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

Muhammed Sinan Pehlivanoğlu
1901042664

1. SYSTEM REQUIREMENTS

There must be Street created before User. User must be enter the length of the street. Length of the building is must be valid.

Another requirement is the building. Building must be unique according to its location. No building should in same location. The building must be within the street length.

The building array memory is increased by twice if the array's capacity is full. And if the total building number is equal to quarter of capacity, the array capacity is decreased half.

OS requirements:

Operating System should have openjdk 11.0.13 2021-10-19 LTS or above and Jre for OpenJDK Runtime Environment Corretto-11.0.13.8.1 (build 11.0.13+8-LTS) or above to run program.

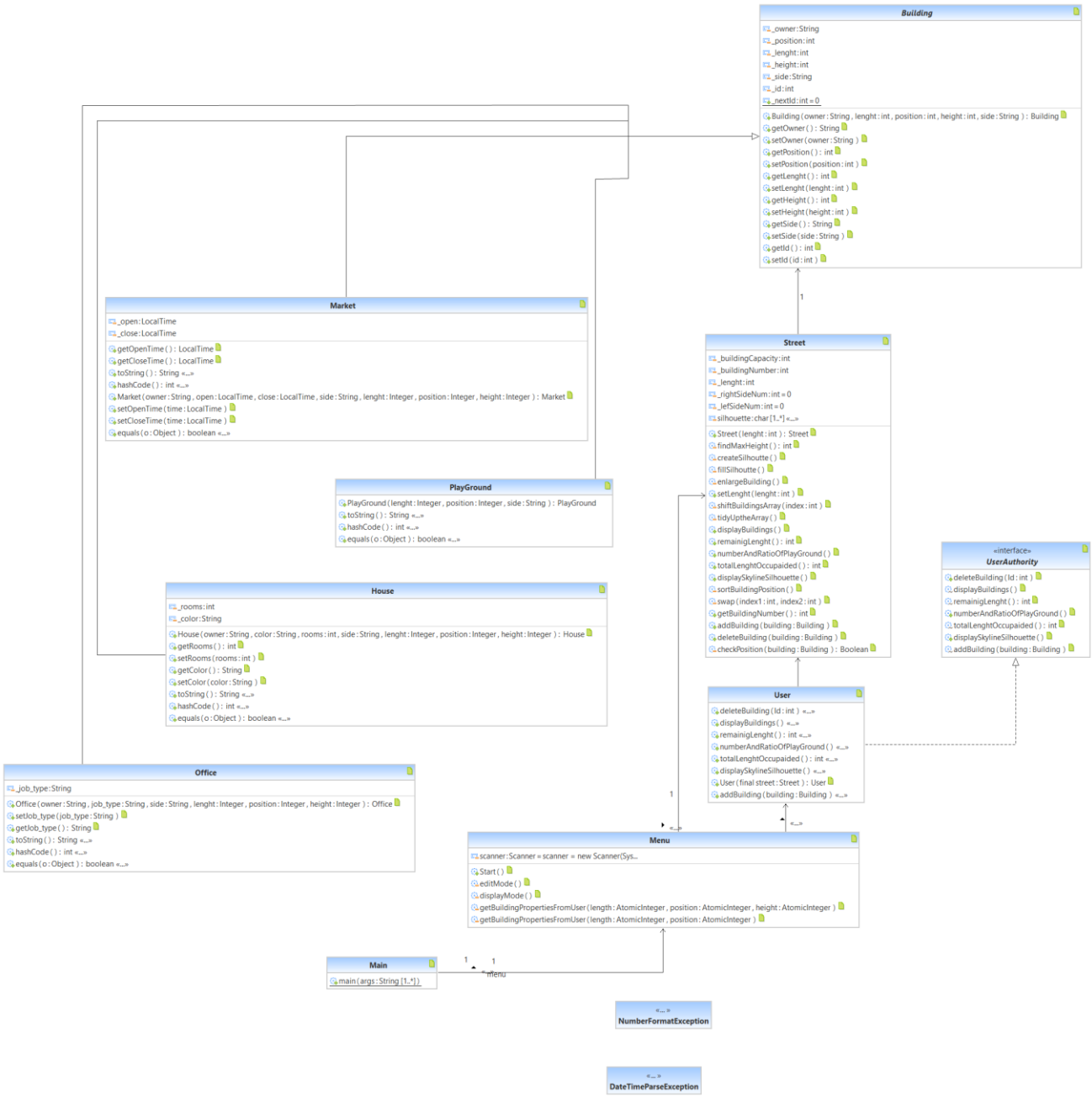
There will be some mistakes while running on jdk and jre below this version.

There are 4 Version of Program.
V0 is primitive version. İt uses array.
V1 uses arraylist.
V2 uses LinkedList.
V3 uses LDLinkedList.

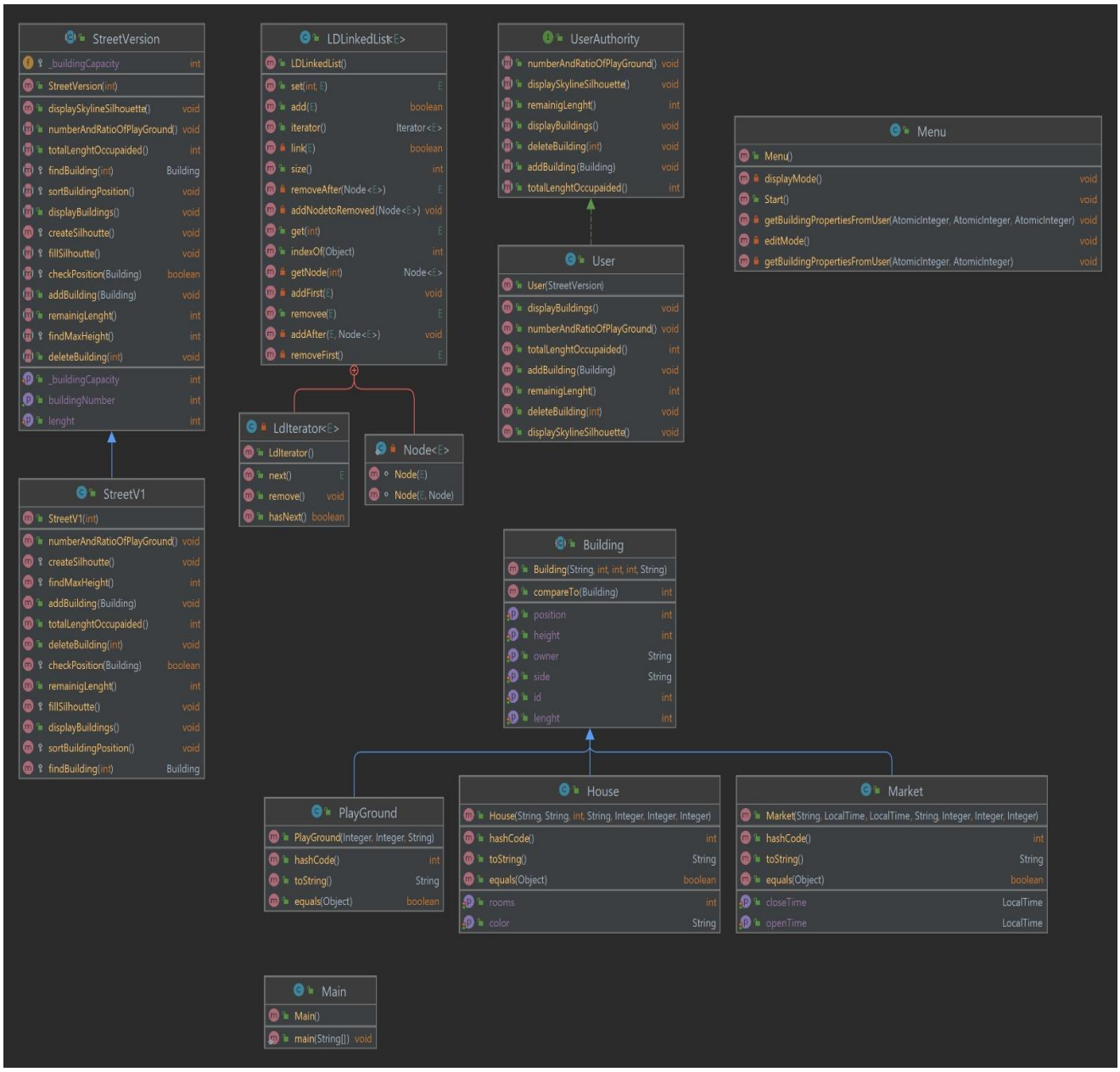
The user need to enough space to store datas of building.

