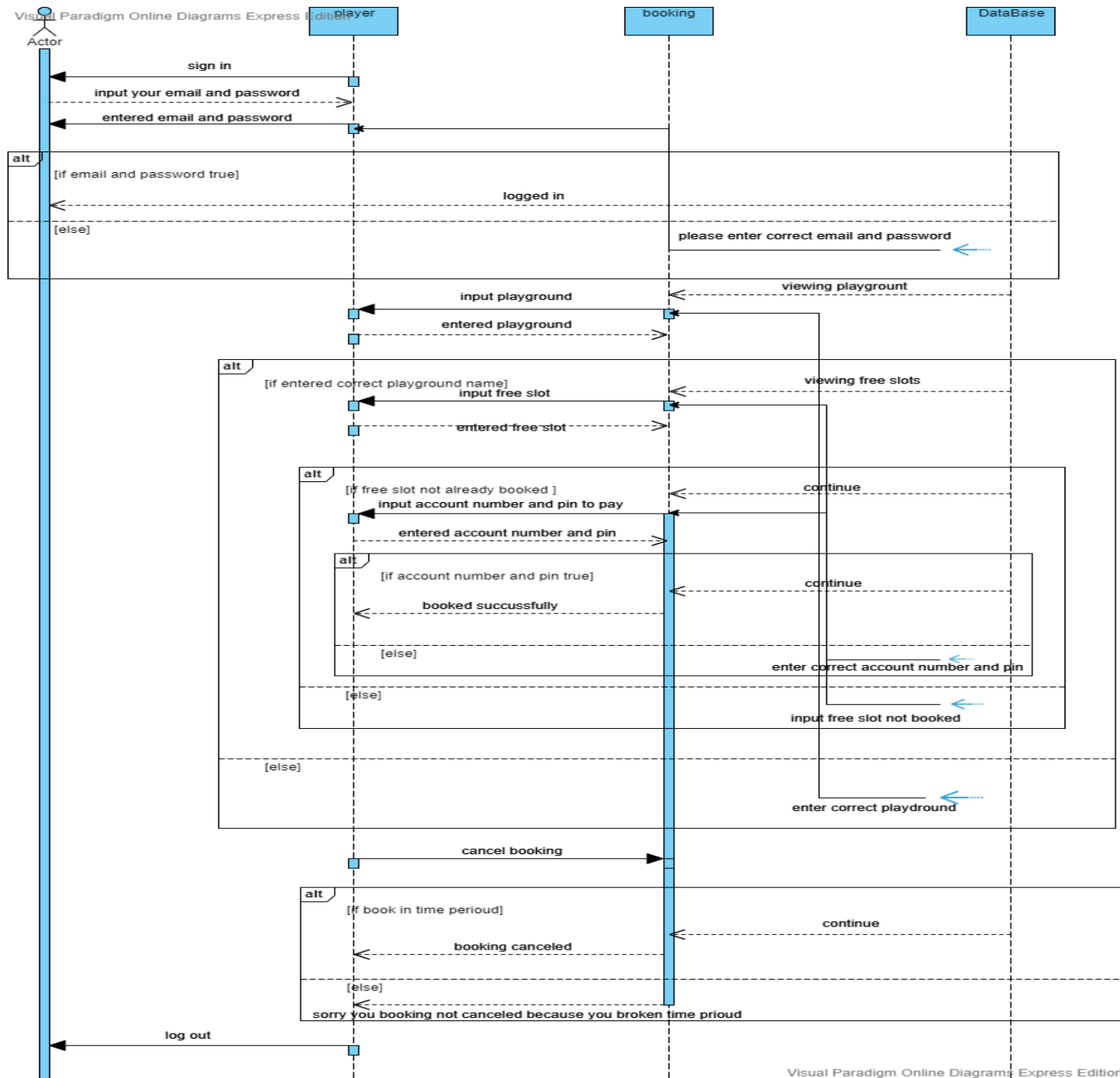




CS251: Phase 2 – <Team Name>

Project: <Project Name>

Software Design Specification

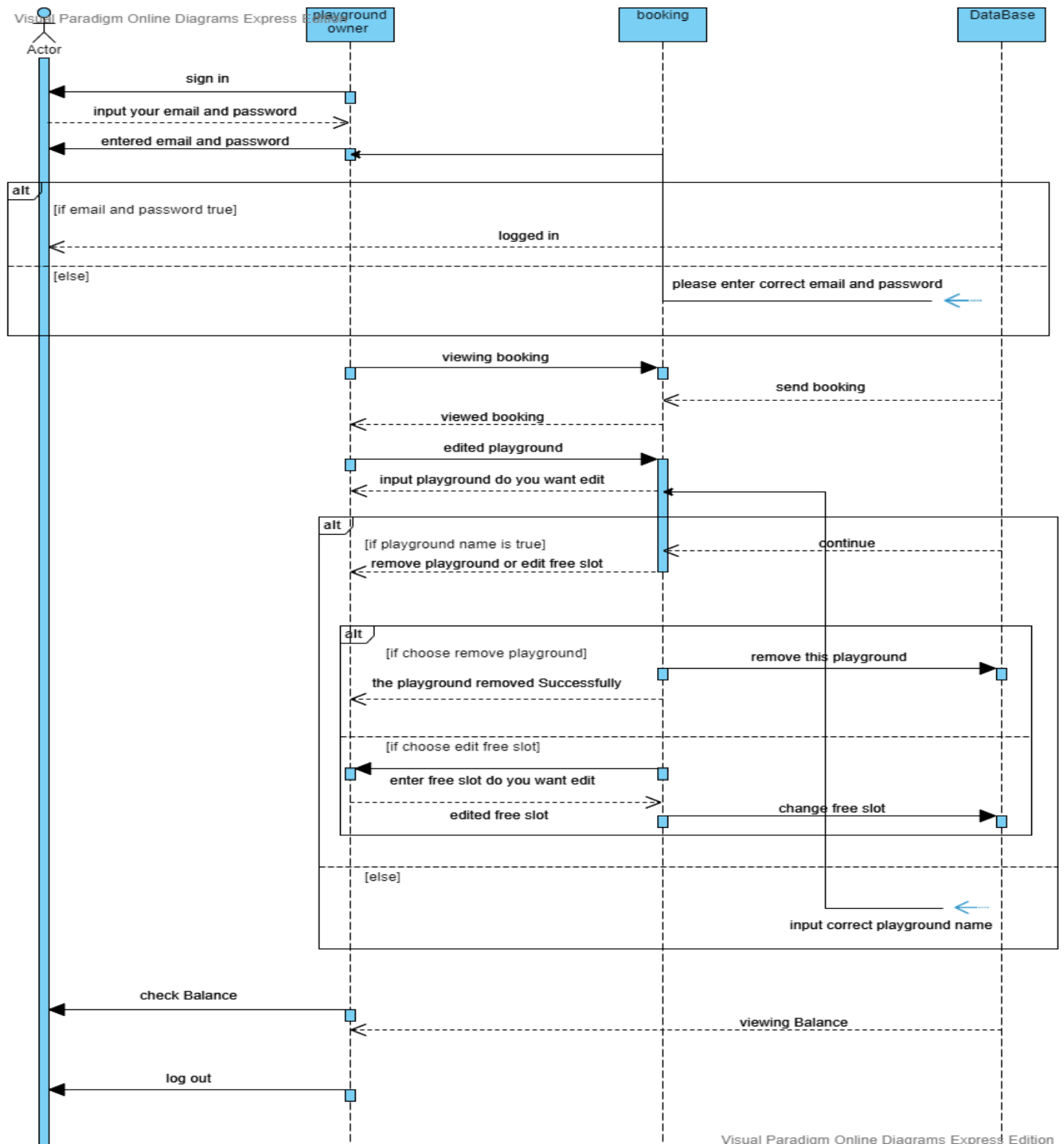




CS251: Phase 2 – <Team Name>

Project: <Project Name>

Software Design Specification



Prepared by Mostara Saad and Monammad El-Ramly V1.0

Edited by Mohamed Samir, Updated to V2.0 by Mohammad El-Ramly 10 Apr 2020



CS251: Phase 2 – <Team Name>

Project: <Project Name>

Software Design Specification

Class - Sequence Usage Table

Sequence Diagram	Classes Used	All Methods Used
1. Signup player	User Player	setName setPassword setPhone setLocation setEmail setWallet setType
2.signup playground owner	User Playground owner	setName setPassword setPhone setLocation setEmail setWallet setType
3.add playground	User Playground owner Playground freeslots	setPassword getPassword getEmail setEmail setNameplayground setLocationplayground setSize setPrice setOwner set_freeslot



