GIT Department of Computer Engineering
CSE 222/505 - Spring 2022
Homework 3 Report

Yunus Emre Yumşak
1801042659

1. SYSTEM REQUIREMENTS

City planning software is implemented using two main classes;

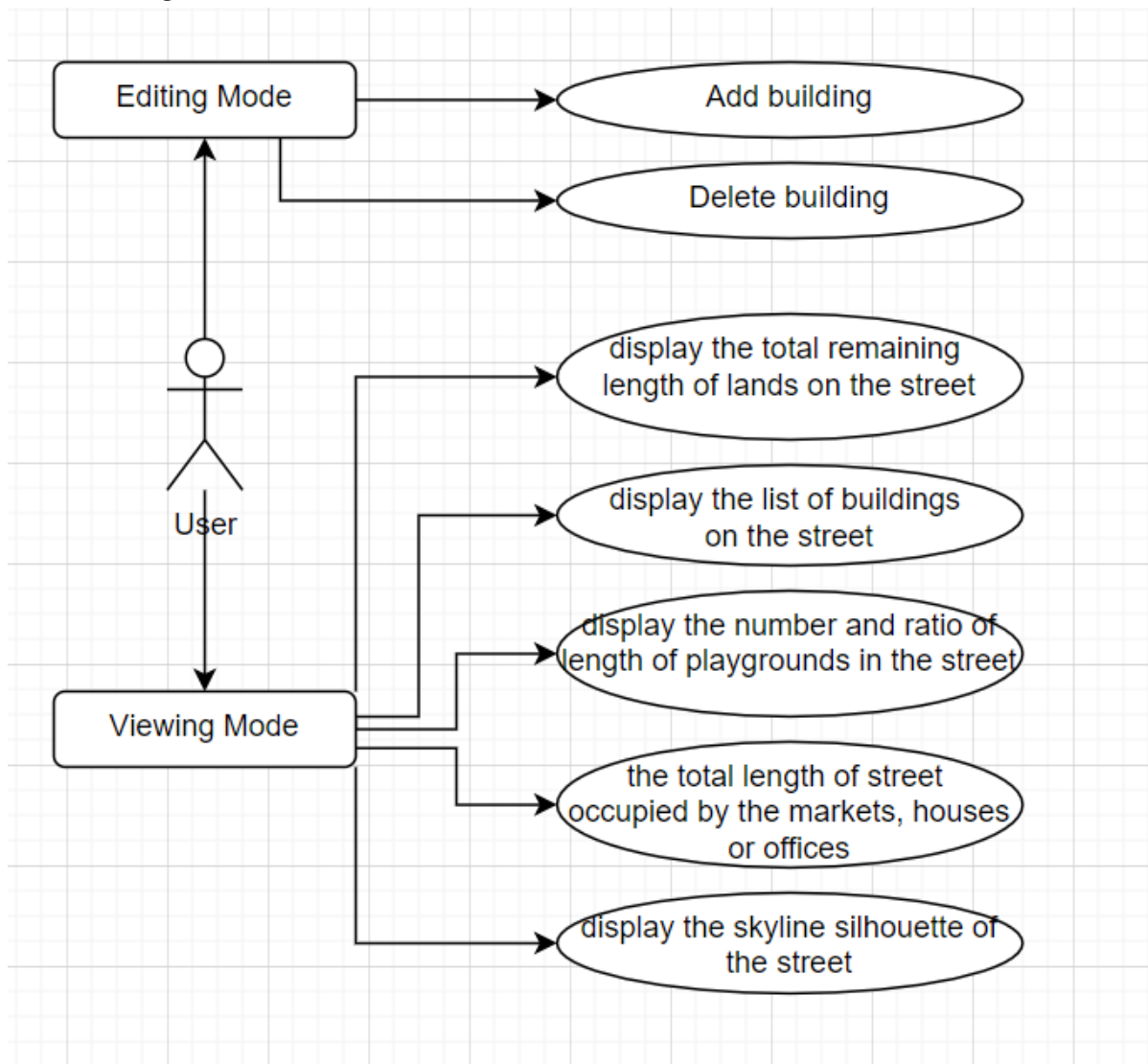Street Class which contains an array of building class

Building Super class and 4 subclasses of it such as House, Market, Office and Playground. Building class has several variables which are common (length, height, starting point, ending point, owner). Other subclasses has unique variables for each of themselves.