2.USE CASE AND CLASS DIAGRAMS

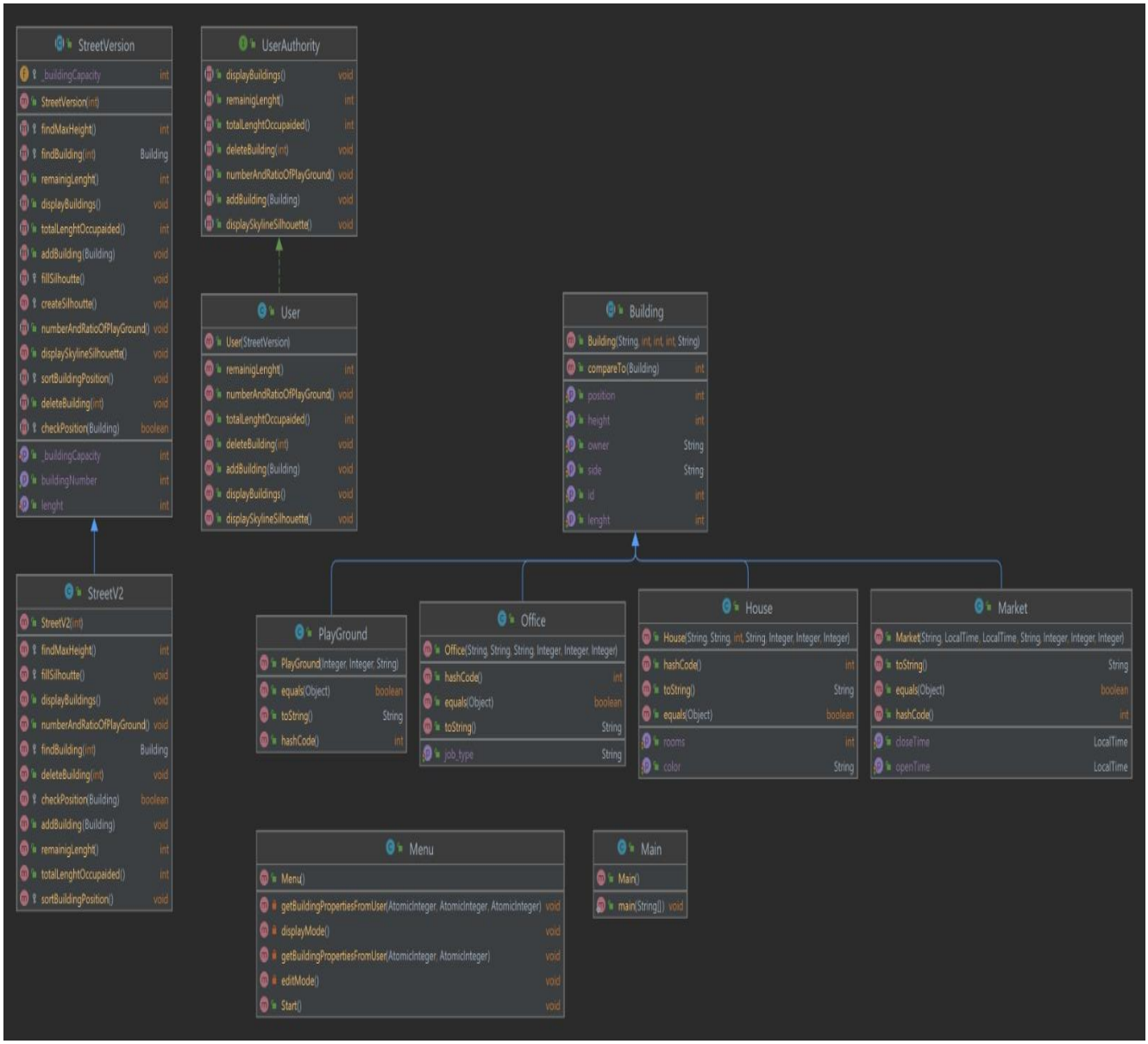
VERSION0



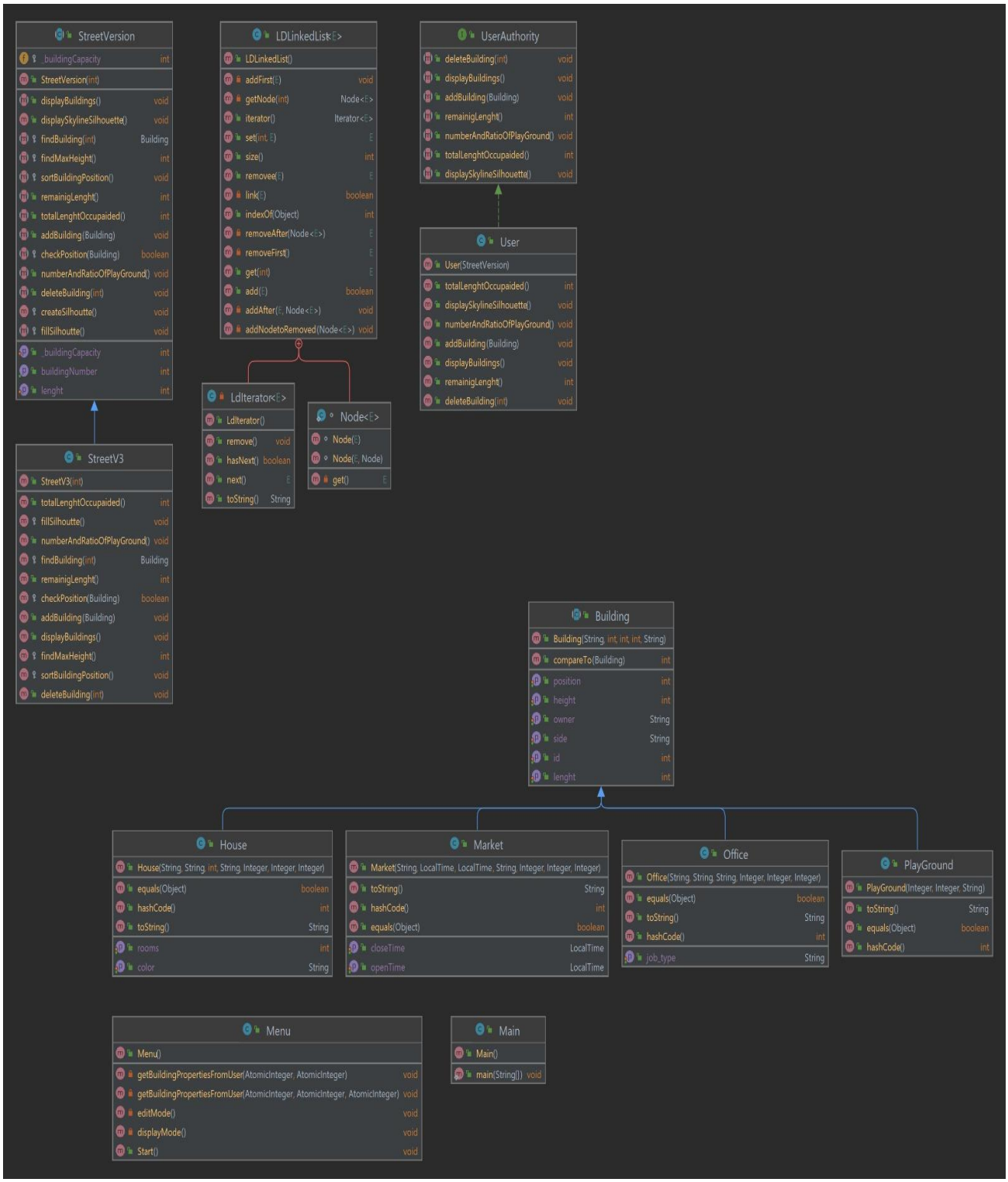
VERSION 1


StreetVersion
_buildingCapacity int
StreetVersion(int)
displaySkylineSilhouette() void
numberAndRatioOfPlayGround() void
totalLenghtOccupaided() int
findBuilding(int) Building
sortBuildingPosition() void
displayBuildings() void
createSilhoutte() void
fillSilhoutte() void
checkPosition(Building) boolean
addBuilding(Building) void
remainigLenght() int
findMaxHeight() int
deleteBuilding(int) void
_buildingCapacity int
buildingNumber int
lenght int
LDLinkedList<E>
LDLinkedList()
set(int, E) E
add(E) boolean
iterator() Iterator<E>
link(E) boolean
size() int
removeAfter(Node<E>) E
addNodetoRemoved(Node<E>) void
get(int) E
indexOf(Object) int
getNode(int) Node<E>
addFirst(E) void
removee(E) E
addAfter(E, Node<E>) void
removeFirst() E
UserAuthority
numberAndRatioOfPlayGround() void
displaySkylineSilhouette() void
remainigLenght() int
displayBuildings() void
deleteBuilding(int) void
addBuilding(Building) void
totalLenghtOccupaided() int
Menu
Menu()
displayMode() void
Start() void
getBuildingPropertiesFromUser(AtomicInteger, AtomicInteger, AtomicInteger) void
editMode() void
getBuildingPropertiesFromUser(AtomicInteger, AtomicInteger) void