CS251: Phase 2 – <Team Name>

Project: <Project Name>

Software Design Specification

Sequence Diagram	Classes Used	All Methods Used
4.player booking	User Player bookings freeslots playground ewallet	setPassword getPassword getEmail setEmail get_playground set_playground getfreeslot set freeslot getbooking paymoney getAccount setAccount getPin setPin bookfree
5.playground owner booking	User Player Playground ewallet	setPassword getPassword getEmail setEmail setAccount



CS251: Phase 2 – <Team Name>

Project: <Project Name>

Software Design Specification

Sequence Diagram	Classes Used	All Methods Used
	bookings	getPin
	freeslots	setPin
		getAccount
		collectmoney
		get_playground
		set_freeslot
		getfreeslot
		isBook
		set_owner
		freeslotsize
		viewbooked
		setLocation
		getLocation
		setName
		getName
		get_playground
		set_playground
		setBook

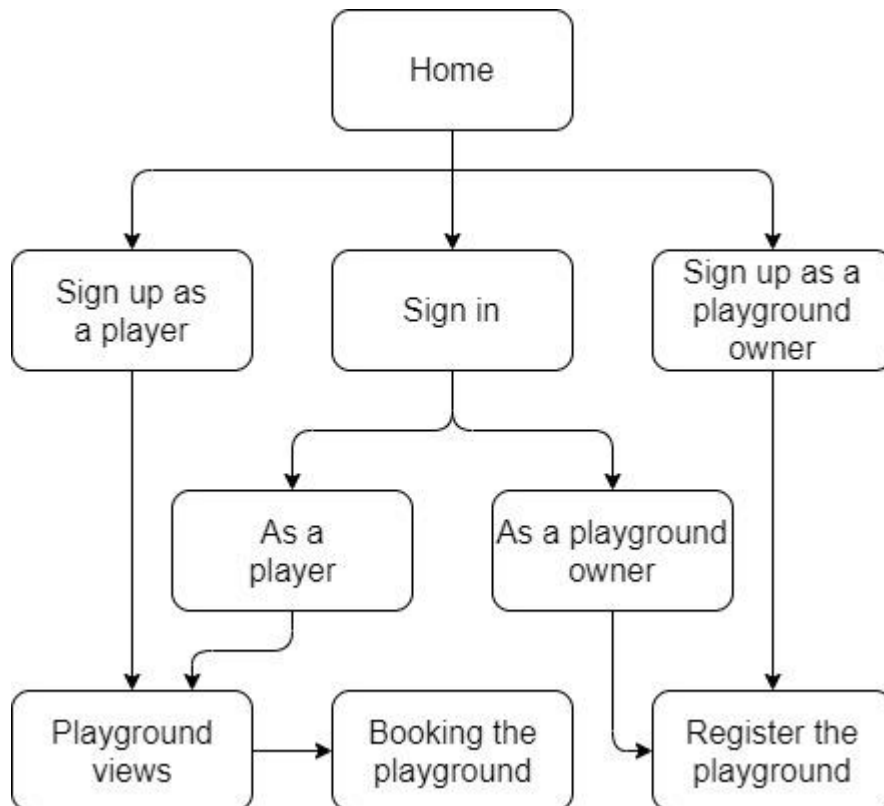


CS251: Phase 2 – <Team Name>

Project: <Project Name>

Software Design Specification

IV. User Interface Design





CS251: Phase 2 – <Team Name>

Project: <Project Name>

Software Design Specification

Home Page

GoFo

https://www.GoFo.com

Logo

About Support

Search

GoFo helps you to manage playgrounds,
book playgrounds, create your team and
deal with your eWallet.

Let's get you started

User Name:

Password:

[Forgot Password?](#)

SIGN IN

If you don't have an account, sign up.

Sign up as **Playground owner**

Sign up as **Player**

For details, [click here](#)

Picture (1)



CS251: Phase 2 – <Team Name>

Project: <Project Name>

Software Design Specification

GoFo

https://www.GoFo.com/player-signup

< Back

Logo

Create your player account

After creating your account, you can book a playground and enjoy with your teams

Sign up with Google

Sign up with Facebook

OR

First Name

Last Name

Phone

National ID

Location

Email

Password

Create your account

By signing up, you agree to our **Terms of Service** and **Privacy Policy**

Already have an account? [Click here to log in](#)

Picture (2)



CS251: Phase 2 – <Team Name>

Project: <Project Name>

Software Design Specification

GoFo

https://www.GoFo.com/playgroundowner-signup

< Back

Logo

Let's get your playground owner account set up

You're on your way to showing your playgrounds and making money

Sign up with Google

Sign up with Facebook

OR

First Name

Last Name

Phone

National ID

Location

Email

Password

Create your account

By signing up, you agree to our **Terms of Service** and **Privacy Policy**

Already have an account? [Click here to log in](#)

Picture (3)



CS251: Phase 2 – <Team Name>

Project: <Project Name>

Software Design Specification

GoFo

https://www.GoFo.com/playgroundowner-signup/register

Logo

Register your playground

Playground Name

Playground Location

Playground Size

Price for hour

Cancellation period

Register

By register, you agree to our **Terms of Service** and **Privacy Policy**
Already have an account? [Click here to log in](#)

