GTU Department of Computer Engineering
CSE 222/505 - Spring 2022

Homework 1 Report
Name : Md Sarwar Hossain
Student Number : 161044121

System Requirements:

This software is for small one street city planning. The buildings of the city are placed around the street. So the first thing the software requires a length for the street. Based on the length we can position the buildings of the city.

The city can have different kinds of building. We categorize the building into 3 groups. They are Houses, Markets and Offices. Besides these we also have playground. As playground is not owned by anyone we are still counting it as kind of building but not completely.

For the general kind of building , all of them length , height, owner , where they are positioned on the street. For the playground we will have just length but no owner or height. Besides these common properties, a particular type of building may have some other properties.

Suppose, a house is mainly known by it's owner , what is it's color ? Or how many rooms it has. When building the system, we will keep this thing in mind.
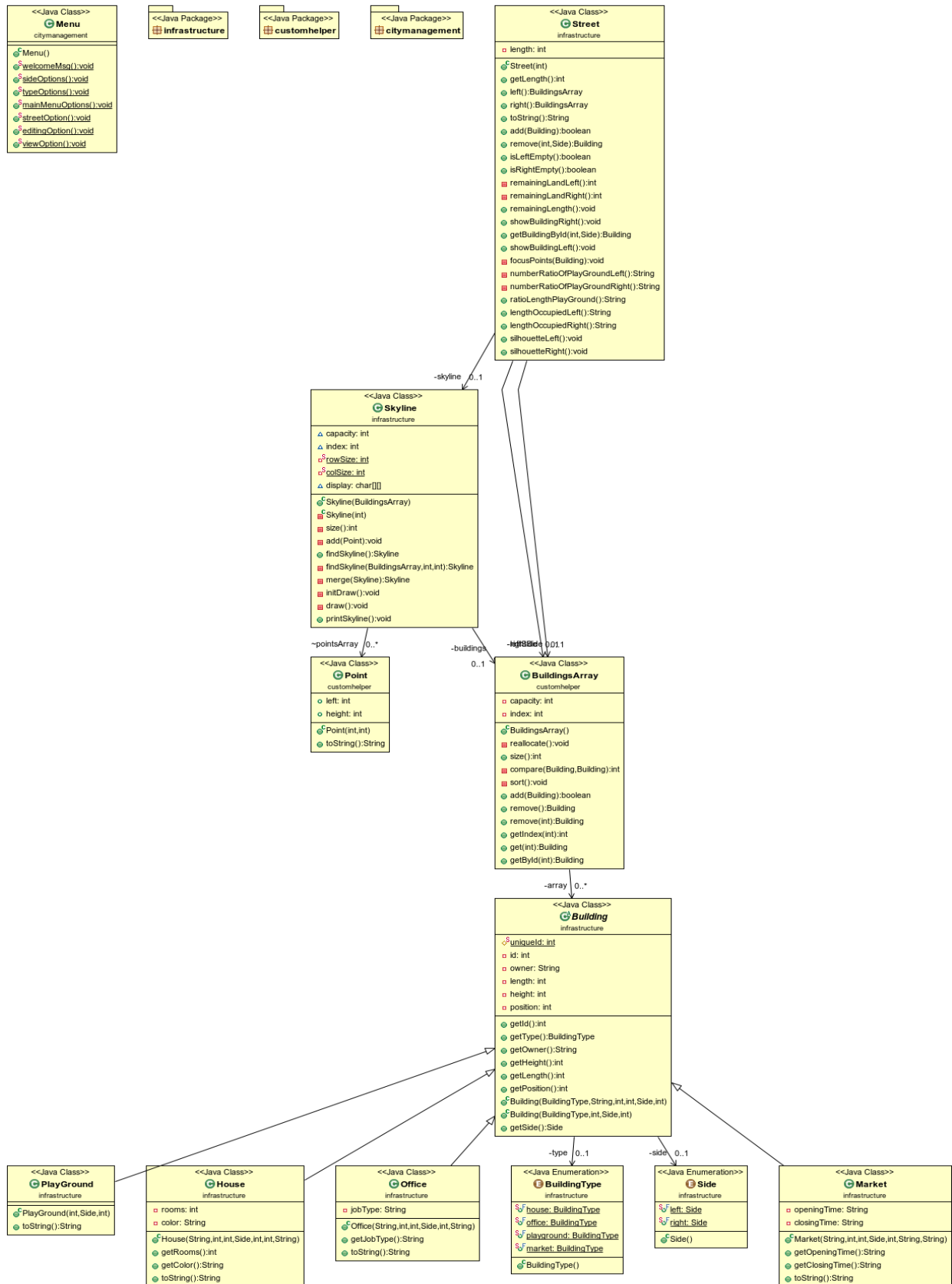
Like the house, A office is mainly known for what kind of service does it offer. Same way , A market is marked by it's opening and closing time.