User
User(StreetVersion)
displayBuildings() void
numberAndRatioOfPlayGround() void
totalLenghtOccupaided() int
addBuilding(Building) void
remainigLenght() int
deleteBuilding(int) void
displaySkylineSilhouette() void
LdIterator<E>
LdIterator()
next() E
remove() void
hasNext() boolean
Node<E>
Node(E)
Node(E, Node)
StreetV1
StreetV1(int)
numberAndRatioOfPlayGround() void
createSilhoutte() void
findMaxHeight() int
addBuilding(Building) void
totalLenghtOccupaided() int
deleteBuilding(int) void
checkPosition(Building) boolean
remainigLenght() int
fillSilhoutte() void
displayBuildings() void
sortBuildingPosition() void
findBuilding(int) Building
Building
Building(String, int, int, int, String)
compareTo(Building) int
position int
height int
owner String
side String
id int
lenght int
PlayGround
PlayGround(Integer, Integer, String)
hashCode() int
toString() String
equals(Object) boolean
House
House(String, String, int, String, Integer, Integer, Integer)
hashCode() int
toString() String
equals(Object) boolean
rooms int
color String
Market
Market(String, LocalTime, LocalTime, String, Integer, Integer, Integer)
hashCode() int
toString() String
equals(Object) boolean
closeTime LocalTime
openTime LocalTime
Main
Main()
main(String[]) void