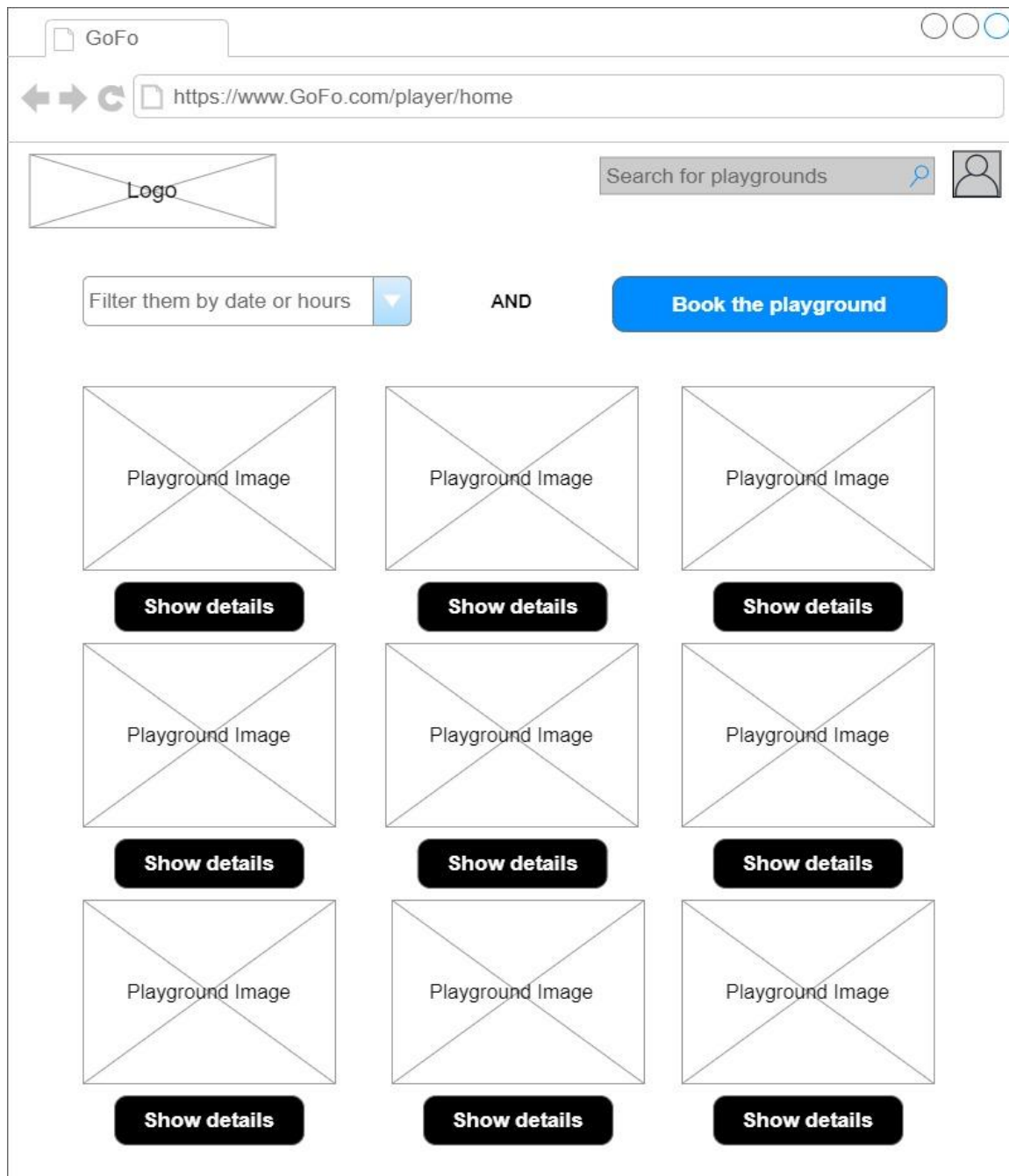
Picture (4)



CS251: Phase 2 – <Team Name>

Project: <Project Name>

Software Design Specification



Picture (5)



CS251: Phase 2 – <Team Name>

Project: <Project Name>

Software Design Specification

GoFo

https://www.GoFo.com/player/home/booking

Logo

Booking the playground

Select the free time slots

Calculating the total price

Send invention to your team

Your friend Email

Your friend Email

Your friend Email

Your friend Email

Pay the amount from your eWallet

Book

Cancel

Playground Image

Picture (6)



CS251: Phase 2 – <Team Name>

Project: <Project Name>

Software Design Specification

Screen ID	Screen Name	Screen / Wireframe Description
Picture (1)	Home	The home of the website
Picture (2)	Sign up as player	The player can create a profile.
Picture (3)	Sign up as playground	The playground owner can create a profile.
Picture (4)	Register the playground owner	The playground owner can register his playground.
Picture (5)	Playgrounds view	The home of the player to show the playgrounds.
Picture (6)	Booking the playground	The player can book playground that he chooses it.

4.Tools

-(untitled Diagram <https://online.visual-paradigm.com/w/tbnnmvoi/app/diagrams/#proj=0&type=SequenceDiagram>, Lucidchart https://app.lucidchart.com/documents/edit/dae99f24-01e1-4d8b-b380-bda6dcd0ab18/15E4yOmcSrN1#?folder_id=home&browser=icon)

-(User interface <https://www.google.com/url?sa=t&source=web&rct=j&url=https://www.draw.io/&ved=2ahUKEwjX-7burPDpAhUKEBQKH78CoUQFjAAegQICBAB&usg=AOvVaw0q79gxfa0gEc3wjqp3nu9>)



CS251: Phase 2 – <Team Name>

Project: <Project Name>

Software Design Specification

5. Ownership Report

Owners	item
Salah Mohamed salah	III. Sequence diagrams, Class - Sequence Usage Table and part in Code Listing and Screen Snapshots
Yasser eid	I. Class Diagram, II. Class Descriptions and part in Code Listing and Screen Snapshots
Amr ayman	IV. User Interface Design and part in Code Listing and Screen Snapshots

6. References

- UML Class Diagram (<https://www.youtube.com/watch?v=UI6lqHOVHic>)
- Creating a Website Wireframe in draw.io (<https://www.youtube.com/watch?v=GDPDIPj5XWY>)
- How to Make a UML Sequence Diagram(<https://www.youtube.com/watch?v=pCK6prSq8aw>)



CS251: Phase 2 – <Team Name>

Project: <Project Name>

Software Design Specification

Appendix A: Code Listing and Screen Snapshots

```
package playground;
import java.util.ArrayList;
import java.util.Scanner;
public class system {
    public static void main (String[] args) {
        int k,searching=0,p,y=0,u=0,price;
        double account,size;
        boolean type =true;
        String
name,password,email,phone,location,pin,free,_searching="0",back="0";
        ArrayList<user> userlist= new ArrayList<user>();
        ArrayList<playground> playgroundlist = new ArrayList<playground>();
        ArrayList<player> playerlist=new ArrayList<player>();
        ArrayList<eWallet>eWalletlist=new ArrayList<eWallet>();
        ArrayList<playgroundowner>ownerlist=new
ArrayList<playgroundowner>();
        do {
            System.out.println("1-signup"+"\\n2-sign in" );
            Scanner myscanner = new Scanner(System.in);
            k=myscanner.nextInt();
            switch(k) {
                case 1:
                    System.out.println("1-Signup as player"+"\\n2-Signup as
playgroundowner");
                    do{
                        k=myscanner.nextInt();
                        if(k==1 || k==2)
                            break;
                        else
                            System.out.println("You entered invalid input please
enter valid input: ");
                    }while(true);
                    user _user=new user();
                    System.out.println("Enter your name: ");
                    name=myscanner.nextLine();
                    name=myscanner.nextLine();
                    _user.setName(name);
                    System.out.println("Enter your password: \\t (password
mustn` t contain space)");
                    do {
                        password=myscanner.nextLine();
```




CS251: Phase 2 – <Team Name>

Project: <Project Name>

Software Design Specification

```
System.out.println("Enter your accountnumber: ");
account=myscanner.nextDouble();
_eWallet.setAccount(account);
System.out.println("Enter your password: ");
pin=myscanner.nextLine();
pin=myscanner.nextLine();
_eWallet.setPin(pin);
eWalletlist.add(_eWallet);
_user.setWallet(_eWallet);
if(k==1) {
    player _player=new player();
    ArrayList<bookings> booking=new
ArrayList<bookings>();

    _player.setBooking(booking);
    _player.setEmail(email);
    _player.setName(name);
    _player.setPhone(phone);
    _player.setPassword(password);
    _player.setLocation(location);
    _player.setWallet(_eWallet);
    _user.setType(true);
    playerlist.add(_player);
}
else {
    playgroundowner _owner=new playgroundowner();
    _owner.setEmail(email);
    _owner.setName(name);
    _owner.setPhone(phone);
    _owner.setPassword(password);
    _owner.setLocation(location);
    _owner.setWallet(_eWallet);
    _user.setType(false);
    ownerlist.add(_owner);
}
userlist.add(_user);
System.out.println("Profile is created successfully!");
break;

case 2:
    System.out.println("Enter your e-mail: ");
    do {
        searching=0;
        email=myscanner.next();
        for(int i=0;i<userlist.size();i++) {
```



