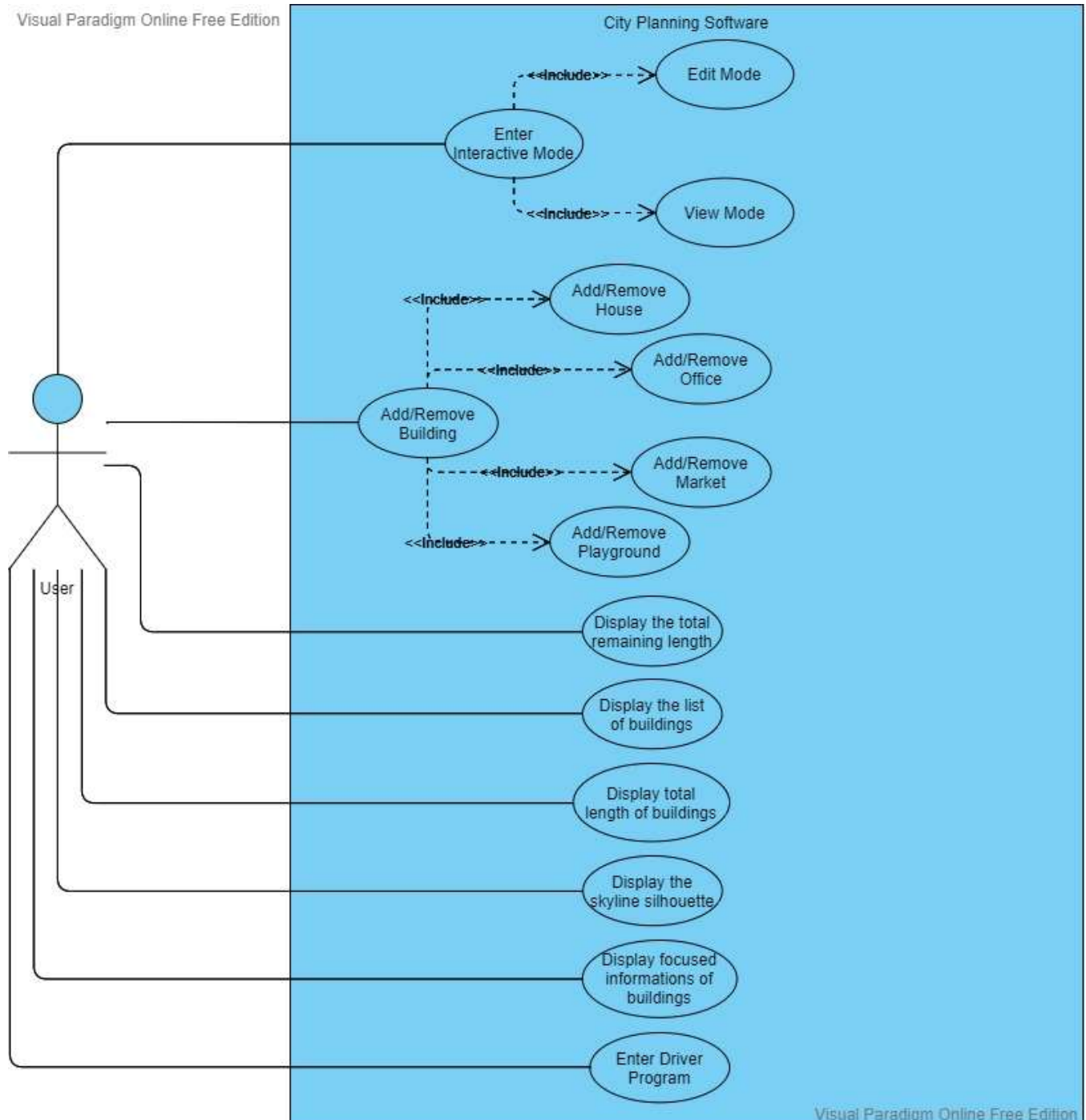# 1. System Requirements

Firstly, most important requirement is a Street object. We need Street object to keep buildings and fields. We'll put buildings to the Street and design them.