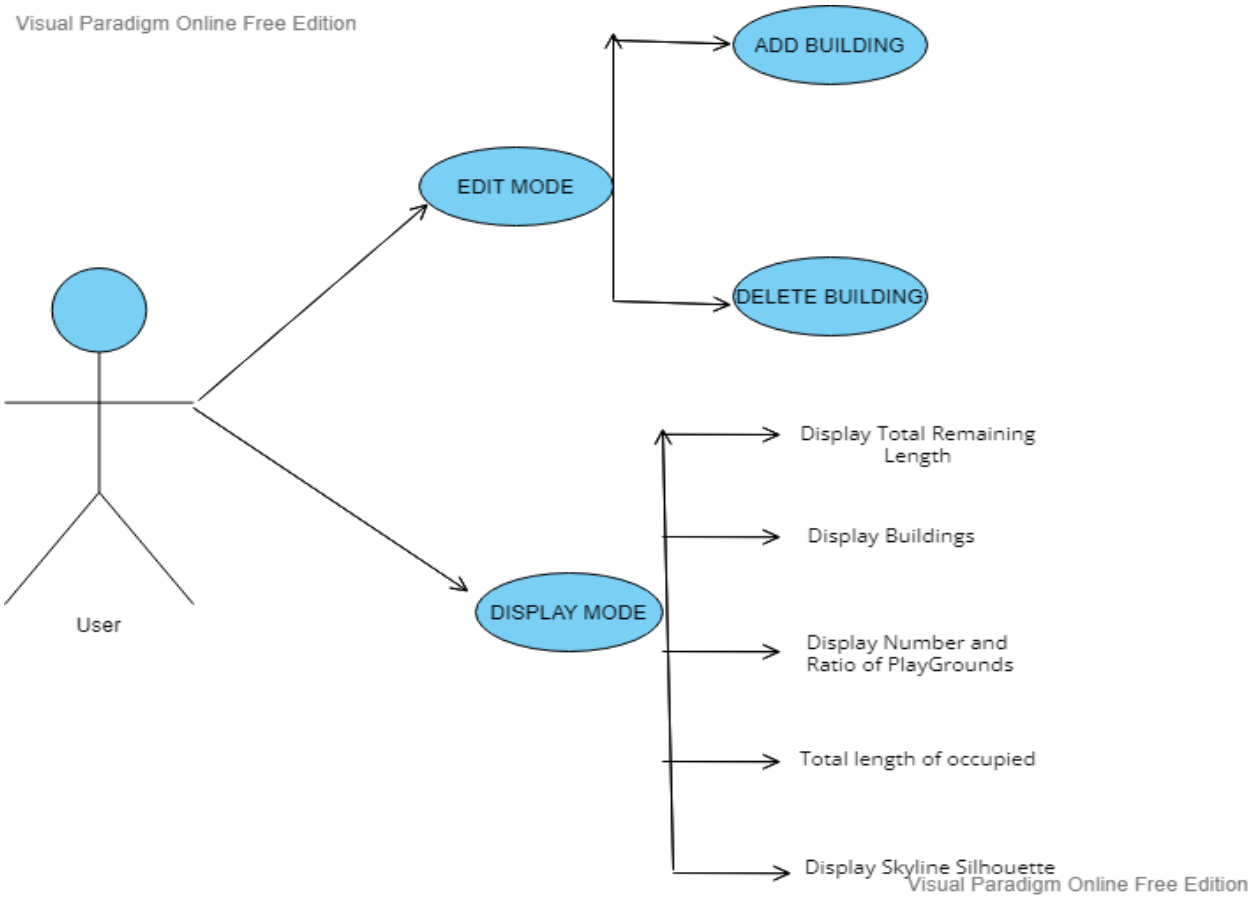
VERSION 2


StreetVersion
_buildingCapacity int
StreetVersion(int)
findMaxHeight() int
findBuilding(int) Building
remainigLenght() int
displayBuildings() void
totalLenghtOccupaided() int
addBuilding(Building) void
fillSilhoutte() void
createSilhoutte() void
numberAndRatioOfPlayGround() void
displaySkylineSilhouette() void
sortBuildingPosition() void
deleteBuilding(int) void
checkPosition(Building) boolean
_buildingCapacity int
buildingNumber int
lenght int
UserAuthority
displayBuildings() void
remainigLenght() int
totalLenghtOccupaided() int
deleteBuilding(int) void
numberAndRatioOfPlayGround() void
addBuilding(Building) void
displaySkylineSilhouette() void
User
User(StreetVersion)
remainigLenght() int
numberAndRatioOfPlayGround() void
totalLenghtOccupaided() int
deleteBuilding(int) void
addBuilding(Building) void
displayBuildings() void
displaySkylineSilhouette() void
Building
Building(String, int, int, int, String)
compareTo(Building) int
position int
height int
owner String
side String
id int
lenght int
StreetV2
StreetV2(int)
findMaxHeight() int
fillSilhoutte() void
displayBuildings() void
numberAndRatioOfPlayGround() void
findBuilding(int) Building
deleteBuilding(int) void
checkPosition(Building) boolean
addBuilding(Building) void
remainigLenght() int
totalLenghtOccupaided() int
sortBuildingPosition() void
PlayGround
PlayGround(Integer, Integer, String)
equals(Object) boolean
toString() String
hashCode() int
Office
Office(String, String, String, Integer, Integer, Integer)
hashCode() int
equals(Object) boolean
toString() String
job_type String
House
House(String, String, int, String, Integer, Integer, Integer)
hashCode() int
toString() String
equals(Object) boolean
rooms int
color String
Market
Market(String, LocalTime, LocalTime, String, Integer, Integer, Integer)
toString() String
equals(Object) boolean
hashCode() int
closeTime LocalTime
openTime LocalTime
Menu
Menu()
getBuildingPropertiesFromUser(AtomicInteger, AtomicInteger, AtomicInteger) void
displayMode() void
getBuildingPropertiesFromUser(AtomicInteger, AtomicInteger) void
editMode() void
Start() void
Main
Main()
main(String[]) void

VERSION 3


StreetVersion
_buildingCapacity int
StreetVersion(int)
displayBuildings() void
displaySkylineSilhouette() void
findBuilding(int) Building
findMaxHeight() int
sortBuildingPosition() void
remainigLenght() int
totalLenghtOccupaided() int
addBuilding(Building) void
checkPosition(Building) boolean
numberAndRatioOfPlayGround() void
deleteBuilding(int) void
createSilhoutte() void
fillSilhoutte() void
_buildingCapacity int
buildingNumber int
lenght int
LDLinkedList<E>
LDLinkedList()
addFirst(E) void
getNode(int) Node<E>
iterator() Iterator<E>
set(int, E) E
size() int
removee(E) E
link(E) boolean
indexOf(Object) int
removeAfter(Node<E>) E
removeFirst() E
get(int) E
add(E) boolean
addAfter(E, Node<E>) void
addNodetoRemoved(Node<E>) void
UserAuthority
deleteBuilding(int) void
displayBuildings() void
addBuilding(Building) void
remainigLenght() int
numberAndRatioOfPlayGround() void
totalLenghtOccupaided() int
displaySkylineSilhouette() void
User
User(StreetVersion)
totalLenghtOccupaided() int
displaySkylineSilhouette() void
numberAndRatioOfPlayGround() void
addBuilding(Building) void
displayBuildings() void
remainigLenght() int
deleteBuilding(int) void
LdIterator<E>
LdIterator()
remove() void
hasNext() boolean
next() E
toString() String
Node<E>
Node(E)
Node(E, Node)
get() E
StreetV3
StreetV3(int)
totalLenghtOccupaided() int
fillSilhoutte() void
numberAndRatioOfPlayGround() void
findBuilding(int) Building
remainigLenght() int
checkPosition(Building) boolean
addBuilding(Building) void
displayBuildings() void
findMaxHeight() int
sortBuildingPosition() void
deleteBuilding(int) void
Building
Building(String, int, int, int, String)
compareTo(Building) int
position int
height int
owner String
side String
id int
lenght int
House
House(String, String, int, String, Integer, Integer, Integer)
equals(Object) boolean
hashCode() int
toString() String
rooms int
color String
Market
Market(String, LocalTime, LocalTime, String, Integer, Integer, Integer)
toString() String
hashCode() int
equals(Object) boolean
closeTime LocalTime
openTime LocalTime
Office
Office(String, String, String, Integer, Integer, Integer)
equals(Object) boolean
toString() String
hashCode() int
job_type String
PlayGround
PlayGround(Integer, Integer, String)
toString() String
equals(Object) boolean
hashCode() int
Menu
Menu()
getBuildingPropertiesFromUser(AtomicInteger, AtomicInteger) void
getBuildingPropertiesFromUser(AtomicInteger, AtomicInteger, AtomicInteger) void
editMode() void
displayMode() void
Start() void
Main
Main()
main(String[]) void

2.2 USE CASE DIAGRAM



3.PROBLEM SOLUTION APPROACH

Our problem is that making the Street Planning Software that user can organize the street with two sides. Building has position, length and height properties. The house, office , market and Playground is derived from Building with some additional properties. We use polymorphisim in here.