CS251: Phase 2 – <Team Name>

Project: <Project Name>

Software Design Specification

```
        if(userlist.get(i).getEmail().equals(email)) {
            searching++;
            _searching=userlist.get(i).getPassword();
            type=userlist.get(i).isType();
            if(type==true) {
                for(int h=0;h<playerlist.size();h++) {

                    if(playerlist.get(h).getEmail().equals(email)) {
                        y=h;
                        break;
                    }
                    else
                        continue;
                }
            }
            else {
                for(int h=0;h<ownerlist.size();h++) {

                    if(ownerlist.get(h).getEmail().equals(email)) {
                        y=h;
                        break;
                    }
                    else
                        continue;
                }
            }
            break;
        }
    else
        continue;
    }
    if(searching==0)
        System.out.println("Invalid E-mail! please enter valid
e-mail: ");
    else
        break;
    }while(true);
    System.out.println("Enter your password: ");
    do {
        searching=0;
        password=myscanner.next();
        if(password.equals(_searching))
            break;
```



CS251: Phase 2 – <Team Name>

Project: <Project Name>

Software Design Specification

```

else
    System.out.println("Wrong password! Enter your
password: ");
    while(true);
    if(type==true) {
        do {

            System.out.println("1- View playgrounds"+"\\n2- Creating
team" + "\\n3- View bookings" + "\\n4- View favourite team" + "\\n5- Check money in
eWallet"+"\\n6- view profile"+"\\n7- log out");
            p=myscanner.nextInt();
            switch(p) {
                case 1:
                    if(playgroundlist.size()==0)
                        System.out.println("\\t\\t playgrounds \\n\\n
There is no playgrounds!");
                    else {
                        System.out.println("\\t\\t playgrounds \\n\\n");
                        for(int i=0;i<playgroundlist.size();i++) {
                            System.out.print((i+1)+"- ");

                            System.out.println(playgroundlist.get(i).display()+"\\n");
                        }
                        System.out.println("Choice playground: ");
                        do {
                            p=myscanner.nextInt();
                            if(p>0&&p<=playgroundlist.size())
                                break;
                            else
                                System.out.println("You entered invalid
input please enter valid input: ");
                        }while(true);
                        for(int i=0;i<playgroundlist.size();i++) {
                            if(p==(i+1)) {
                                playgroundlist.get(i).getFreeslot();
                                System.out.println("Enter free slot:

");
                                do {
                                    k=myscanner.nextInt();

                                    if(k>0&&k<playgroundlist.get(i).freeslotsize()+1) {

                                        if(playgroundlist.get(i).checking(k-1)==true)

```



CS251: Phase 2 – <Team Name>

Project: <Project Name>

Software Design Specification

```
{System.out.println("This freeslot is booked !");
                                break;
                                }
else {
    bookings _booking =new
bookings();

    _booking.setFree(playgroundlist.get(i).getFreeslotlist().get(k-
1).getfreeslot());

    _booking.set_playground(playgroundlist.get(i));
    _booking.set_playground(playgroundlist.get(i));
    playerlist.get(y).getBooking().add(_booking);
    playgroundlist.get(i).bookfree(k-1);;
    playgroundlist.get(i).setplayer(k-1,playerlist.get(y));
    playerlist.get(y).setBook(true);

    playerlist.get(y).getWallet().paymoney(playgroundlist.get(i).getPrice());

    playgroundlist.get(i).get_owner().getWallet().collectmoney(playgroundlist.
get(i).getPrice());

                                System.out.println("The
free slot is booked succesfully!");
                                break;
                                }
else
                                System.out.println("You
entered invalid input please enter valid input: ");
                                }while(true);
                                }
                                else
                                        continue;
                                }}
                                System.out.println("Press b if you want to go
back: ");

                                do {
                                        back=myscanner.next();
```



CS251: Phase 2 – <Team Name>

Project: <Project Name>

Software Design Specification

```
        if(back.equals("b")||back.equals("B"))
            break;
        else
            System.out.println("You entered invalid
input! please enter valid input: ");
            }while(true);
            break;
        case 2:
            for(int
r=playerlist.get(y).teamnum();r<5;r++) {
                System.out.println("Enter your friend
e-mail: ");

                do {
                    back="0";
                    searching=0;
                    email=myscanner.next();
                    for(int i=0;i<userlist.size();i++) {

                        if(userlist.get(i).getEmail().equals(email)&&userlist.get(i).isType()==true
e&&playerlist.get(y).getEmail().equals(email)!=true&&playerlist.get(y).searching
(email)==false) {

                            searching++;
                            for(int h=0;h<playerlist.size();h++) {

                                if(playerlist.get(h).getEmail().equals(email)) {
                                    u=h;
                                    break;
                                }
                                else
                                    continue;
                            }
                            break;
                        }
                    }
                }
                else
                    continue;
            }
            if(searching==0)
                {System.out.println("E-mail is not valid!
press a to enter another email or press b if you want to go back to the menu:
");
                do {
                    back=myscanner.next();
```



CS251: Phase 2 – <Team Name>

Project: <Project Name>

Software Design Specification

```
        if(back.equals("b")||back.equals("B")||back.equals("a")||back.equals("A"))
            break;
        else
            System.out.println("You entered
invalid input! please enter valid input: ");
            }while(true);
            if(back.equals("b")||back.equals("B"))
                break;
            else
                continue;
        }
        else
            break;
    }while(true);
    if(back.equals("b")||back.equals("B"))
        break;
    System.out.println("Invitation is sent");
    playerlist.get(y).addplayer(playerlist.get(u));
    back="0";
    System.out.println("Press b if you want to go back
to menu or a to add another player to the team: ");
    do {
        back=myscanner.next();

        if(back.equals("b")||back.equals("B")||back.equals("a")||back.equals("A"))
            break;
        else
            System.out.println("You entered invalid
input! please enter valid input: ");
            }while(true);
            if(back.equals("b")||back.equals("B"))
                break;
            }
            break;
    case 3:
        do {
            if(playerlist.get(y).bookingsize()==0)
                {System.out.println("\t\t Bookings \n\n
There are no bookings!");
                System.out.println("Press b if you want to
go back: ");
                do {
```



CS251: Phase 2 – <Team Name>

Project: <Project Name>

Software Design Specification

```

        back=myscanner.next();
        if(back.equals("b")||back.equals("B"))
            break;
        else
            System.out.println("You entered
invalid input! please enter valid input: ");
            }while(true);
            break;
        }
    else{
        System.out.println("\t \t bookings\n\n");
        playerlist.get(y).getbooking();
        System.out.println("Press b if you want to
go back and c if you want to cancel booking: ");
        do {
            back=myscanner.next();

            if(back.equals("b")||back.equals("B")||back.equals("c")||back.equals("C"))
                break;
            else
                System.out.println("You entered
invalid input please enter valid input: ");
            }while(true);
            if(back.equals("c")||back.equals("C")) {
                System.out.println("Enter the choice
you want to cancel: ");

                do {
                    p=myscanner.nextInt();

                    if(p>0&&p<playerlist.get(y).bookingsize()+1)
                        break;
                    else
                        System.out.println("You entered
invalid input please enter valid input: ");
                }while(true);
                for(int
i=0;i<playgroundlist.size();i++) {

                    if(playerlist.get(y).getBooking().get(p-
1).get_playground()==playgroundlist.get(i))

                        {playgroundlist.get(i).searching(playerlist.get(y).getBooking().get(p-
1).getFree());

