

*A project report on*

# **Traffic Simulation**

*Submitted by*

**Advait Patel**

*Student Id Number: - 1529198*

Masters of Computer Science

In

College of Computing and Digital Media

DePaul University

November-2015

## INDEX

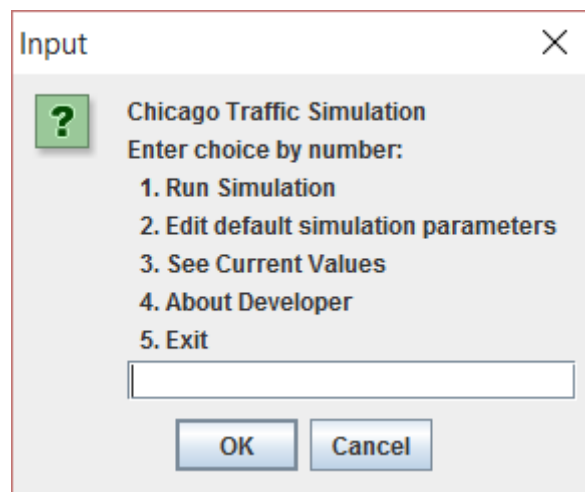
SR. No.	CONTENT	Page No
	<b>Part:- I Introduction</b>	
<b>1</b>	<b>Introduction to Project</b>	
	1.1 Project Description & User Interface	<b>1</b>
	<b>Part:- II System Design &amp; Analysis</b>	
<b>2</b>	<b>Project Planning</b>	
	2.1 Project Time Chart	<b>6</b>
<b>3</b>	<b>System Analysis</b>	
	3.1 Class Diagram	
	3.1.1 Package “main” class diagram	<b>7</b>
	3.1.2 Package “util” class Diagram	<b>7</b>
	3.1.3 Package “ui” class Diagram	<b>8</b>
	3.1.4 Package “Car Functionality” Diagram	<b>9</b>
	3.1.5 Package “Road Functionality”	<b>10</b>
	3.1.6 Package “ Light Functionality”	<b>11</b>
	3.2 Sequence Diagram	
	3.2.1 Sequence Diagram for Car Positioning	<b>12</b>
	3.3 Use Case Diagram	
	3.3.1 Use case Diagram for Traffic Simulator	<b>13</b>
<b>4</b>	<b>Appendix</b>	
	4.1 Notes on Pattern	<b>14</b>
	4.2 Successes & Failure	<b>14</b>

## Chapter -1 Introduction to Project

### 1.1 Project Description & User Interface

This report follows the design and implementation of a “Traffic Simulation” written in Java. It has 5 options as described below.

1. Run simulation
2. Edit simulation parameters
3. See current values
4. About Developer
5. Exit



Welcome Screen

Each single option has their sub options with some different kind of functionalities like, while user will press 1. Run simulation, the swing builder will create a swing model with component on it and will display the simulation.