All these properties of a building needs to handled accordingly. After we plan the city, We should have a nice view about , how it's gonna look. For that we need to plot the skyline silhouette.
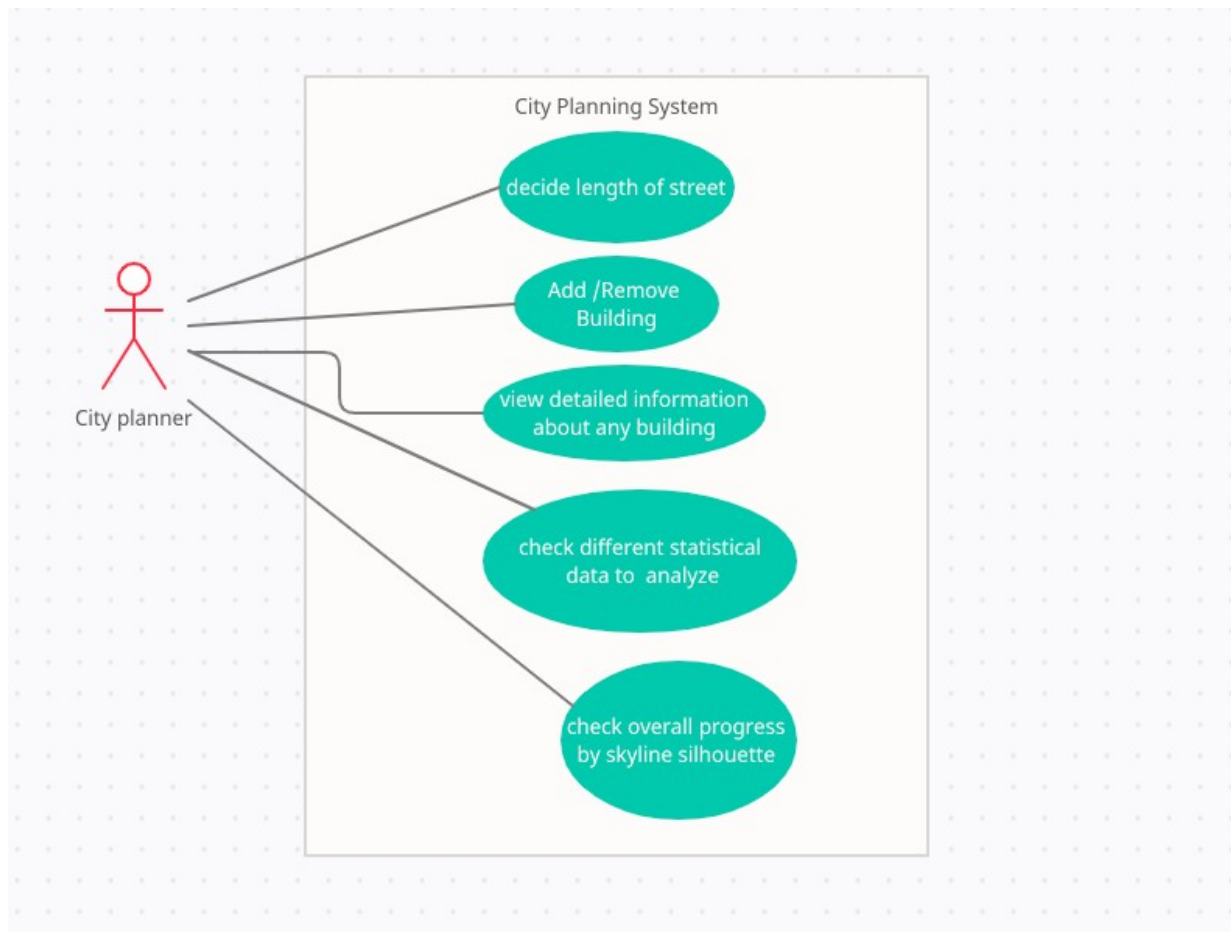
We should also ensure that , when placing a building in the city , It should properly city with the aspect of the city.

All these things we need to keep in mind while building the system.