```



CS251: Phase 2 – <Team Name>

Project: <Project Name>

Software Design Specification

```
playerlist.get(y).getBooking().remove(p-1);

playerlist.get(y).getWallet().collectmoney(playgroundlist.get(i).getPrice(
));

playgroundlist.get(i).get_owner().getWallet().paymoney(playgroundlist.get(
i).getPrice());

System.out.println("The
booking is canceled! \n\n");

break;
}
else
continue;
}
}
else
break;
}
}while(true);
break;
case 4:
do {
System.out.println("\t\t Favourite Team \n\n");
playerlist.get(y).getPlayer();
System.out.println("Press b if you want to go back
or m if you want to modify your favourite team: ");
do {
back=myscanner.next();

if(back.equals("b")||back.equals("B")||back.equals("m")||back.equals("M"))
break;
else
System.out.println("You entered invalid
input! please enter valid input: ");
}while(true);
if(back.equals("m")||back.equals("M"))
{
System.out.println("1-add player \n2-remove
player");

p=myscanner.nextInt();
switch(p) {
case 1:
```




CS251: Phase 2 – <Team Name>

Project: <Project Name>

Software Design Specification

```
if(playerlist.get(y).teamnum()==5)
    System.out.println("The team is
full");
friend E-mail: ");
do {
    searching=0;
    email=myscanner.next();
    for(int
i=0;i<userlist.size();i++) {
        if(userlist.get(i).getEmail().equals(email)&&userlist.get(i).isType()==true
e&&playerlist.get(y).getEmail().equals(email)!=true&&playerlist.get(y).searching
(email)==false) {
            searching++;
            for(int
h=0;h<playerlist.size();h++) {
                if(playerlist.get(h).getEmail().equals(email)) {
                    u=h;
                    break;
                }
                else
                    continue;
            }
            break;
        }
        else
            continue;
    }
    if(searching==0)
    {System.out.println("E-mail
is not found! press a to enter another email or press b if you want to go back:
");
    do {
        back=myscanner.next();
        if(back.equals("b")||back.equals("B")||back.equals("a")||back.equals("A"))
            break;
        else
```



CS251: Phase 2 – <Team Name>

Project: <Project Name>

Software Design Specification

```

        System.out.println("You entered invalid input! please enter valid input:
");
                                                                    }while(true);

if(back.equals("b")||back.equals("b"))
                                                                    break;
                                                                    }
                                                                    else
                                                                    break;
                                                                    }while(true);
                                                                    if(searching!=0)
                                                                    {

        System.out.println("Invitation is sent");

        playerlist.get(y).addplayer(playerlist.get(u));
                                                                    }
                                                                    }
                                                                    break;
        case 2:
        playerlist.get(y).getplayer();
        System.out.println("Enter player you
want to remove:");
        do {
            p=myscanner.nextInt();

            if(p>0&&p<playerlist.get(y).teamnum()+1)
            {

                playerlist.get(y).removeplayer(p-1);

                                                                    break;
                                                                    }
                                                                    else
                                                                    System.out.println("You
entered invalid input please enter valid input: ");
                                                                    }while(true);
                                                                    break;
        default:
        System.out.println("You entered invalid
input please enter valid input: ");
                                                                    }
                                                                    }

```



CS251: Phase 2 – <Team Name>

Project: <Project Name>

Software Design Specification

```
        else
            break;
        }while(true);
        break;
    case 5:
        System.out.println("\t\t eWallet \n\n Your balance
is "+playerlist.get(y).getWallet().getBalance());
        System.out.println("Press b if you want to go
back: ");

        do {
            back=myscanner.next();
            if(back.equals("b")||back.equals("B"))
                break;
            else
                System.out.println("You entered invalid
input! please enter valid input: ");
        }while(true);
        break;
    case 6:
        System.out.println("\t\t Profile \n\n Name:
"+playerlist.get(y).getName()+"\n");
        System.out.println("E-mail:
"+playerlist.get(y).getEmail()+"\n");
        System.out.println("Phone:
"+playerlist.get(y).getPhone()+"\n");
        System.out.println("Location:
"+playerlist.get(y).getLocation()+"\n");
        System.out.println("Press b if you want to go
back: ");

        do {
            back=myscanner.next();
            if(back.equals("b")||back.equals("B"))
                break;
            else
                System.out.println("You entered invalid
input! please enter valid input: ");
        }while(true);
        break;
    case 7:
        break;
    default:
        System.out.println("You entered invalid input
please enter valid input: ");
```



CS251: Phase 2 – <Team Name>

Project: <Project Name>

Software Design Specification

```
        }
    }while(p!=7);

    }
else {
    do {
        System.out.println("1-Adding playground"+"\\n2-check
bookings "+"\\n3-check my playground"+"\\n4-check money in eWallet"+"\\n5-view
profile"+"\\n6-logout");
        k=myscanner.nextInt();
        switch(k) {
        case 1:
            playground _playground=new playground();
            System.out.println("Enter playground name: ");
            name=myscanner.nextLine();
            name=myscanner.nextLine();
            _playground.setName(name);
            System.out.println("Enter playground location: ");
            location=myscanner.nextLine();
            _playground.setLocation(location);
            System.out.println("Enter the price per hour: ");
            price=myscanner.nextInt();
            _playground.setPrice(price);
            System.out.println("Enter playground size: ");
            size=myscanner.nextDouble();
            _playground.setSize(size);
            _playground.setOwner(ownerlist.get(y).getName());
            _playground.set_owner(ownerlist.get(y));
            ArrayList<freeslots> freeslotlist=new
ArrayList<freeslots>();

            System.out.println("Enter the numbers of
freeslots: ");
            do {
                k=myscanner.nextInt();
                if(k>0&&k<=24)
                    break;
                else
                    System.out.println("You entered invalid
input enter valid input: \\t (the number of freeslot must be less than or equal
24)");
            }while(true);
            for(int i=0;i<k;i++) {
                freeslots freeslot=new freeslots();
```



CS251: Phase 2 – <Team Name>

Project: <Project Name>

Software Design Specification

```
System.out.println("Enter freeslot
" + (i+1) + ": \t\t" + "(freeslot must be in format 00:00)");
do {
    free = myscanner.next();
    if (free.contains(":"))
        break;
    else
        System.out.println("this is
invalid input please enter valid input: ");
} while (true);
freeslot.set_freeslot(free);
freeslot.set_playground(_playground);
freeslotlist.add(freeslot);
}
_playground.setFreeslotlist(freeslotlist);
System.out.println("Playground Added!");
playgroundlist.add(_playground);
System.out.println("Press b if you want to go
back: ");
do {
    back = myscanner.next();
    if (back.equals("b") || back.equals("B"))
        break;
    else
        System.out.println("You entered invalid
input! please enter valid input: ");
} while (true);
break;
case 2:
    System.out.println("\t\t Bookings\n");
    for (int i = 0; i < playgroundlist.size(); i++) {
        if (playgroundlist.get(i).get_owner() == ownerlist.get(y)) {
            System.out.println("Playground:
" + playgroundlist.get(i).getName() + "\n");
            playgroundlist.get(i).viewbooked();
            System.out.println("\n \n");
        }
        else
            continue;
    }
    System.out.println("Press b if you want to go
back: ");
```



CS251: Phase 2 – <Team Name>

Project: <Project Name>

Software Design Specification

```
do {
    back=myscanner.next();
    if(back.equals("b")||back.equals("B"))
        break;
    else
        System.out.println("You entered invalid
input! please enter valid input: ");
}while(true);
break;
case 3:
do {
    searching=0;
    int q=0;
    System.out.println("\t\t Playgrounds\n\n");
    for(int i=0;i<playgroundlist.size();i++) {

        if(playgroundlist.get(i).get_owner()==ownerlist.get(y)) {
            q++;
            System.out.println(q+"- ");

            System.out.println(playgroundlist.get(i).display()+"\n\n");
            searching++;
        }
        else
            continue;
    }
    if(searching==0)
        {System.out.println("There are no
playgrounds");
        System.out.println("Press b if you want to
go back: ");
        do {
            back=myscanner.next();
            if(back.equals("b")||back.equals("B"))
                break;
            else
                System.out.println("You entered
invalid input! please enter valid input: ");
        }while(true);
        break;
    }
}
else {
```