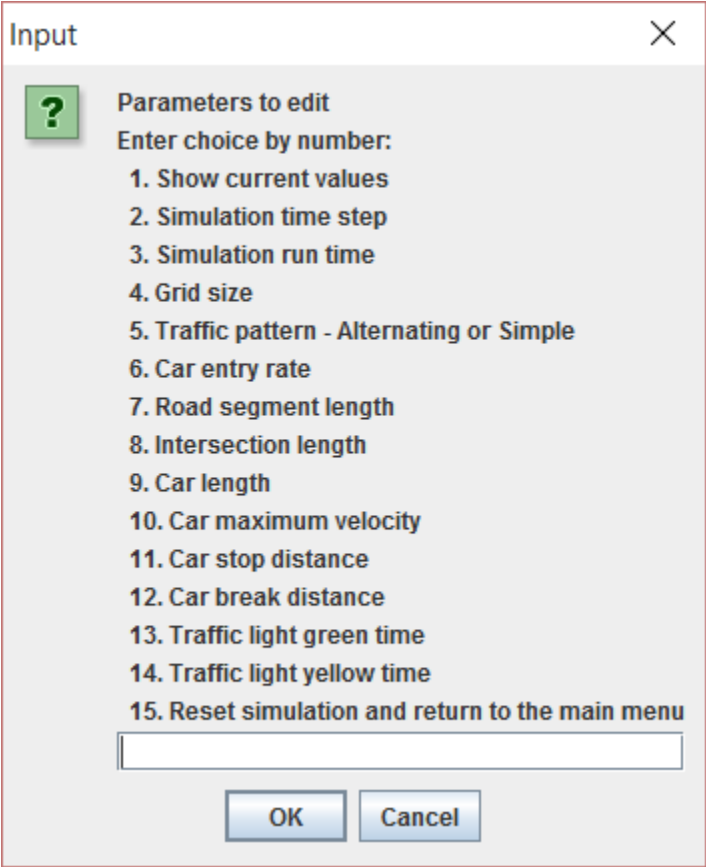


Traffic Simulation

Apart from this, user has privileges to change the default parameters as per their wish and requirements. As below screen shot shows user can handle 15 options inside the simulations.

1. Show current values
2. Simulation time step
3. Simulation run time
4. Grid size
5. Traffic Pattern
6. Car entry rate
7. Road segment length
8. Intersection length
9. Car length

10. Car maximum velocity
11. Car stop distance
12. Car break distance
13. Traffic light green time
14. Traffic light yellow time
15. Reset simulation and return to the main menu



Input

?

Parameters to edit

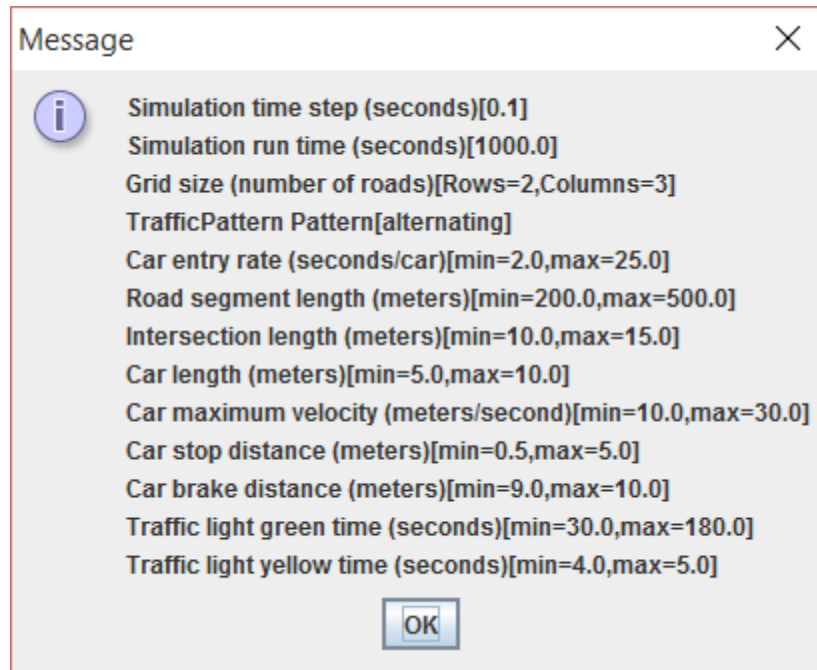
Enter choice by number:

1. Show current values
2. Simulation time step
3. Simulation run time
4. Grid size
5. Traffic pattern - Alternating or Simple
6. Car entry rate
7. Road segment length
8. Intersection length
9. Car length
10. Car maximum velocity
11. Car stop distance
12. Car break distance
13. Traffic light green time
14. Traffic light yellow time
15. Reset simulation and return to the main menu

