

GTU Department of Computer Engineering

CSE 222/505 - Spring 2022

Homework #3 Report

Serhat SARI
200104004028

1. SYSTEM REQUIREMENTS

This is a city planning software that will be used for designing a small one street town.
We can create our town with Streets.
So first, we need to have Street objects to hold the buildings.

```
public Street(int userLength)
```

To create a street object, we should give the length of the street as parameter.

```
Street mainStreet = new Street(50);
```

We can create a Street object like this with a given length parameter.

User can add a building to the street. User can delete any building on the street.

```
public void addBuilding(Building newBuilding)
```

To add building to the street, building object should be given as parameter.

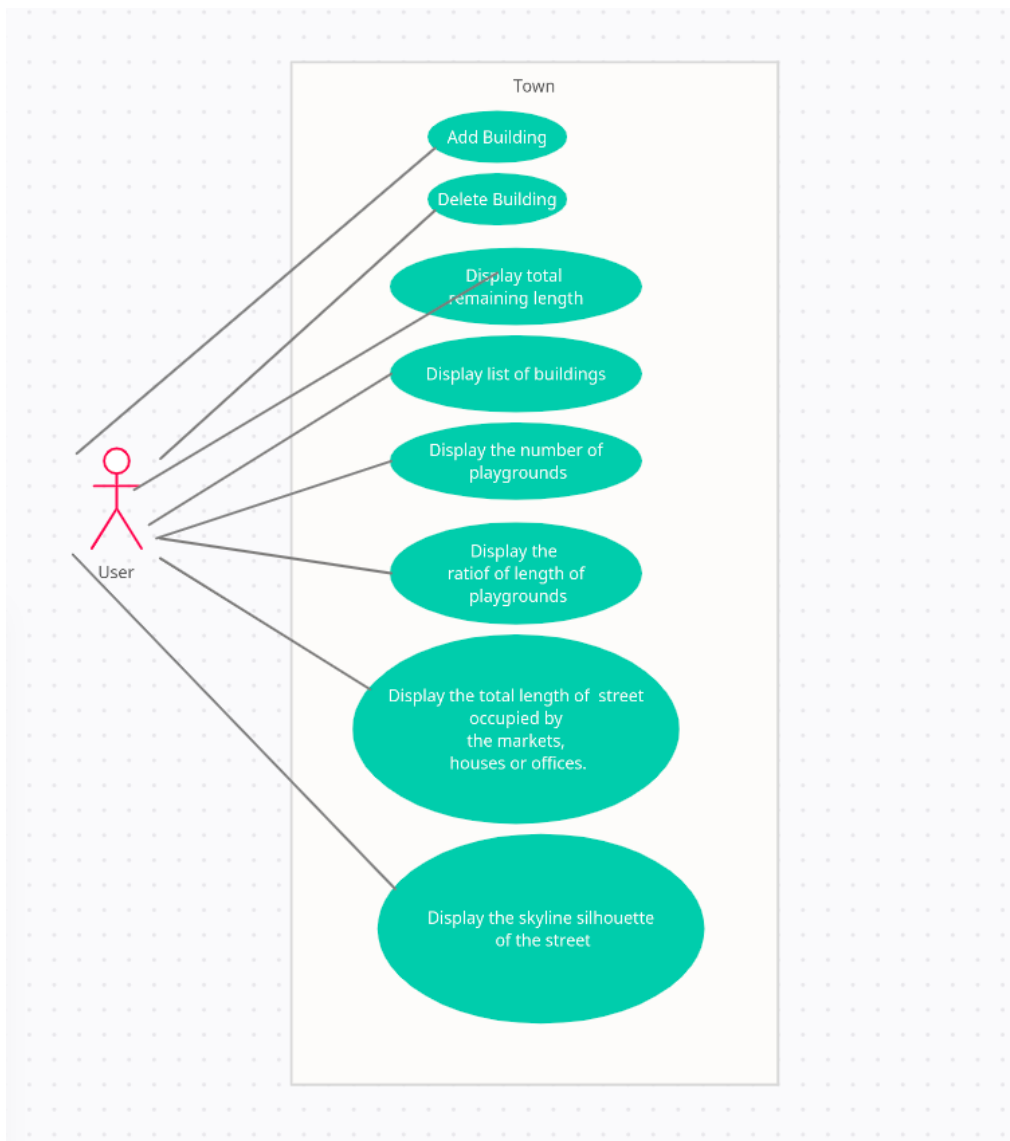
```
public void deleteBuilding(int index)
```

To delete a building on the street, index of the building should be given as parameter

User has 2 modes:

- 1: Edit mode, which user can add or delete building,
- 2: View mode, which user can display information about streets

2. USE CASE AND CLASS DIAGRAMS



3. PROBLEM SOLUTION APPROACH

In this homework, we are asked to reimplement the first homework with 3 different data structures. Implementing ArrayList and LinkedList were easiest part of the homework. All i need to do was replace all use of array with them.