Then we need building object and its subclasses (House, Market, Office, Playground). Because Street is meaningless without buildings. We'll put Houses, Markets, Offices, Playgrounds to the Street to provide city planning.
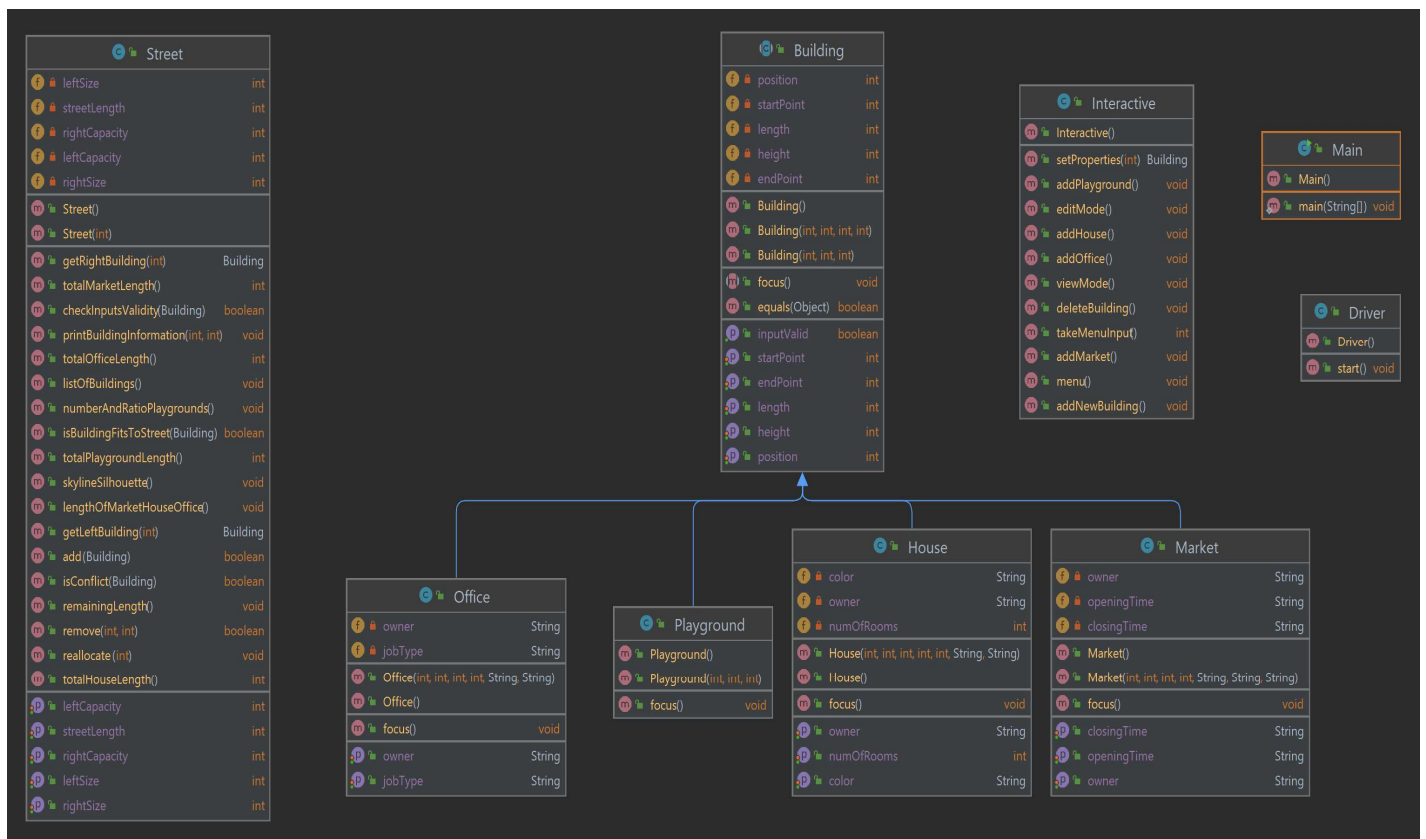
We also need Operating system that have jdk 8 and jre 8 for use the program.