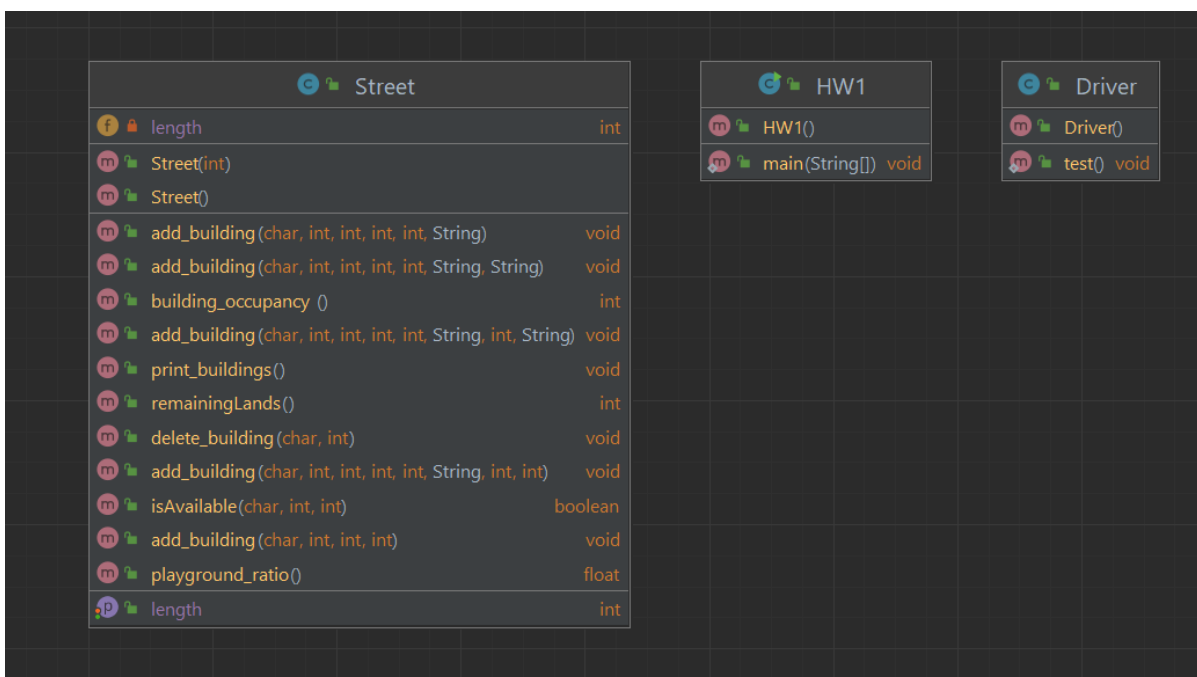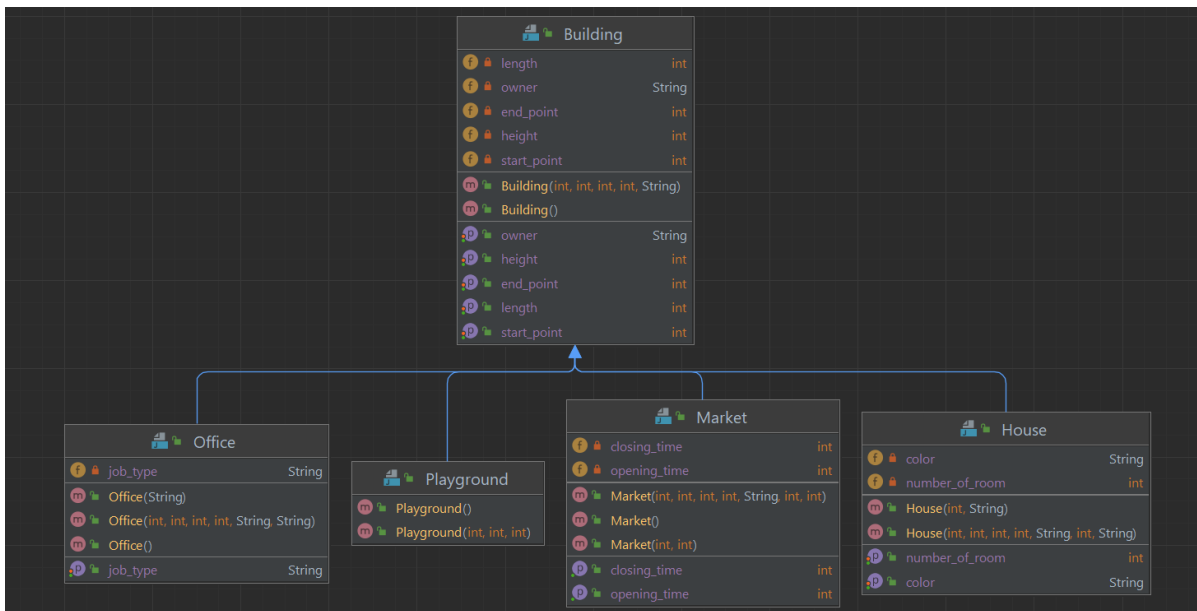The only non-working function is skyline silhouette at the moment.