Hard part of the homework was to create our own Linked List which is LDLinkedList. I have done a lot of research while implementing it. I have rewatched the lessons to see proper way of implementing custom linked list class. I even checked the official implementation of linked list class. After this research, it was easy for me to implement my own linked list class. In my linked list class, i had to implement 2 inner classes which were: Node class to represent the node of the list, ListIterator class to create an iterator of the class. I checked the official ListIterator class like what methods it contains. I implemented all methods of the ListIterator class.

After implementation, i started to calculate complexity of the classes, I calculated how much time requires for every methods in the classes. Then, i compare them with the theoretical result.

4. TEST CASES

Create a Street

```
Street mainStreet = new Street(50);
```

Create a Building Object (House, Office, Market, Playground)

```
House house = new House(1, 10, 15, "Right", 3, "Yellow", "Serhat", 100);
```

Add Buildings to the Street

```
mainStreet.addBuilding(house);
```

Delete Building From the Street

```
mainStreet.deleteBuilding(2);
```

Display the skyline silhouette of the street

```
mainStreet.printStreet();
```

Display the total remaining length of lands on the street

```
mainStreet.printRemainingLength();
```

Display the number and ratio of length of playgrounds in the street

```
mainStreet.printPlaygroundInfo();
```

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

```
mainStreet.printTotalLengths();
```

5. RUNNING AND RESULTS

Create a Street

```
CITY PLANNING SOFTWARE
Enter the length of the street you want to create: █
```

Main Menu

```
MAIN MENU
1: Edit Mode
2: View Mode
3: Focus BUIlding
4: Exit Program
Enter your input (1-4): █
```

Edit Menu

```
EDIT MENU
1: Add Building
2: Delete Building
3: Exit Edit Menu
What do you wanna do (1-3): █
```

Choose building type to add

```
BUILDING TYPES
1- House
2- Office
3- Market
4- Playground
5- Cancel adding building
Which building you want to add to your street(1-4): █
```

Add Building

```
Enter the position of the house: 30
Enter the length of the house: 30
Enter the height of the house: 30
1: Left Side
2: Right Side
Which side of the street do you wanna add to house (1,2):2
Enter the number of rooms of the house: 3
Enter the color of the house: red
Enter the owner of the house: serhat sari
BUILDING HAS BEEN ADDED TO STREET.
```

Delete Building

```
LIST OF BUILDINGS
1: Type = House, Street Side = Right, Position = 30

Enter the number of the building to delete: 1
```

View Menu

```
VIEW MENU
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 lenth 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: Exit View Menu
What do you wanna do (1-6): █
```

Display remaining length

```
Remaining Length: 78
```

Display the list of the buildings

```
1.BUILDING:

HOUSE INFO:
Position: 2
Length: 2
Height: 12
Street Side: Right
Number of Rooms: 3
Color: red
Owner: serhat
```

Display the number and ratio of playgrounds

```
Number of playgrounds: 1
Ratio of lenth of playgrounds in the street: 0.15
```

Display the total lenthgs of the buildings

```
TOTAL LENGTHS
Total Length of Markets: 0
Total Length of Houses: 2
Total Length of Offices: 0
```

Display the skyline silhouette of the street

```
Skyline silhouette of the street

*****
*           *
*           *
*           *
*           *
*           *
*           *
*           *
*           *
*           *
*           *
*           *
*           *
#####
```

Focus Building

```
LIST OF BUILDINGS
1: Type = House, Street Side = Right, Position = 2
2: Type = Playground, Street Side = Right, Position = 1

Which building do you want to focus on: 1

Focusing on house:
Owner of the house: serhat
```


TIME COMPLEXITY OF THE CLASSES

$O(N)$	$O(N)$	$O(N)$	$O(N)$
$O(N)$	$O(N)$	$O(N)$	$O(N)$
$O(N)$	$O(N)$	$O(N)$	$O(N)$
$O(N)$	$O(N)$	$O(N)$	$O(N)$
$O(N)$	$O(N)$	$O(N)$	$O(N)$
$O(N)$	$O(N)$	$O(N)$	$O(N)$
$O(N^2)$	$O(N^2)$	$O(N^2)$	$O(N^2)$
$O(N^2)$	$O(N^2)$	$O(N^2)$	$O(N^2)$
$O(N^2)$	$O(N^2)$	$O(N^2)$	$O(N^2)$

COMPARING REAL RESULTS:

ADD BUILDING : LINKED LIST > ARRAY > ARRAY LIST > CUSTOM LINKED LIST

DELETE BUILDING: ARRAY LIST > LINKED LIST > CUSTOM LINKED LIST > ARRAY

PRINT BUILDING: ARRAY LIST > ARRAY > CUSTOM LINKED LIST > LINKED LIST