OK Cancel

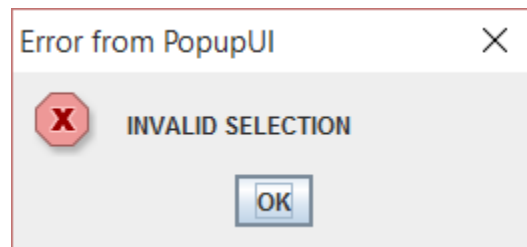
Parameters to edit

Any new user or fresh user can check current values of simulation by pressing 3. See current values

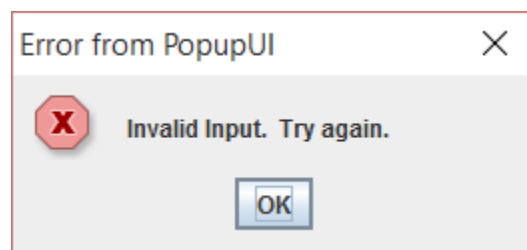


Current values

This system has some detailed functionalities like it can reply back with any invalid input or any blank input.

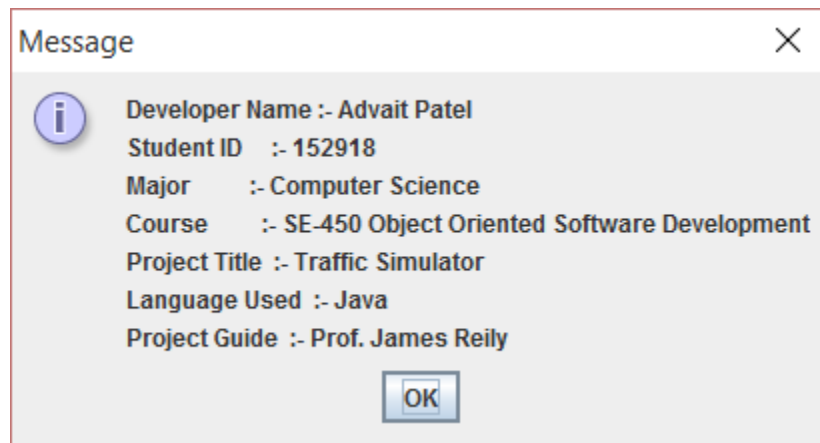


Blank input



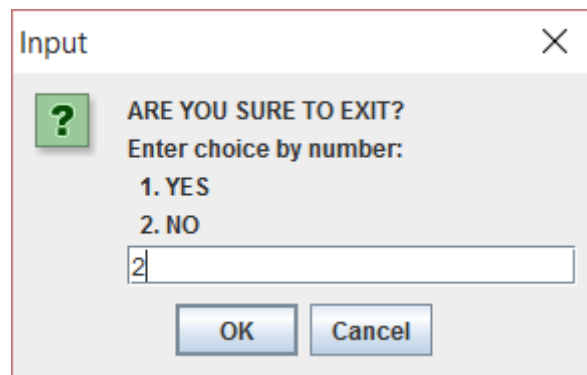
Invalid input

User can know about the developer of simulation by pressing 4. About Developer. It will display the information about its developer



About Developer/Project

If you mistakenly or knowingly press exit option it will ask about your confirmation.