This software is reliable, scalable, reusable.

2. USE CASE AND CLASS DIAGRAMS

Use Case Diagram

Class Diagram

Building

f	length	int
f	owner	String
f	end_point	int
f	height	int
f	start_point	int
m	Building(int, int, int, int, String)	
m	Building()	
p	owner	String
p	height	int
p	end_point	int
p	length	int
p	start_point	int

Office

f	job_type	String
m	Office(String)	
m	Office(int, int, int, int, String, String)	
m	Office()	
p	job_type	String

Playground

m	Playground()
m	Playground(int, int, int)

Market

f	closing_time	int
f	opening_time	int
m	Market(int, int, int, int, String, int, int)	
m	Market()	
m	Market(int, int)	
p	closing_time	int
p	opening_time	int

House

f	color	String
f	number_of_room	int
m	House(int, String)	
m	House(int, int, int, int, String, int, String)	
p	number_of_room	int
p	color	String

Street

f	length	int
m	Street(int)	
m	Street()	
m	add_building(char, int, int, int, int, String)	void
m	add_building(char, int, int, int, int, String, String)	void
m	building_occupancy()	int
m	add_building(char, int, int, int, int, String, int, String)	void
m	print_buildings()	void
m	remainingLands()	int
m	delete_building(char, int)	void
m	add_building(char, int, int, int, int, String, int, int)	void
m	isAvailable(char, int, int)	boolean
m	add_building(char, int, int, int)	void
m	playground_ratio()	float
p	length	int

HW1

m	HW1()
m	main(String[]) void

Driver

m	Driver()
m	test() void

Street has Arraylist of building as parameter for part a.

Street has Linkedlist of building as parameter for part b.

Street has LDLinkedlist of building as parameter for part c.

3. PROBLEM SOLUTION APPROACH

City planning software is needed for designing a small one street town which has two sides, right and left. I implemented 2 essential classes in order to represent the town in object oriented concept. Which are Street and Building classes. Building class has 4 subclasses such as House, Market, Office and Playground, and subclasses has constructors ,setters and getters only. Street class has 2 required data structure of Building class for right side and left side of the street, and has methods in order to represent the town (add, delete, print buildings and other informational methods).

4. TEST CASES

```
Length of the street : 100
Checking if the land between 80 and 90 in right side is available:
true
-----List of Buildings-----
Left side:
House owned by Mehmet
Playground owned by Government
Market owned by Bim
Office owned by PTT
Right side:
House owned by Yunus
Market owned by Emre
Office owned by Michael
Playground owned by Government
House owned by Ahmet
Market owned by Erdal
Office owned by Kenan
Playground owned by Government
--------------------------
Total remaining length of lands on the street: 108
3 playgrounds in the street, the ratio is 0.2
Total length of street occupied by the markets, houses or offices: 52
Kenan's office has been deleted
Mehmet's house has been deleted
Playground has been deleted
-----List of Buildings-----
Left side:
Market owned by Bim
Office owned by PTT
Right side:
House owned by Yunus
Market owned by Emre
Office owned by Michael
Playground owned by Government
House owned by Ahmet
Market owned by Erdal
Playground owned by Government
--------------------------
Total remaining length of lands on the street: 120
2 playgrounds in the street, the ratio is 0.16
Total length of street occupied by the markets, houses or offices: 48
Driver function is executed, Main program is being evaluated from now on:
Please enter the length of the street:
|
```