CS251: Phase 2 – <Team Name>

Project: <Project Name>

Software Design Specification

```
System.out.println("press b if you want to
go back or r to remove playground: ");
do {
    back=myscanner.next();

    if(back.equals("b")||back.equals("B")||back.equals("r")||back.equals("R"))
        break;
    else
        System.out.println("You entered
invalid input! please enter valid input: ");
}while(true);
if(back.equals("r")||back.equals("R")) {
    System.out.println("Enter the playground
you want to remove:");
    do {
        q=0;
        p=myscanner.nextInt();
        if(p>0&&p<=searching) {
            for(int
i=0;i<playgroundlist.size();i++) {
                if(playgroundlist.get(i).get_owner()==ownerlist.get(y)) {
                    q++;
                    if(p==q)

                        break;
                }
            }
            else
                continue;
        }
    }
    break;
}
else
    System.out.println("You entered
invalid input please enter valid input: ");
}while(true);
}
```



CS251: Phase 2 – <Team Name>

Project: <Project Name>

Software Design Specification

```
        }
        }while(true);
        break;
    case 4:
        System.out.println("\t\t eWallet \n\n Your balance
is "+ownerlist.get(y).getWallet().getBalance());
        System.out.println("Press b if you want to go
back: ");

        do {
            back=myscanner.next();
            if(back.equals("b")||back.equals("B"))
                break;
            else
                System.out.println("You entered invalid
input! please enter valid input: ");
        }while(true);
        break;
    case 5:
        System.out.println("\t\t Profile \n\n Name:
"+ownerlist.get(y).getName()+"\n");
        System.out.println("E-mail:
"+ownerlist.get(y).getEmail()+"\n");
        System.out.println("Phone:
"+ownerlist.get(y).getPhone()+"\n");
        System.out.println("Location:
"+ownerlist.get(y).getLocation()+"\n");
        System.out.println("Press b if you want to go
back: ");

        do {
            back=myscanner.next();
            if(back.equals("b")||back.equals("B"))
                break;
            else
                System.out.println("You entered invalid
input! please enter valid input: ");
        }while(true);
        break;
    case 6:
        break;
    default:
        System.out.println("You entered invalid input
please enter valid input: ");
}
```




CS251: Phase 2 – <Team Name>

Project: <Project Name>

Software Design Specification

```
        }while(k!=6);
    }
    }
    }while(true);
}
}
```

```
package playground;
```

```
public class user {
    private String name;
    private static int id =0;
    private String password;
    private String email;
    private String phone;
    private String location;
    private eWallet wallet;
    private boolean type;
    user(){
        id++;
    }
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public static int getId() {
        return id;
    }
    public static void setId(int id) {
        user.id = id;
    }
    public String getPassword() {
        return password;
    }
}
```



CS251: Phase 2 – <Team Name>

Project: <Project Name>

Software Design Specification

```
}  
public void setPassword(String password) {  
    this.password = password;  
}  
public String getPhone() {  
    return phone;  
}  
public void setPhone(String phone) {  
    this.phone = phone;  
}  
public String getLocation() {  
    return location;  
}  
public void setLocation(String location) {  
    this.location = location;  
}  
public String getEmail() {  
    return email;  
}  
public void setEmail(String email) {  
    this.email = email;  
}  
public eWallet getWallet() {  
    return wallet;  
}  
public void setWallet(eWallet wallet) {  
    this.wallet = wallet;  
}  
public boolean isType() {  
    return type;  
}  
public void setType(boolean type) {  
    this.type = type;  
}  
}
```

```
package playground;
```

```
public class playgroundowner extends user {  
    private playground _playground;
```

CU – FCAI – CS251 Introduction to Software Engineering – 2020 - Software Design Specifications

Prepared by Mostafa Saad and Mohammad El-Ramly V1.0

Edited by Mohamed Samir, Updated to V2.0 by Mohammad El-Ramly 10 Apr 2020



CS251: Phase 2 – <Team Name>

Project: <Project Name>

Software Design Specification

```
public playground get_playground() {
    return _playground;
}

public void set_playground(playground _playground) {
    this._playground = _playground;
}

}

package playground;

import java.util.ArrayList;

public class playground {
    private String name;
    private playgroundowner _owner;
    private String owner;
    private ArrayList<freeslots> freeslotlist=new ArrayList<freeslots>();
    private String location;
    private double size;
    private double price;
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public String getLocation() {
        return location;
    }
    public void setLocation(String location) {
        this.location = location;
    }
    public double getSize() {
        return size;
    }
    public void setSize(double size) {
        this.size = size;
    }
    public double getPrice() {
```



CS251: Phase 2 – <Team Name>

Project: <Project Name>

Software Design Specification

```
        return price;
    }
    public void setPrice(double price) {
        this.price = price;
    }
    public String display() {
        return
            "Name: "+name+"\nlocation: "+location+"\nSize:
"+size+"\nPrice: "+price+"\nOwner: "+owner+"\n";
    }
    public String getOwner() {
        return owner;
    }
    public void setOwner(String owner) {
        this.owner = owner;
    }

    public playgroundner get_owner() {
        return _owner;
    }
    public void set_owner(playgroundner _owner) {
        this._owner = _owner;
    }
    public void getFreeslot() {
        for(int i=0;i<freeslotlist.size();i++) {
            if(freeslotlist.get(i).isBook()==true)
                System.out.println((i+1)+"- "+freeslotlist.get(i).getfreeslot()+"\t
(is Booked)");
            else
                System.out.println((i+1)+"-
"+freeslotlist.get(i).getfreeslot());
        }
    }

    public int freeslotsize() {
        return freeslotlist.size();
    }
    public boolean checking(int p) {
        return freeslotlist.get(p).isBook();
    }
    public void bookfree(int p) {
        freeslotlist.get(p).setBook(true);
    }
}
```



CS251: Phase 2 – <Team Name>

Project: <Project Name>

Software Design Specification

```
public void setplayer(int p,player _player) {
    freeslotlist.get(p).set_player(_player);
}
public void searching(String free) {
    for(int i=0;i<freeslotlist.size();i++) {
        if(freeslotlist.get(i).getfreeslot().equals(free))
            freeslotlist.get(i).setBook(false);
        else
            continue;
    }
}

public void viewbooked() {
    int searching=0;
    for(int i=0;i<freeslotlist.size();i++) {
        if(freeslotlist.get(i).isBook()==true)
            searching++;
        else
            continue;
    }
    if(searching==0)
        System.out.println("There are no bookings!");
    else {
        for(int i=0;i<freeslotlist.size();i++) {
            if(freeslotlist.get(i).isBook()==true)
                System.out.println("Freeslot:
"+freeslotlist.get(i).getfreeslot()+"\t Player:
"+freeslotlist.get(i).get_player().getName());
            else
                continue;
        }
    }
}

public ArrayList<freeslots> getFreeslotlist() {
    return freeslotlist;
}

public void setFreeslotlist(ArrayList<freeslots> freeslotlist) {
    this.freeslotlist = freeslotlist;
}

}
```