Confirmation screen

## Chapter -2 Project Planning

### 2.1 Project Time Chart

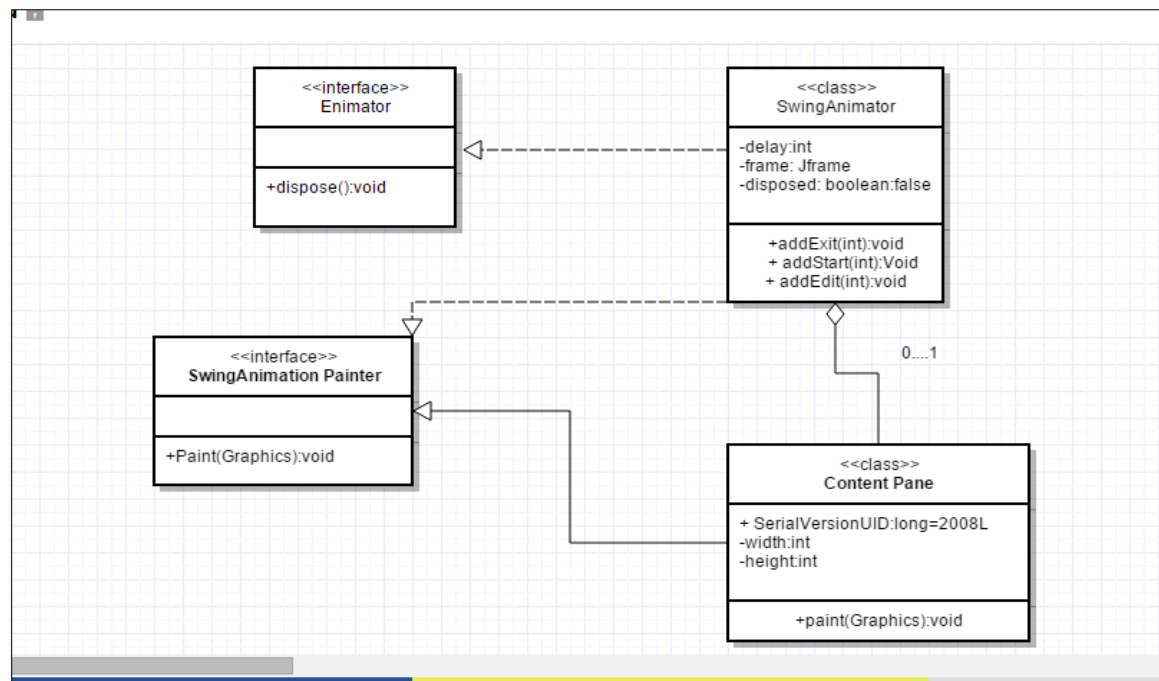
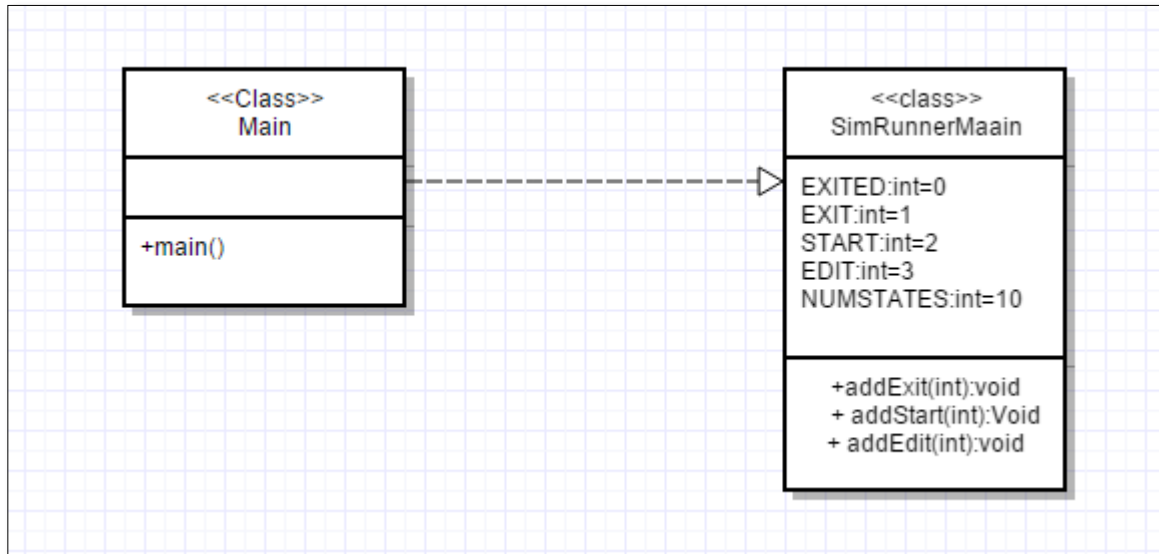
Week	1	2	3	4	Total
Design	0.0	1.0	2.5	11	14.5
Code	2.0	3.5	7.0	28	40.5
Big Bug	0.0	0.0	0.0	4.5	4.5