Other Test Cases

```
Please enter the length of the street:
100
*****MENU*****
1-Editing Mode
2-Viewing Mode
3-Exit
*************
1

****Editing Mode****
1-Add Building
2-Delete Building
3-Return to Main Menu
*************
1
Please chose the type of building you want to add:
1-House
2-Market
3-Office
4-Playground
1
Please enter side(r/l), length, height, start point, owner, number of room and color in order
side:r
length:10
height:7
start:5
owner:yunus
number of rooms:5
color:red

****Editing Mode****
1-Add Building
2-Delete Building
3-Return to Main Menu
*************
```

```
****Editing Mode****
1-Add Building
2-Delete Building
3-Return to Main Menu
*************
1
Please chose the type of building you want to add:
1-House
2-Market
3-Office
4-Playground
1
Please enter side(r/l), length, height, start point, owner, number of room and color in order
side:r
length:5
height:8
start:3
owner:Ahmet
number of rooms:4
color:purple
This land is not available
```

Test case for intersecting buildings

Test case for invalid inputs

```
*****MENU*****
1-Editing Mode
2-Viewing Mode
3-Exit
**************
2


****Viewing Mode****
1-Total Remaining length of Lands
2-List of Buildings
3-Number and Ratio of Length of Playgrounds
4-Occupied length by buildings
5-Skyline Silhouette
6-Return to Main Menu
**************
2
-----List of Buildings-----
Left side:
Market owned by Ahmet
House owned by Emre
Right side:
--------------------------
Enter any number to go back:
|
****Viewing Mode****
1-Total Remaining length of Lands
2-List of Buildings
3-Number and Ratio of Length of Playgrounds
4-Occupied length by buildings
5-Skyline Silhouette
6-Return to Main Menu
**************
3
0 playgrounds in the street, the ratio is  0.0
Enter any number to go back:
```

5. RUNNING AND RESULTS

```
-----List of Buildings-----
Left side:
Market owned by Ahmet
House owned by Emre
Right side:
Office owned by yunus
Playground owned by Government
--------------------------
Enter any number to go back:
```

```
****Editing Mode****
1-Add Building
2-Delete Building
3-Return to Main Menu
**************
2
Please enter the side and building number:
Side(r/l):r
Number:0
```

```
-----List of Buildings-----          ****Viewing Mode****
Left side:                           1-Total Remaining length of Lands
Market owned by Ahmet                2-List of Buildings
House owned by Emre                  3-Number and Ratio of Length of Playgrounds
Right side:                          4-Occupied length by buildings
Playground owned by Government       5-Skyline Silhouette
                                     6-Return to Main Menu
-------------------------            **************
                                     3
Enter any number to go back:         1 playgrounds in the street, the ratio is  0.05
                                     Enter any number to go back:
```

6. TIME COMPLEXITY

Add method;
Hw1 : O(1) because has random access and constant time
Part a : O(1) because has random access and constant time
Part b : O(n) because iterates to the end of LinkedList
Part c : O(n) because iterates to the end of LinkedList
Remove method;
Hw1 : O(n) because shifts the rest by one (if 8th element is removed 9th becomes new 8th)
Part a : O(n)
Part b : O(n) because iterates to the index of item
Part c : O(n) because iterates to the index of item
Getters;
Hw1 : O(1) because has random access
Part a : O(1) because has random access
Part b : O(n) because iterates to the index of item
Part c : O(n) because iterates to the index of item

Rest of the program has same time complexity but according to the time complexities above, if a method has O(n) complexity it would be O(n) times Time complexity of respective data structure's getter:
Hw1 : O(n) x O(1) = O(n)
Part a : O(n) x O(1) = O(n)
Part b : O(n) x O(n) = O(n^2)
Part c : O(n) x O(n) = O(n^2)

```
Elapsed time for add_building function in hw1 in miliseconds: 33.805
Elapsed time for add_building function in part 1 in miliseconds: 30.696
Elapsed time for add_building function in part 2 in miliseconds: 40.386
Elapsed time for add_building function in part 3 in miliseconds: 40.804
```

Add_building is called 12 times in each test but since the linkedlists are empty at first it works faster than expected.