We need some space to store objects of Street, Building, House, Market etc. For example, as default, Building has array with size 10.

2. Use Case Diagram

City Planning Software

<<Include>> → Edit Mode

Enter Interactive Mode

<<Include>> → View Mode

<<Include>> → Add/Remove House

Add/Remove Building

<<Include>> → Add/Remove Office

<<Include>> → Add/Remove Market

<<Include>> → Add/Remove Playground

Display the total remaining length

Display the list of buildings

Display total length of buildings

Display the skyline silhouette

Display focused informations of buildings

Enter Driver Program

User

3. Class Diagram

Street

leftSize	int
streetLength	int
rightCapacity	int
leftCapacity	int
rightSize	int
Street()	
Street(int)	
getRightBuilding(int)	Building
totalMarketLength()	int
checkInputsValidity(Building)	boolean
printBuildingInformation(int, int)	void
totalOfficeLength()	int
listOfBuildings()	void
numberAndRatioPlaygrounds()	void
isBuildingFitsToStreet(Building)	boolean
totalPlaygroundLength()	int
skylineSilhouette()	void
lengthOfMarketHouseOffice()	void
getLeftBuilding(int)	Building
add(Building)	boolean
isConflict(Building)	boolean
remainingLength()	void
remove(int, int)	boolean
reallocate(int)	void
totalHouseLength()	int
leftCapacity	int
streetLength	int
rightCapacity	int
leftSize	int
rightSize	int

Building

position	int
startPoint	int
length	int
height	int
endPoint	int
Building()	
Building(int, int, int, int)	
Building(int, int, int)	
focus()	void
equals(Object)	boolean
inputValid	boolean
startPoint	int
endPoint	int
length	int
height	int
position	int

Interactive

Interactive()	
setProperties(int)	Building
addPlayground()	void
editMode()	void
addHouse()	void
addOffice()	void
viewMode()	void
deleteBuilding()	void
takeMenuInput()	int
addMarket()	void
menu()	void
addNewBuilding()	void

Main

Main()	
main(String[])	void

Driver

Driver()	
start()	void

Office

owner	String
jobType	String
Office(int, int, int, int, String, String)	
Office()	
focus()	void
owner	String
jobType	String

Playground

Playground()	
Playground(int, int, int)	
focus()	void

House

color	String
owner	String
numOfRooms	int
House(int, int, int, int, String, String)	
House()	
focus()	void
owner	String
numOfRooms	int
color	String

Market

owner	String
openingTime	String
closingTime	String
Market()	
Market(int, int, int, int, String, String, String)	
focus()	void
closingTime	String
openingTime	String
owner	String

4. Problem Solutions Approach

Our problem is make an city planning software that will be used for designing a small one Street town. We have Street and we'll fill this Street with buildings. So we also need building class. In Street class, there should be Building array to keep all type of buildings.

There is Building base class and House, Market, Office, Playground subclasses. Each subclass have common and different properties. I wrote common properties to Building base class, like length, height, start point of building etc. I store all of the Building types in one Building array using polymorphism. I assign subclasses references into building array and they make polymorphic call.

To design a city, I need to be able to add/remove building. I do it in Street class because I'll add this objects into building array. They have to placed into street properly.

I both create user menu and driver program. In driver program, every possibilities are evaluated. In user menu, i take an input, i evaluate it, if it is valid, action; else take an input again. When i take input from user, i check validity of input. So i prevent misadd objects to the Street.