In the V0 version the street hold buildings in an array .The Building Array is dyamically reallocated if the array capacity is full. In the V1 version the street hold building in an ArrayList. In the V2 the street hold building in LinkedList. In the V3 version the street hold building in an LDLİnkedList created by myself. LDlinked list

extends the AbstractList and implements List interface. All The Street class of versions extend StreetVersion abstract class. The mechanism of the LDLinked the nodes containing removed elements are moved to a separate Linked List for future use. When a new element is added, a node is obtained from this second list if one available. Of course, a new node should be created, otherwise.

When user wanted to add building to the street, program must check the position if Any building is already there. The building is compared with building in its side. There is a Id for each building. The Id is incremented by one each time building is added. Then assign these id to building to be added. First id is 0.

There are two mode. Display and edit mode. User can remove any building in street the querying with building id in edit mode.

User can display all building details in display mode, overridden toString() method is run in backend.

Displaying Skyline Silhouette is solved with some basic steps. Skyline silhouette is stored in 2D char array with initial value of every char is '.' . Firstly all of buildings is sorted by position. When the building is finded, array's value is changed to '#' charechter. Then these value is printed but not '.' Characters in the 2d array.

Other displaying operations can be run in display mode.

4.TEST CASES

- Create User with Street Length.

- Create 2 House, 2 Market, 2 Office, 2 Playground.

-Display created building details.

- Display building details using iterator. (This is mandatory except Version0).

-Delete some buildings from street.

-Display the total remaining length of lands on the street.

-Display the number and ratio of lenth of playgrounds in the street.

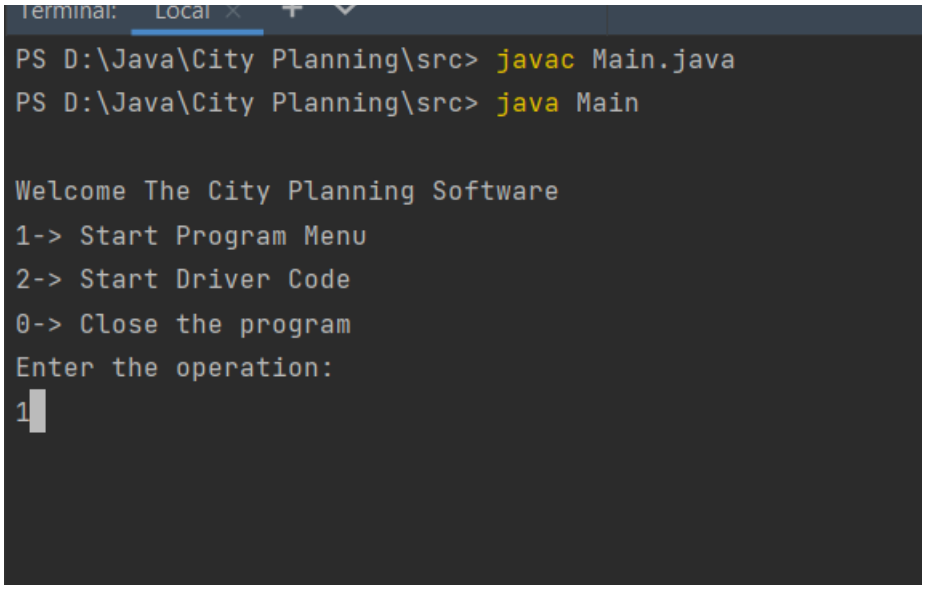
-Calculate the total length of street occupied by the markets, houses or offices.

-Display Skyline Silhouette.

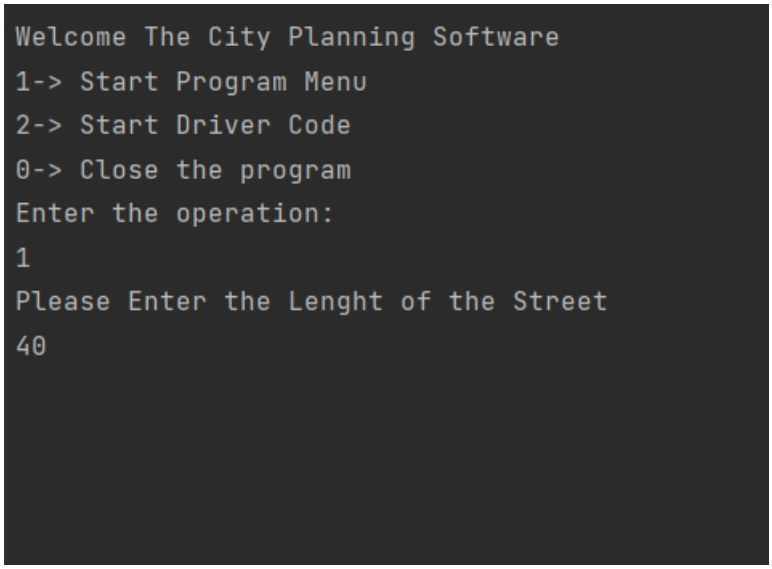
5.RUNNING AND RESULTS

There are most of operation's photos in this part.