### Project Plan Summary

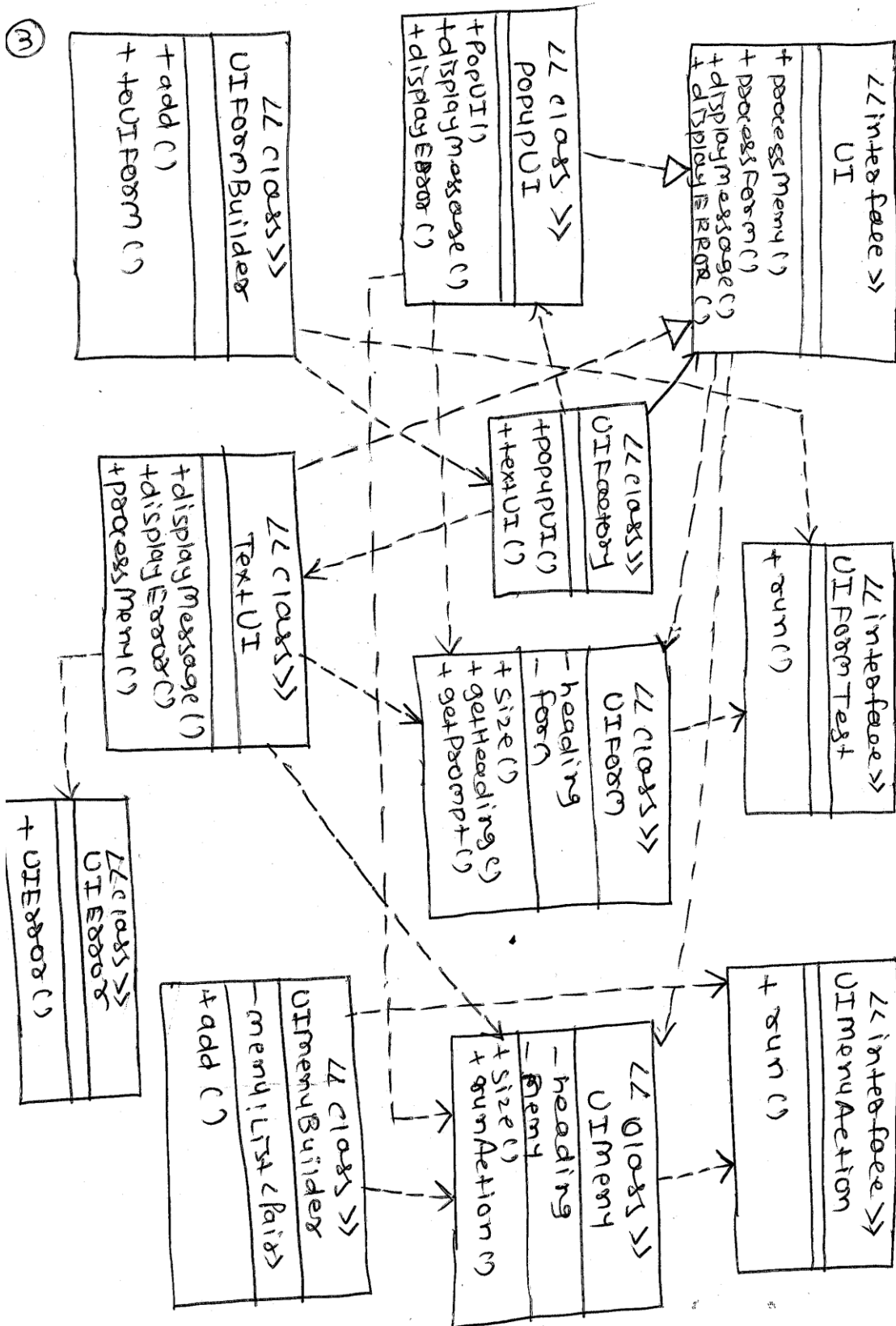


## Chapter -3 