CS251: Phase 2 – <Team Name>

Project: <Project Name>

Software Design Specification

```
package playground;

import java.util.ArrayList;

public class player extends user{
    private ArrayList<player> team=new ArrayList<player>();
    private ArrayList<bookings> booking=new ArrayList<bookings>();
    private boolean book;
    player(){
        book=false;
    }

    public boolean isBook() {
        return book;
    }

    public void setBook(boolean book) {
        this.book = book;
    }

    public void addplayer(player _player ) {
        if(team.size()==5)
            System.out.println("The team is full");
        else
            team.add(_player);
    }

    public void getplayer() {
        if(team.size()==0)
            System.out.println("There is no favourite team!");
        else {
            for(int i=0;i<team.size();i++) {
                System.out.print((i+1)+"- ");
                System.out.println("Email: "+team.get(i).getEmail());
                System.out.println("\n ");
            }
        }
    }

    public boolean searching(String email) {
        boolean searching=false;
        for(int i=0;i<team.size();i++) {
            if(team.get(i).getEmail().equals(email))
                {System.out.println("This player is invited");
                return true;
            }
        }
        return false;
    }
}
```



CS251: Phase 2 – <Team Name>

Project: <Project Name>

Software Design Specification

```
                searching=true;
                break;
            }
        else
            searching=false;
    }
    return searching;
}

public int teamnum() {
    return team.size();
}

public void removeplayer(int p) {
    team.remove(p);
}

public ArrayList<bookings> getBooking() {
    return booking;
}

public void setBooking(ArrayList<bookings> booking) {
    this.booking = booking;
}

public int bookingsize() {
    return booking.size();
}

public void getbooking() {
    for(int i=0;i<booking.size();i++) {
        System.out.print((i+1)+"- ");
        System.out.println("Freeslot:
"+booking.get(i).getFree()+"\t"+"Playground:
"+booking.get(i).get_playground().getName());
    }
}

package playground;

public class freeslots {
    private boolean book;
    private String _freeslot;
```



CS251: Phase 2 – <Team Name>

Project: <Project Name>

Software Design Specification

```
private player _player;
private playground _playground;
freeslots(){
    _freeslot=null;
    setBook(false);
}

public void set_freeslot(String _freeslot) {
    this._freeslot = _freeslot;
}
public String getfreeslot() {
    return _freeslot;
}
public playground get_playground() {
    return _playground;
}
public void set_playground(playground _playground) {
    this._playground = _playground;
}
public player get_player() {
    return _player;
}
public void set_player(player _player) {
    this._player = _player;
}
public boolean isBook() {
    return book;
}
public void setBook(boolean book) {
    this.book = book;
}
}
```

```
package playground;
```

```
public class eWallet {
private double account;
private String pin;
private double balance;
eWallet(){
    setPin("0");
    balance=5000;
}
```

CU – FCAI – CS251 Introduction to Software Engineering – 2020 - Software Design Specifications

Prepared by Mostafa Saad and Mohammad El-Ramly V1.0

Edited by Mohamed Samir, Updated to V2.0 by Mohammad El-Ramly 10 Apr 2020



CS251: Phase 2 – <Team Name>

Project: <Project Name>

Software Design Specification

```
        account=0;
    }

    public double getBalance() {
        return balance;
    }
    public void setBalance(double balance) {
        this.balance = balance;
    }
    public void paymoney(double money) {
        if(balance<money)
            System.out.println("Your balance is less than the amount of
withdrawl!");
        else
            balance-=money;
    }
    public void collectmoney(double money) {
        balance+=money;
    }

    public double getAccount() {
        return account;
    }

    public void setAccount(double account) {
        this.account = account;
    }

    public String getPin() {
        return pin;
    }

    public void setPin(String pin) {
        this.pin = pin;
    }
}

package playground;

public class bookings {
    private playground _playground;
    private String free;
```

CU – FCAI – CS251 Introduction to Software Engineering – 2020 - Software Design Specifications

Prepared by Mostafa Saad and Mohammad El-Ramly V1.0

Edited by Mohamed Samir, Updated to V2.0 by Mohammad El-Ramly 10 Apr 2020



CS251: Phase 2 – <Team Name>

Project: <Project Name>

Software Design Specification

```
public playground get_playground() {  
    return _playground;  
}  
public void set_playground(playground _playground) {  
    this._playground = _playground;  
}  
public String getFree() {  
    return free;  
}  
public void setFree(String free) {  
    this.free = free;  
}  
}
```



CS251: Phase 2 – <Team Name>

Project: <Project Name>

Software Design Specification

```
eclipse-workspace - playground/src/playground/system.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help

system [Java Application] C:\Program Files\Java\jre1.8.0_121\bin\javaw.exe (Jun 8, 2020, 5:52:08 AM)
1-signup
2-sign in
1
1-Signup as player
2-Signup as playgroundowner
1
Enter your name:
salah mohamed
Enter your password: (password mustn't contain space)
12345
Enter your e-mail:
salah@gmail.com
Enter your location:
13 gizastreet
Enter your phone:
01013020831
Your e-mail has been created successfully!
eWallet System
Enter your accountnumber:
2244668899
Enter your password:
2468
Profile is created successfully!
1-signup
2-sign in
```



CS251: Phase 2 – <Team Name>

Project: <Project Name>

Software Design Specification

```
eclipse-workspace - playground/src/playground/system.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
12345
Enter your e-mail:
salah@gmail.com
Enter your location:
13 gizastreet
Enter your phone:
01013020831
Your e-mail has been created successfully!
eWallet System
Enter your accountnumber:
2244668899
Enter your password:
2468
Profile is created successfully!
1-signup
2-sign in
1
1-Signup as player
2-Signup as playgroundowner
2
Enter your name:
yasser eid
Enter your password: (password mustn't contain space)
13579
Enter your e-mail:
yasser.com
please enter correct form of e_mail!(There must be @ and .and there mustn't be space)!
yasser@gmail.com
Enter your location:
haram
Enter your phone:
0123547892
Your e-mail has been created successfully!
eWallet System
Enter your accountnumber:
324891
Enter your password:
322448
Profile is created successfully!
1-signup
2-sign in
```



CS251: Phase 2 – <Team Name>

Project: <Project Name>

Software Design Specification

eclipse-workspace - playground/src/playground/system.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help



Problems Javadoc Declaration Console

system [Java Application] C:\Program Files\Java\jre1.8.0_121\bin\javaw.exe (Jun 8, 2020, 5:52:08 AM)

1-signup
2-sign in
2

Enter your e-mail:

yasser@gmail.com

Enter your password:

13579

1-Adding playground

2-check bookings

3-check my playground

4-check money in eWallet

5-view profile

6-logout

1

Enter playground name:

oldtraford

Enter playground location:

feasel

Enter the price per hour:

60

Enter playground size:

100.5

Enter the numbers of freeslots:

5

Enter freeslot 1: (freeslot must be in format 00:00)

1:00

Enter freeslot 2: (freeslot must be in format 00:00)

3:00

Enter freeslot 3: (freeslot must be in format 00:00)

5:00

Enter freeslot 4: (freeslot must be in format 00:00)

7:00

Enter freeslot 5: (freeslot must be in format 00:00)

10:00

Playground Added!

Press b if you want to go back:

b



CS251: Phase 2 – <Team Name>

Project: <Project Name>

Software Design Specification

eclipse-workspace - playground/src/playground/system.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help



Problems @ Javadoc Declaration Console

system [Java Application] C:\Program Files\Java\jre1.8.0_121\bin\javaw.exe (Jun 8, 2020, 5:52:08 AM)

1-Adding playground

2-check bookings

3-check my playground

4-check money in eWallet

5-view profile

6-logout

1

Enter playground name:

elathad

Enter playground location:

shobra

Enter the price per hour:

50

Enter playground size:

200

Enter the numbers of freeslots:

3

Enter freeslot 1: (freeslot must be in format 00:00)

12:00

Enter freeslot 2: (freeslot must be in format 00:00)

1:00

Enter freeslot 3: (freeslot must be in format 00:00)

2:00

Playground Added!