Class Diagram :

<<Java Class>>
Menu
citymanagement
- Menu()
- welcomeMsg():void
- sideOptions():void
- typeOptions():void
- mainMenuOptions():void
- streetOption():void
- editingOption():void
- viewOption():void

<<Java Package>>
infrastructure

<<Java Package>>
customhelper

<<Java Package>>
citymanagement

<<Java Class>>
Street
infrastructure
- length: int
- Street(int)
- getLength():int
- left():BuildingsArray
- right():BuildingsArray
- toString():String
- add(Building):boolean
- remove(int,Side):Building
- isLeftEmpty():boolean
- isRightEmpty():boolean
- remainingLandLeft():int
- remainingLandRight():int
- remainingLength():void
- showBuildingRight():void
- getBuildingById(int,Side):Building
- showBuildingLeft():void
- focusPoints(Building):void
- numberRatioOfPlayGroundLeft():String
- numberRatioOfPlayGroundRight():String
- ratioLengthPlayGround():String
- lengthOccupiedLeft():String
- lengthOccupiedRight():String
- silhouetteLeft():void
- silhouetteRight():void

-skyline 0..1

<<Java Class>>
Skyline
infrastructure
- capacity: int
- index: int
- rowSize: int
- colSize: int
- display: char[][]
- Skyline(BuildingsArray)
- Skyline(int)
- size():int
- add(Point):void
- findSkyline():Skyline
- findSkyline(BuildingsArray,int,int):Skyline
- merge(Skyline):Skyline
- initDraw():void
- draw():void
- printSkyline():void

~pointsArray 0..*

<<Java Class>>
Point
customhelper
- left: int
- height: int
- Point(int,int)
- toString():String

-buildings 0..1

-leftSide 0..1 -rightSide 0..1

<<Java Class>>
BuildingsArray
customhelper
- capacity: int
- index: int
- BuildingsArray()
- reallocate():void
- size():int
- compare(Building,Building):int
- sort():void
- add(Building):boolean
- remove():Building
- remove(int):Building
- getIndex(int):int
- get(int):Building
- getById(int):Building

-array 0..*

<<Java Class>>
Building
infrastructure
- uniqueId: int
- id: int
- owner: String
- length: int
- height: int
- position: int
- getId():int
- getType():BuildingType
- getOwner():String
- getHeight():int
- getLength():int
- getPosition():int
- Building(BuildingType,String,int,int,Side,int)
- Building(BuildingType,int,Side,int)
- getSide():Side

-type 0..1   -side 0..1

<<Java Class>>
PlayGround
infrastructure
- PlayGround(int,Side,int)
- toString():String

<<Java Class>>
House
infrastructure
- rooms: int
- color: String
- House(String,int,int,Side,int,int,String)
- getRooms():int
- getColor():String
- toString():String

<<Java Class>>
Office
infrastructure
- jobType: String
- Office(String,int,int,Side,int,String)
- getJobType():String
- toString():String

<<Java Enumeration>>
BuildingType
infrastructure
- house: BuildingType
- office: BuildingType
- playground: BuildingType
- market: BuildingType
- BuildingType()

<<Java Enumeration>>
Side
infrastructure
- left: Side
- right: Side
- Side()

<<Java Class>>
Market
infrastructure
- openingTime: String
- closingTime: String
- Market(String,int,int,Side,int,String,String)
- getOpeningTime():String
- getClosingTime():String
- toString():String

Use case :

Problem Solution Approach:
         To fulfill the requirements of the system, I started from very root of the problem. That is the buildings. Since all the building have some sorts of similarities, I put them in a base abstract class.
These class will hold some of the basic features of a building , like length , height , position, owner.
Besides typical building , Playground is also under the building but with no owner and height. So I introduced constructor overload to deal with playground. Besides the position co-ordinate, the side of the street, where the building will be placed is also important. So for all the building , Side became a important factor while designing the system.

The most challenging part, is the skyline silhouette printing. I decided to divide and conquer algorithm for this place. This works like same as merger sort. So first we divide the buildings into two smaller sub section until there is no more sub section possible . Then from the smallest point we recursively merge them on basis of changing height. For displaying the silhouette, I used a char array.

For better understanding , I introduced some enumeration like Building Type , Side. And finally to combine all the classes, I used a Street class like the main character.