System Analysis

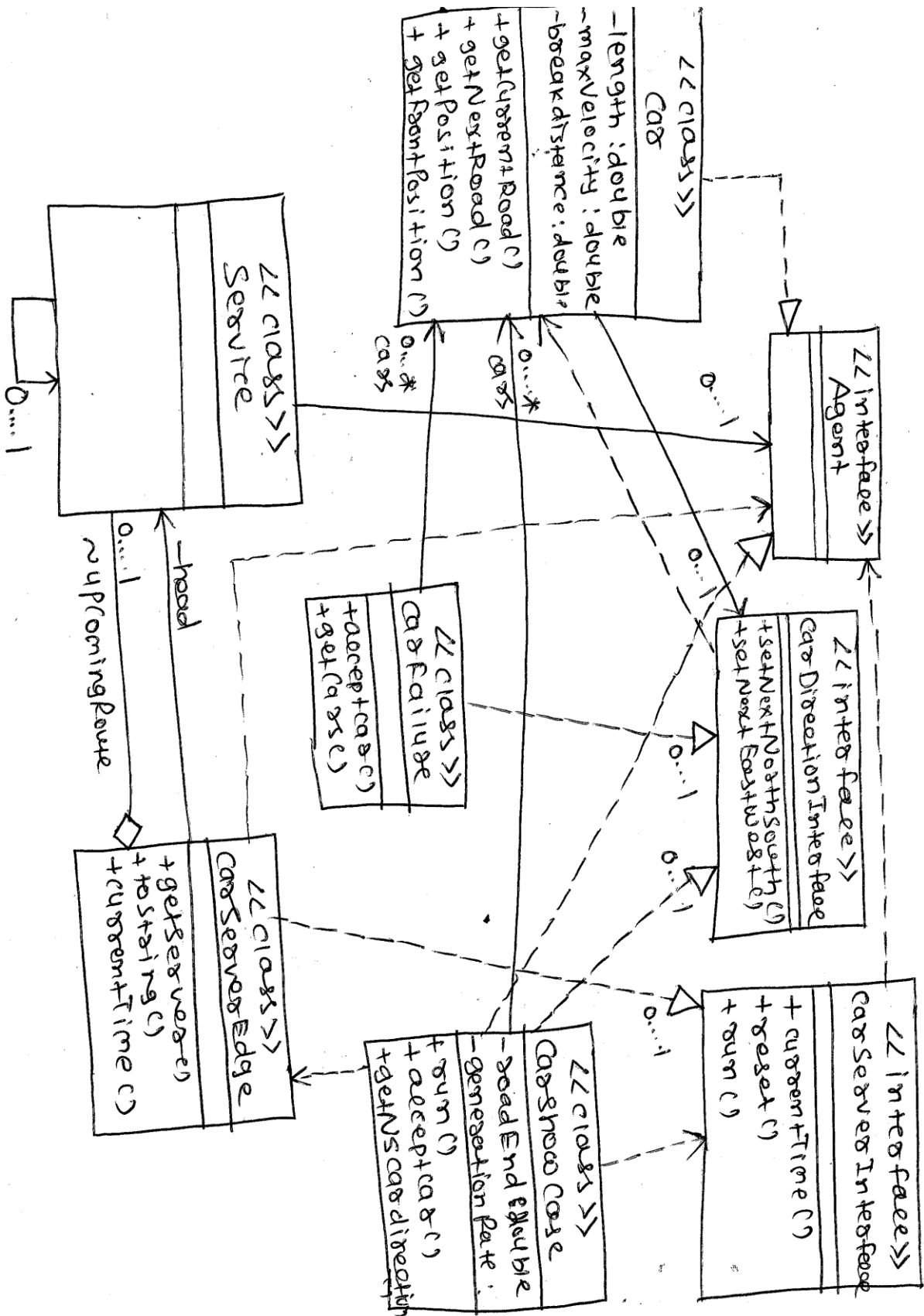
### 3.1 Class Diagram