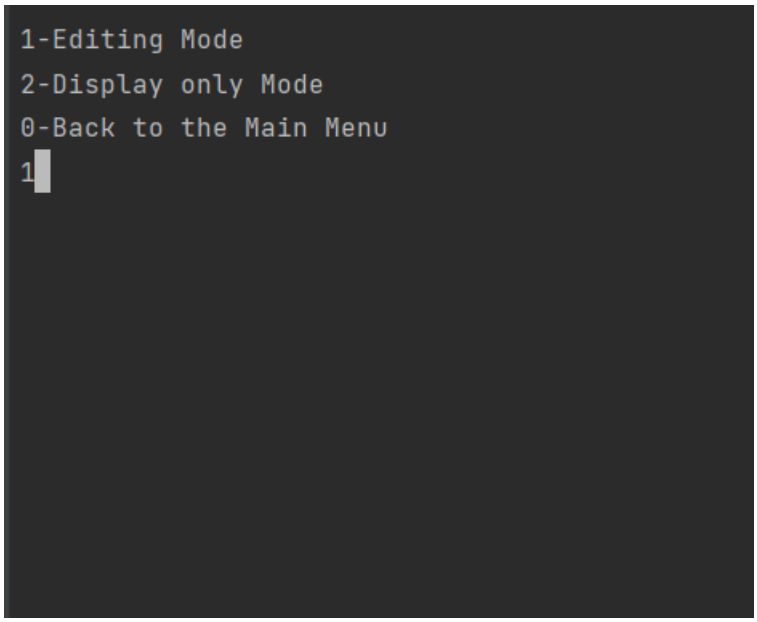
Main Menu:



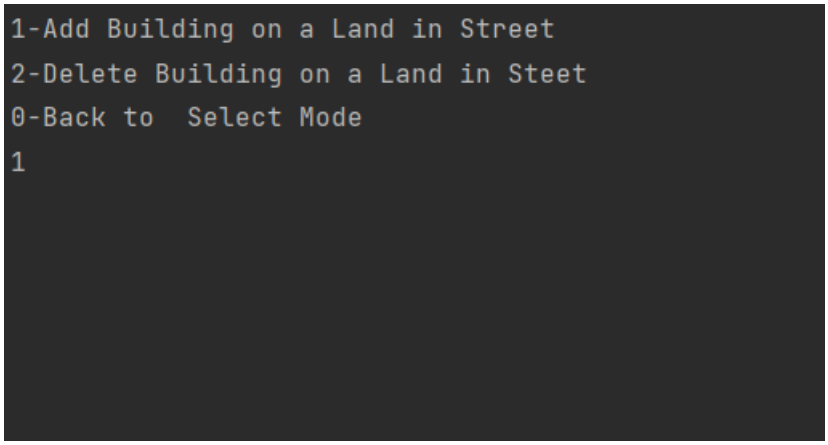
Getting Street Length



Choosing the Mode Menu



Adding, Deleting Operation Menu



Add House

```
1-Add House
2-Add Market
3-Add Office
4-Add Playground
Press another key to Edit Menu
1
```

```
Adding the House
Please Enter the Owner of House :  House1
Please Enter the Color of House : Red
Please Enter the Number of Rooms : 3
Please Enter the Position of the Building : 1
Please Enter the Lenght of the Building : 5
Please Enter the Height of the Building : 8
Please Enter the Side of the Street (right to r, left to l): r
House is added
1-Add Building on a Land in Street
2-Delete Building on a Land in Steet
0-Back to  Select Mode
```

Add Market

```
1-Add House
2-Add Market
3-Add Office
4-Add Playground
Press another key to Edit Menu
2
Adding the Market
Please Enter the Owner of Market :  Market1
Please Enter the Opening Time of Market (ex: 07:06) : 07:30
Please Enter the Closing Time  of Market (ex: 09:12) : 21:00
Please Enter the Position of the Building : 3
Please Enter the Lenght of the Building : 7
Please Enter the Height of the Building : 6
Please Enter the Side of the Street (right to r, left to l): l
Market is added
```

Add Office

```
1-Add House
2-Add Market
3-Add Office
4-Add Playground
Press another key to Edit Menu
3
Adding the Office
Please Enter the Owner of Office :  Office1
Please Enter the Job Type of Office : Software
Please Enter the Position of the Building : 7
Please Enter the Lenght of the Building : 4
Please Enter the Height of the Building : 7
Please Enter the Side of the Street (right to r, left to l): r
Office is added
1-Add Building on a Land in Street
2-Delete Building on a Land in Steet
0-Back to  Select Mode
```

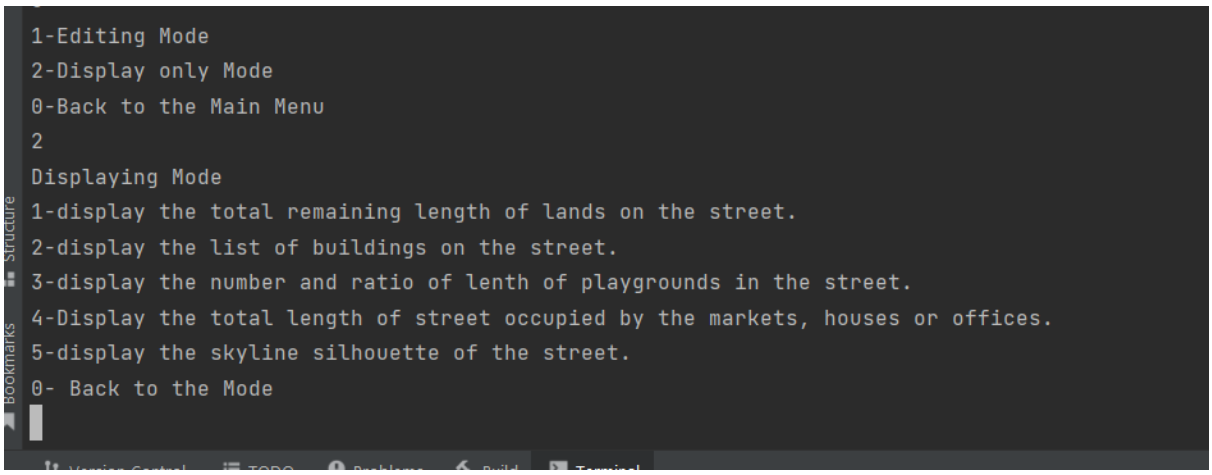
Add PlayGround

```
1-Add House
2-Add Market
3-Add Office
4-Add Playground
Press another key to Edit Menu
4
Adding the PlayGround
Please Enter the Position of the Building : 11
Please Enter the Lenght of the Building : 6
Please Enter the Side of the Street (right to r, left to l) : l
PlayGround is added
1-Add Building on a Land in Street
2-Delete Building on a Land in Steet
0-Back to  Select Mode
```