Press b if you want to go back:

c

You entered invalid input! please enter valid input:

b

1-Adding playground

2-check bookings

3-check my playground

4-check money in eWallet

5-view profile

6-logout

5

Profile

Name: yasser eid

E-mail: yasser@gmail.com

Phone: 0123547892



CS251: Phase 2 – <Team Name>

Project: <Project Name>

Software Design Specification

eclipse-workspace - playground/src/playground/system.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help



Problems Javadoc Declaration Console

system [Java Application] C:\Program Files\Java\jre1.8.0_121\bin\javaw.exe (Jun 8, 2020, 5:52:08 AM)

1-Adding playground

2-check bookings

3-check my playground

4-check money in eWallet

5-view profile

6-logout

3

Playgrounds

1-

Name: oldtraford

location: feasel

Size: 100.5

Price: 60.0

Owner:yasser eid

2-

Name: elathad

location: shobra

Size: 200.0

Price: 50.0

Owner:yasser eid

press b if you want to go back or r to remove playground:

r

Enter the playground you want to remove:

1

Playground is removed

Playgrounds

1-

Name: elathad

location: shobra

Size: 200.0

Price: 50.0

Owner:yasser eid



CS251: Phase 2 – <Team Name>

Project: <Project Name>

Software Design Specification

```
eclipse-workspace - playground/src/playground/system.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Problems Javadoc Declaration Console
system [Java Application] C:\Program Files\Java\jre1.8.0_121\bin\javaw.exe (Jun 8, 2020, 5:52:08 AM)

press b if you want to go back or r to remove playground:
b
1-Adding playground
2-check bookings
3-check my playground
4-check money in eWallet
5-view profile
6-logout
2
                Bookings

Playground: elathad

There are no bookings!

Press b if you want to go back:
b
1-Adding playground
2-check bookings
3-check my playground
4-check money in eWallet
5-view profile
6-logout
4
                eWallet

Your balance is 5000.0
Press b if you want to go back:
b
1-Adding playground
2-check bookings
3-check my playground
4-check money in eWallet
5-view profile
6-logout
6
1-signup
2-sign in
```




CS251: Phase 2 – <Team Name>

Project: <Project Name>

Software Design Specification

eclipse-workspace - playground/src/playground/system.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help



Problems @ Javadoc Declaration Console

system [Java Application] C:\Program Files\Java\jre1.8.0_121\bin\javaw.exe (Jun 8, 2020, 5:52:08 AM)

```
1-signup
2-sign in
2
Enter your e-mail:
salah@gmail.com
Enter your password:
12345
1- View playgrounds
2- Creating team
3- View bookings
4- View favourite team
5- Check money in eWallet
6- view profile
7- log out
1
           playgrounds

1- Name: elathad
location: shobra
Size: 200.0
Price: 50.0
Owner:yasser eid

Choice playground:
1
1- 12:00
2- 1:00
3- 2:00
Enter free slot:
3
The free slot is booked succesfully!
Press b if you want to go back:
b
1- View playgrounds
2- Creating team
3- View bookings
4- View favourite team
5- Check money in eWallet
6- view profile
7- log out
```



CS251: Phase 2 – <Team Name>

Project: <Project Name>

Software Design Specification

eclipse-workspace - playground/src/playground/system.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help



Problems @ Javadoc Declaration Console

system [Java Application] C:\Program Files\Java\jre1.8.0_121\bin\javaw.exe (Jun 8, 2020, 5:52:08 AM)

1- View playgrounds

2- Creating team

3- View bookings

4- View favourite team

5- Check money in eWallet

6- view profile

7- log out

1

playgrounds

1- Name: elathad

location: shobra

Size: 200.0

Price: 50.0

Owner:yasser eid

Choice playground:

1

1- 12:00

2- 1:00

3- 2:00

Enter free slot:

3

The free slot is booked succesfully!

Press b if you want to go back:

b

1- View playgrounds

2- Creating team

3- View bookings

4- View favourite team

5- Check money in eWallet

6- view profile

7- log out

3

bookings

1- Freeslot: 2:00 Playground: elathad

Press b if you want to go back and c if you want to cancel booking:



CS251: Phase 2 – <Team Name>

Project: <Project Name>

Software Design Specification

```
eclipse-workspace - playground/src/playground/system.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Problems @ Javadoc Declaration Console
system [Java Application] C:\Program Files\Java\jre1.8.0_121\bin\javaw.exe (Jun 8, 2020, 5:52:08 AM)
1-Signup as player
2-Signup as playgroundowner
1
Enter your name:
amr ayman
Enter your password: (password mustn't contain space)
2468
Enter your e-mail:
amr@gmail.com
Enter your location:
shobra
Enter your phone:
0101549854
Your e-mail has been created successfully!
eWallet System
Enter your accountnumber:
321457
Enter your password:
220003
Profile is created successfully!
1-signup
2-sign in
1
1-Signup as player
2-Signup as playgroundowner
2
Enter your name:
ahmed
Enter your password: (password mustn't contain space)
00000
Enter your e-mail:
ahmed@gmail.com
Enter your location:
giza
Enter your phone:
010268126
Your e-mail has been created successfully!
eWallet System
Enter your accountnumber:
65
Enter your password:
5966515
```

Prepared by Mostafa Saad and Mohammad El-Ramly V1.0

Edited by Mohamed Samir, Updated to V2.0 by Mohammad El-Ramly 10 Apr 2020



CS251: Phase 2 – <Team Name>

Project: <Project Name>

Software Design Specification

```
eclipse-workspace - playground/src/playground/system.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
[Icons] Problems @ Javadoc Declaration Console
system [Java Application] C:\Program Files\Java\jre1.8.0_121\bin\javaw.exe (Jun 8, 2020, 5:52:08 AM)
2
Enter your e-mail:
salah@gmail.com
Enter your password:
12345
1- View playgrounds
2- Creating team
3- View bookings
4- View favourite team
5- Check money in eWallet
6- view profile
7- log out
2
Enter your friend e-mail:
amr@gmail.com
Invitation is sent
Press b if you want to go back to menu or a to add another player to the team:
b
1- View playgrounds
2- Creating team
3- View bookings
4- View favourite team
5- Check money in eWallet
6- view profile
7- log out
4
Favourite Team

1- Email: amr@gmail.com

Press b if you want to go back or m if you want to modify your favourite team:
m
1-add player
2-remove player
2
1- Email: amr@gmail.com

Enter player you want to remove:
2
```



CS251: Phase 2 – <Team Name>

Project: <Project Name>

Software Design Specification

```
eclipse-workspace - playground/src/playground/system.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
system [Java Application] C:\Program Files\Java\jre1.8.0_121\bin\javaw.exe (Jun 8, 2020, 5:52:08 AM)
5- Check money in eWallet
6- view profile
7- log out
4
    Favourite Team

1- Email: amr@gmail.com

Press b if you want to go back or m if you want to modify your favourite team:
m
1-add player
2-remove player
2
1- Email: amr@gmail.com

Enter player you want to remove:
2
You entered invalid input please enter valid input:
1
    Favourite Team

There is no favourite team!
Press b if you want to go back or m if you want to modify your favourite team:
b
1- View playgrounds
2- Creating team
3- View bookings
4- View favourite team
5- Check money in eWallet
6- view profile
7- log out
4
    Favourite Team

There is no favourite team!
Press b if you want to go back or m if you want to modify your favourite team:
```




CS251: Phase 2 – <Team Name>

Project: <Project Name>

Software Design Specification

- Google Drive link(<https://drive.google.com/file/d/1-ckmX1yroXXkH1cSfKJ5-BNAm35Tu2c4/view?usp=sharing>)
- Github link (<https://github.com/yassereid/booking-a-playground.git>)

Information for login:

Email: yassereid247@gmail.com

Password: Yy20180334

Authors

- Mostafa Saad and Mohammad El-Ramly (Edited by Mohamed Samir) (V1.0)
- Updated by Mohammad El-Ramly (V2.0)