5. Test Cases

```
Office office2 = new Office( startPoint: 18, length: 8, height: 15, position: 2, jobType: "a101", owner: "owner4");
Office office3 = new Office( startPoint: 20, length: 4, height: 5, position: 2, jobType: "bim", owner: "owner5");
```

```
street1.add(market1);      //Adding market to right side of the street
street1.add(office2);      //Adding office to right side of the street
street1.add(office3);      //office3 street'e eklenmiyor, cunku koordinatlari diğer office
                           //objesi ile kesisiyor.
street1.listOfBuildings(); //List of Buildings'de de görüldüğü gibi caddenin sağında sadece tek bir office var
```

```
    Left Side Buildings              Right Side Buildings
---------------------------      ---------------------------
        House                             Market

        Playground                        Office

        Office
```

```
street2.remove( pos: 1, index: 0);    //I'm removing all buildings from street.
street2.remove( pos: 1, index: 0);    //I just remove 0th indexes, because when i remove an element from
street2.remove( pos: 1, index: 0);    //street, next elements shifts automatically to the left.
street2.remove( pos: 1, index: 0);
```

```
Select an option:
1) Run Interactive Program
2) Run Drive Code
3) Exit
5
Entered input is out of range, try again.
```

```
Select an option:
1) Run Interactive Program
2) Run Drive Code
3) Exit
asdf
Exception handled. Entered input is invalid, try again.
```

```
Choose one of the followings:

 1) Edit Mode
 2) View Mode
 3) Return to Main Menu...
deneme
Exception handled. Entered input is invalid, try again.
```

```
Enter the number of rooms of the house:
wrongInput
Entered input is invalid, try again.
```

```
Enter the position of the building(1 for left, 2 for right
asfga
Entered input is invalid, try again.
-------------------------------------------------
```

```
Enter the side of building you want to delete(1 For Left, 2 For Right):
2

Enter the index of building you want to delete:
5
Entered input is out of range, try again.
```

```
Choose one of the followings:
1) Display the total remaining length of lands on the street
2) Display the list of buildings on the street
3) Display the number and ratio of length of playgrounds in the street
4) Calculate the total length of street occupied by the markets, houses or offices
5) Display the skyline silhouette of the street
6) Display focused informations of buildings
7) Return to Menu
-5
Entered input is out of range, try again.
```

```
Choose one of the followings:
1) Display the total remaining length of lands on the street
2) Display the list of buildings on the street
3) Display the number and ratio of length of playgrounds in the street
4) Calculate the total length of street occupied by the markets, houses or offices
5) Display the skyline silhouette of the street
6) Display focused informations of buildings
7) Return to Menu
asdf
Entered input is invalid, try again.
```

```
Choose one of the followings:
1) Display the total remaining length of lands on the street
2) Display the list of buildings on the street
3) Display the number and ratio of length of playgrounds in the street
4) Calculate the total length of street occupied by the markets, houses or offices
5) Display the skyline silhouette of the street
6) Display focused informations of buildings
7) Return to Menu
7

Choose one of the followings:

 1) Edit Mode
 2) View Mode
 3) Return to Main Menu...
```

```
    Left Side Buildings                    Right Side Buildings
----------------------------           ----------------------------
        House                                   Office

        House                                   Office

        Market                                  Market

        House

Total Remanining Length is: 26

Number of Playgrounds in the Street: 0
Ratio of Playgrounds in the Street: 0,00

Total Length of Street Occupied by the Houses: 24
Total Length of Street Occupied by the Offices: 27
Total Length of Street Occupied by the Markets: 23
```

```
     Left Side Buildings                Right Side Buildings
----------------------------        ----------------------------
         House                              Market

         Playground                               Office

         Office

Total Remanining Length is: 16

Number of Playgrounds in the Street: 1
Ratio of Playgrounds in the Street: 0,07

Total Length of Street Occupied by the Houses: 10
Total Length of Street Occupied by the Offices: 15
Total Length of Street Occupied by the Markets: 15
```

```
      Left Side Buildings                Right Side Buildings
----------------------------        ----------------------------
Total Remanining Length is: 100

Number of Playgrounds in the Street: 0
Ratio of Playgrounds in the Street: 0,00

Total Length of Street Occupied by the Houses: 0
Total Length of Street Occupied by the Offices: 0
Total Length of Street Occupied by the Markets: 0
```

6. Running Command and Results

The file should be unzipped and opened as a Project in IntellijIdea. All the methods are accessible on driver code.

I couldn't implement skyline shiloutte.