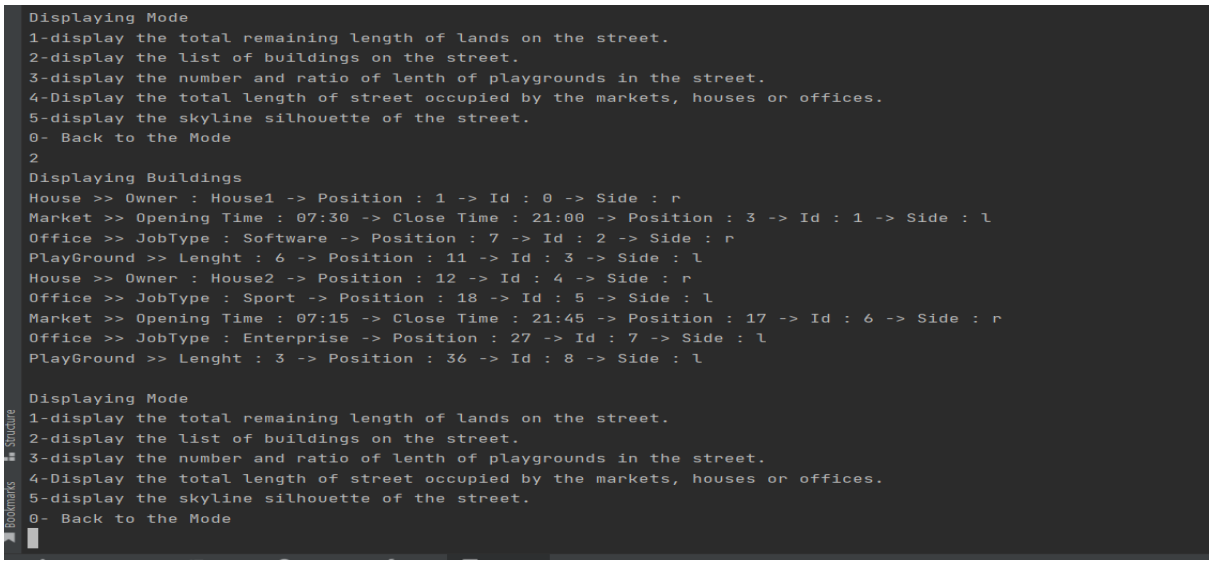
Invalid Location Input

```
1-Add House
2-Add Market
3-Add Office
4-Add Playground
Press another key to Edit Menu
1
Adding the House
Please Enter the Owner of House :  House2
Please Enter the Color of House : Blue
Please Enter the Number of Rooms : 3
Please Enter the Position of the Building : 4
Please Enter the Lenght of the Building : 3
Please Enter the Height of the Building : 10
Please Enter the Side of the Street (right to r, left to l) : r
There is no enough space to locate the House
1-Add Building on a Land in Street
2-Delete Building on a Land in Steet
0-Back to  Select Mode
```

Displaying Menu



Display Building Details (Some Building was added)



Total Remaining Lenght of Lands on the Street

```
Displaying Mode
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-Display the total length of street occupied by the markets, houses or offices.
5-display the skyline silhouette of the street.
0- Back to the Mode
1
Displaying Remaining Length
28

Displaying Mode
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-Display the total length of street occupied by the markets, houses or offices.
5-display the skyline silhouette of the street.
0- Back to the Mode
```

Number and Ratio of length of Playgrounds

```
Displaying Mode
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-Display the total length of street occupied by the markets, houses or offices.
5-display the skyline silhouette of the street.
0- Back to the Mode
3
Displaying the number and ratio of Playground
Total Number : 2 -> Ratio Of Playground:  0,112500

Displaying Mode
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-Display the total length of street occupied by the markets, houses or offices.
5-display the skyline silhouette of the street.
0- Back to the Mode
```

Occupied Land by Markets, Houses or Office

```
Displaying Mode
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-Display the total length of street occupied by the markets, houses or offices.
5-display the skyline silhouette of the street.
0- Back to the Mode
4
Displaying the Occupied Length
43

Displaying Mode
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-Display the total length of street occupied by the markets, houses or offices.
5-display the skyline silhouette of the street.
0- Back to the Mode
```

Skyline Silhouette

```
Displaying Mode
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-Display the total length of street occupied by the markets, houses or offices.
5-display the skyline silhouette of the street.
0- Back to the Mode
5
Displaying Skyline Silhoutte
 *****
 *   * ****
 *   ***  *      ***********
 *        * **** *         *
 *        * *  * *         ******
 *        * *  * *              *
 *        ***  ***              *   ***
 ********************************   ***
***********************************************
0----5----10----15----20----25----30----35----40
Displaying Mode
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-Display the total length of street occupied by the markets, houses or offices.
5-display the skyline silhouette of the street.
0- Back to the Mode
```

Delete Some Buildings

```
1-Add Building on a Land in Street
2-Delete Building on a Land in Steet
0-Back to  Select Mode
2
Please Enter the Id of the Building
0
Building is deleted
1-Add Building on a Land in Street
2-Delete Building on a Land in Steet
0-Back to  Select Mode
2
Please Enter the Id of the Building
3
Building is deleted
1-Add Building on a Land in Street
2-Delete Building on a Land in Steet
0-Back to  Select Mode
2
Please Enter the Id of the Building
4
Building is deleted
1-Add Building on a Land in Street
2-Delete Building on a Land in Steet
0-Back to  Select Mode
```

Display Buildings Array Remained

```
Displaying Mode
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-Display the total length of street occupied by the markets, houses or offices.
5-display the skyline silhouette of the street.
0- Back to the Mode
2
Displaying Buildings
Market >> Opening Time : 07:30 -> Close Time : 21:00 -> Position : 3 -> Id : 1 -> Side : l
Office >> JobType : Software -> Position : 7 -> Id : 2 -> Side : r
Office >> JobType : Sport -> Position : 18 -> Id : 5 -> Side : l
Market >> Opening Time : 07:15 -> Close Time : 21:45 -> Position : 17 -> Id : 6 -> Side : r
Office >> JobType : Enterprise -> Position : 27 -> Id : 7 -> Side : l
PlayGround >> Lenght : 3 -> Position : 36 -> Id : 8 -> Side : l

Displaying Mode
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-Display the total length of street occupied by the markets, houses or offices.
5-display the skyline silhouette of the street.
0- Back to the Mode
```