Test Cases :

Case 1. Setting up street length
Expected Output: Street will have a length
Result : pass

```
Driver [Java Application] /usr/lib/jvm/java-11-openjdk-amd64/bin/java (Mar 7, 2022, 3:35:24 AM)

                ---Welcome To City Planning---

Please Enter Street Length : 55
Street Length : 55

1. Enter Editing Mode
2. Enter Viewing Mode
0. Exit
Enter Your Choice :
```

2. add a house
Expected Output : A new House will be created and added
Result : pass

```
Driver [Java Application] /usr/lib/jvm/java-11-openjdk-amd64/bin/jav
                ---Welcome To City Planning---

Please Enter Street Length : 55
Street Length : 55

1. Enter Editing Mode
2. Enter Viewing Mode
0. Exit
Enter Your Choice : 1

1. add a building
2. remove a building
0. go back
Enter Your Choice : 1

1. Office
2. House
3. Market
4. PlayGround
0. Go Back

Chose a Building Type : 2
Enter Length : 7
Enter Height : 10
Enter position : 0
Enter Number of Rooms : 3
Enter Owner Name : sarwar
Enter Color of the House : red
1. Left
2. Right
Choose side of the street : 1
House Added Successfully (street length - 55) :
owner    : sarwar
length   : 7
height   : 10
side     : left
position: 0
rooms    : 3
color    :red
```

3.add a office
Expected Output : A new office will be created and added
result : pass

```
1. Office
2. House
3. Market
4. PlayGround
0. Go Back

Chose a Building Type : 1
Enter Length : 8
Enter Height : 5
Enter position : 5
Enter Owner Name : swr
Enter Job Type : unknown
1. Left
2. Right
Choose side of the street : 1
Office Added Successfully (street length - 55):
owner    : swr
length   : 8
height   : 5
side     : left
position: 5
job type: unknown
```

4. add a market
Expected Output : A new market will be created and added
result :

```
1. Office
2. House
3. Market
4. PlayGround
0. Go Back

Chose a Building Type : 3
Enter Length : 10
Enter Height : 20
Enter position : 20
Enter Owner Name : swr
Enter Opening Time : morning 10
Enter Closing Time : evening 10
1. Left
2. Right
Choose side of the street : 1
Market Added Successfully(street length - 55) :
owner        : swr
length       : 10
height       : 20
side         : left
position     : 20
opening time: morning 10
closing time:evening 10
```

5. add a playground
Expected Output : A new playground will be created and added

```
1. Office
2. House
3. Market
4. PlayGround
0. Go Back

Chose a Building Type : 4
Enter Length : 10
Enter position : 30
1. Left
2. Right
Choose side of the street : 1
Playground Added Successfully (street length - 55) :
length  : 10
side    : left
position: 30
```

6. view list of building
Expected Output: All the buildings will be shown with by their focus point
result : pass

```
1. Enter Editing Mode
2. Enter Viewing Mode
0. Exit
Enter Your Choice : 2

1. Remaining Length of Lands on the street
2. List of Buildings on the Left Side of the  Street
3. List of Buildings on the Right Side of the  Street
4. Number and ration of Length of Playgrounds on the street
5. Total Length Occupied By Market / House / Office
6. SkyLine Silhouette of the Street
0. Go Back
2
                --Building on Left Side--
id : 1  house owner : sarwar
id : 2  office job type : unknown
id : 3  closing time : evening 10
id : 4  building type : playground
```

7. remove a building
Expected Output : If a valid id is provided , the particular building should be removed
result : pass

```
1. add a building
2. remove a building
0. go back
Enter Your Choice : 2
1. Left
2. Right
Choose side of the street : 1
                        ---Building on the left side of the street---
id : 1  house owner : sarwar
id : 2  office job type : unknown
id : 3  closing time : evening 10
id : 4  building type : playground
Enter Building Id to Remove(Press 0 to GoBack ) :
3
Building removed Successfully :
owner       : swr
length      : 10
height      : 20
side        : left
position    : 20
opening time: morning 10
closing time:evening 10
```

8. add a building with wrong value
Expected Output : It should give us error and building will not be added

result : pass

```
1. add a building
2. remove a building
0. go back
Enter Your Choice : 1

1. Office
2. House
3. Market
4. PlayGround
0. Go Back

Chose a Building Type : 1
Enter Length : 100
invalid length
```

9. remove a building with invalid id
Expected Output : It shouldn't remove any building as the didn't match
Result : pass

```
1. add a building
2. remove a building
0. go back
Enter Your Choice : 2
1. Left
2. Right
Choose side of the street : 1
                        ---Building on the left side of the street---
id : 1  house owner : sarwar
id : 2  office job type : unknown
id : 4  building type : playground
Enter Building Id to Remove(Press 0 to GoBack ) :
22
Failed to Remove Building. Carefully Type Building Id !!!
```

10.total length of street occupied by the markets, houses or offices.