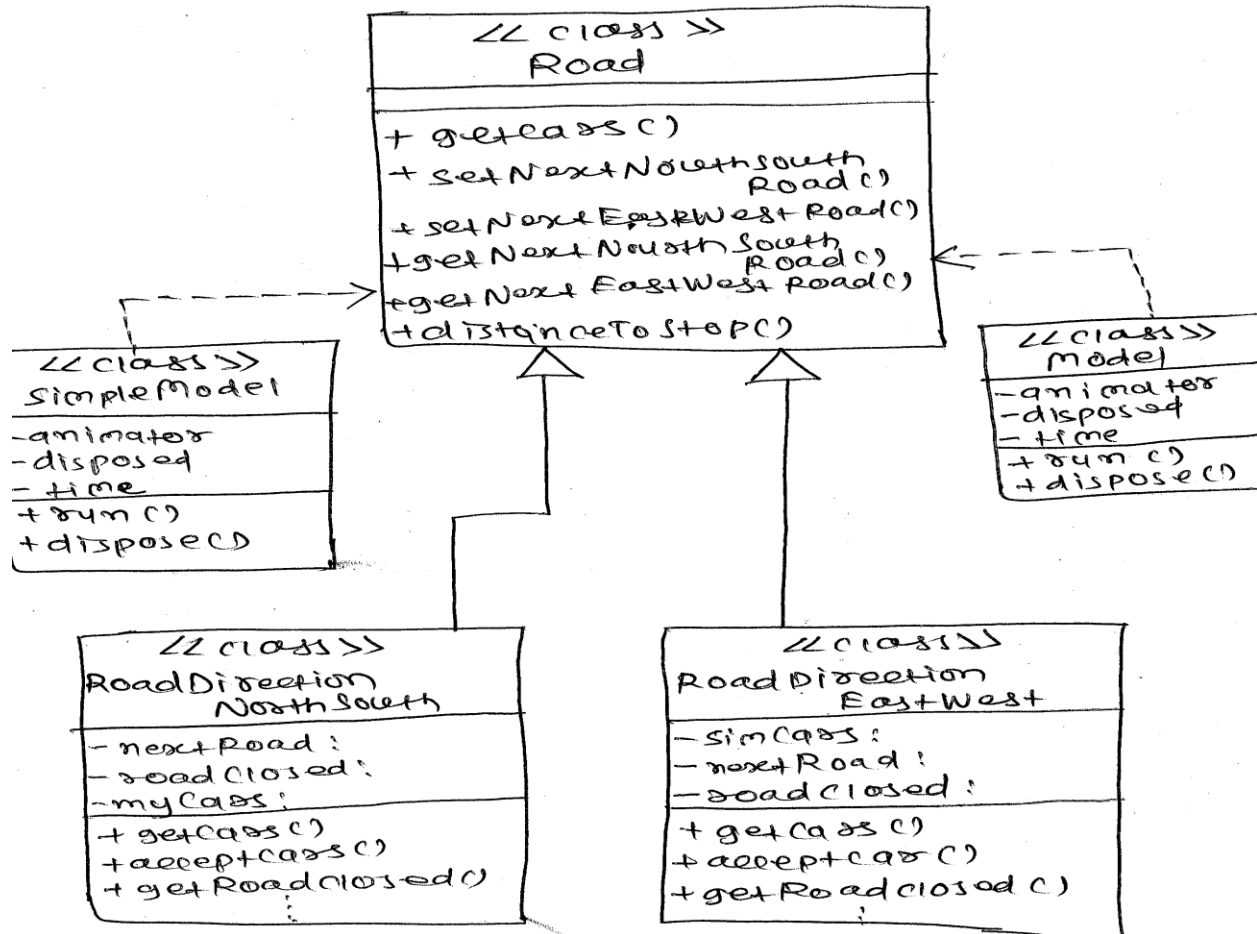
### ⑧ class Diagram For "UI" package :-