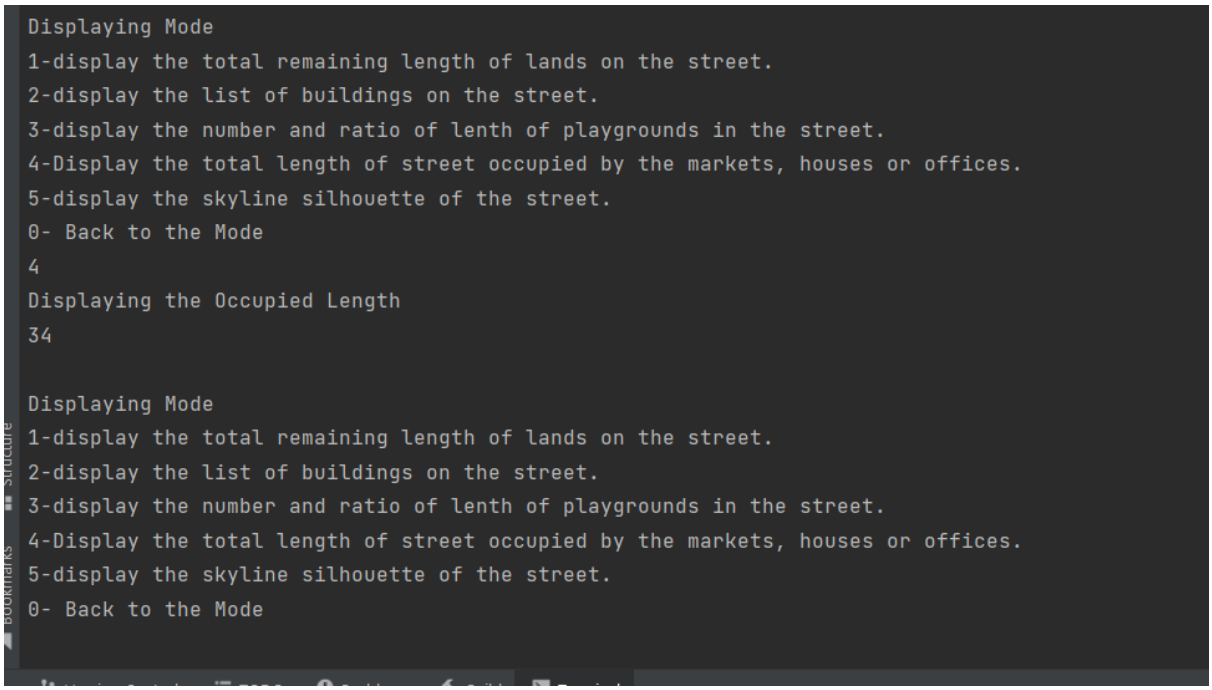
Total Remaining Length of Lands Remained on the Street

```
Displaying Mode
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-Display the total length of street occupied by the markets, houses or offices.
5-display the skyline silhouette of the street.
0- Back to the Mode
1
Displaying Remaining Length
43

Displaying Mode
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-Display the total length of street occupied by the markets, houses or offices.
5-display the skyline silhouette of the street.
0- Back to the Mode
```

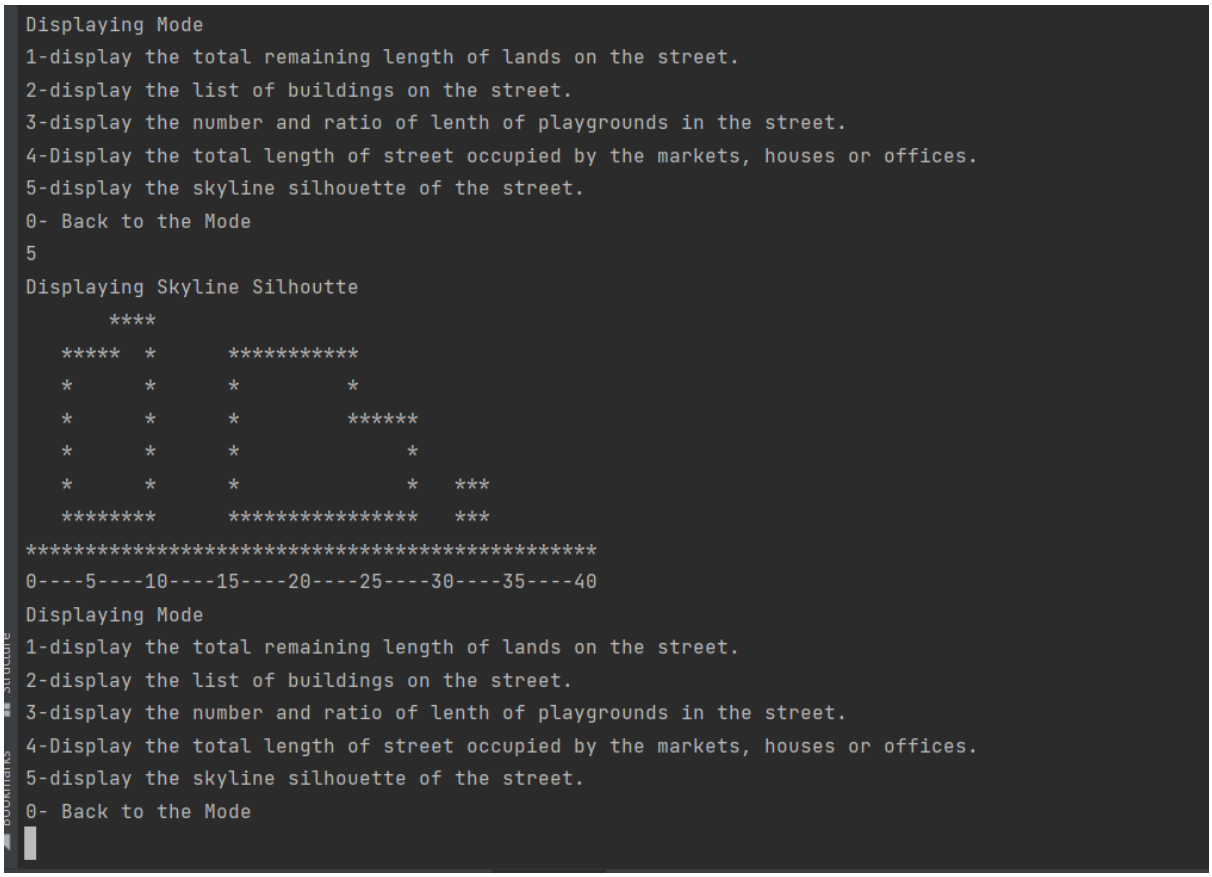
Number and Ratio of length of Playgrounds

```
Displaying Mode
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-Display the total length of street occupied by the markets, houses or offices.
5-display the skyline silhouette of the street.
0- Back to the Mode
3
Displaying the number and ratio of Playground
Total Number : 1 -> Ratio Of Playground:  0,037500

Displaying Mode
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-Display the total length of street occupied by the markets, houses or offices.
5-display the skyline silhouette of the street.
0- Back to the Mode
```

Occupied Land by Markets, Houses or Office



Skyline Silhouette



DISPLAYING BUILDING DETAILS USING ITERATOR(MONDATORY EXCEPT VERSION 0)

```
Iterator<Building> iter = buildings.iterator();
System.out.println("Display Building Using Iterator");
while(iter.hasNext()){
    System.out.println(iter);
    iter.next();
}
```

```
Display Building Using Iterator
House >> Owner : H1 -> Position : 0 -> Id : 0 -> Side : r
House >> Owner : H2 -> Position : 5 -> Id : 1 -> Side : r
Market >> Opening Time : 08:30 -> Close Time : 19:00 -> Position : 12 -> Id : 2 -> Side : r
Market >> Opening Time : 08:00 -> Close Time : 20:00 -> Position : 17 -> Id : 3 -> Side : l
Office >> JobType : Software -> Position : 12 -> Id : 0 -> Side : r
Office >> JobType : Sport -> Position : 10 -> Id : 4 -> Side : l
PlayGround >> Lenght : 3 -> Position : 21 -> Id : 5 -> Side : l
```