Expected Output : It should show all the space taken by market /house and office in details
Result : pass

```
1. Remaining Length of Lands on the street
2. List of Buildings on the Left Side of the  Street
3. List of Buildings on the Right Side of the  Street
4. Number and ration of Length of Playgrounds on the street
5. Total Length Occupied By Market / House / Office
6. SkyLine Silhouette of the Street
0. Go Back
5
On Left Side, Length Occupied By House : 7 Market : 0 Office : 8  Total: 15
On Right Side, Length Occupied By House : 0 Market : 0 Office : 0  Total: 0
```

11. invalid input handling
Expected Output : All forms of error should be caught
Result : pass

```
1. Office
2. House
3. Market
4. PlayGround
0. Go Back

Chose a Building Type : 1
Enter Length : 0
Enter Height : 20
Enter position : -1
invalid position !!!
```

12. adding building with invalid position
Expected Output: Out of bound positions are not allowed
Result : pass

```
1. Office
2. House
3. Market
4. PlayGround
0. Go Back

Chose a Building Type : 1
Enter Length : 0
Enter Height : 20
Enter position : -1
invalid position !!!
```

13. Remaining land
Expected Output : It should display the amount of unoccupied land on both side of the street

Result : pass

```
Choose side of the street : 1
Playground Added Successfully (street length - 55) :
length  : 10
side    : left
position: 0


1. Enter Editing Mode
2. Enter Viewing Mode
0. Exit
Enter Your Choice : 2

1. Remaining Length of Lands on the street
2. List of Buildings on the Left Side of the  Street
3. List of Buildings on the Right Side of the  Street
4. Number and ration of Length of Playgrounds on the street
5. Total Length Occupied By Market / House / Office
6. SkyLine Silhouette of the Street
0. Go Back
Enter your choice : 1
Remaining Length on Left Side : 45
Remaining Length on Right Side : 55
```

14. ratio of playground length
Expected Output: It should show you the number of playground and their ratio in terms of total area
Result : pass

```
1. Remaining Length of Lands on the street
2. List of Buildings on the Left Side of the  Street
3. List of Buildings on the Right Side of the  Street
4. Number and ratio of Length of Playgrounds on the street
5. Total Length Occupied By Market / House / Office
6. SkyLine Silhouette of the Street
0. Go Back
Enter your choice : 4
On Left Side - Number of PlayGround : 1 Ratio of Length Compared to Street : 0.18181818181818182
On Right Side - Number of PlayGround : 0 Ratio of Length Compared to Street : 0.0
```

15. building details
expected Output: It should show you all the information about a particular buildings
result : pass

```
1. Remaining Length of Lands on the street
2. List of Buildings on the Left Side of the  Street
3. List of Buildings on the Right Side of the  Street
4. Number and ratio of Length of Playgrounds on the street
5. Total Length Occupied By Market / House / Office
6. SkyLine Silhouette of the Street
0. Go Back
Enter your choice : 2
                  --Building on Left Side--
id : 1  building type : playground

Do you want to see detail of a property(y/n) : y
Enter a building id : 1
length  : 10
side    : left
position: 0
```

16. silhouette display
Expected Output:
It should display a skyline silhouette in the console.
Result : pass. This shape was printed on the basis of current building cordinates

```
1. Remaining Length of Lands on the street
2. List of Buildings on the Left Side of the  Street
3. List of Buildings on the Right Side of the  Street
4. Number and ratio of Length of Playgrounds on the street
5. Total Length Occupied By Market / House / Office
6. SkyLine Silhouette of the Street
0. Go Back
Enter your choice : 6
1. Left
2. Right
Choose side of the street : 1
                ---Silhouette Display of Left Side of Street---


                    ***********
                    *         *
                    *         *             ******
                    *         *             *    *
                    *         *             *    *
*******  ***********         *         ******  ******
*      *  *         *         *         *            *
*      *  *         *         *         *            *
*      *  *         *         *         *            *
*      *  *         *         *         *            *
*      ****         *         *         *            *
*                   *         *                      *
*                   *         *                      *
*                   *         *                      *
*                   *         *                      *
*                   ***********                      *

1. Remaining Length of Lands on the street
```