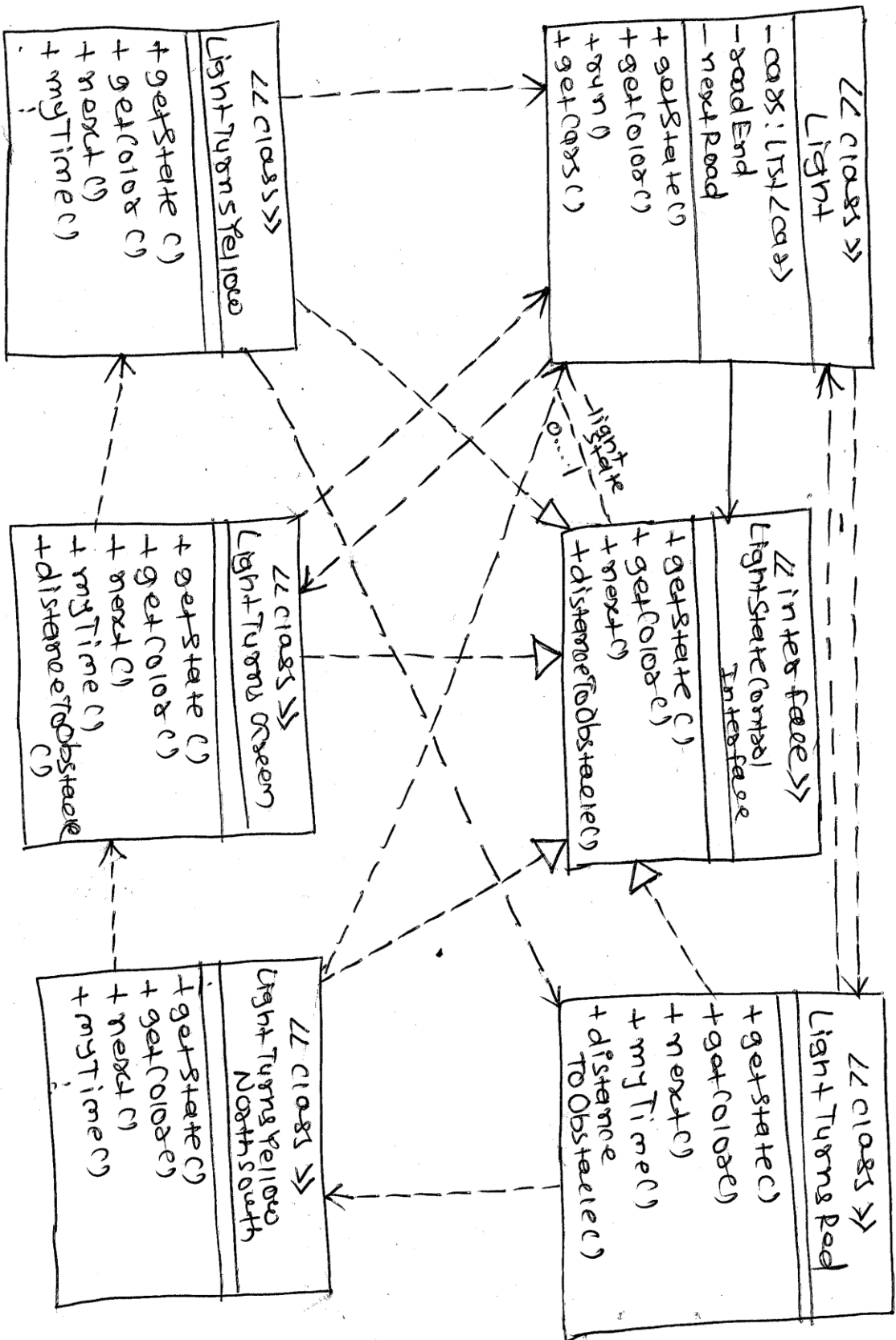
(4) class diagram for car functionalities :-



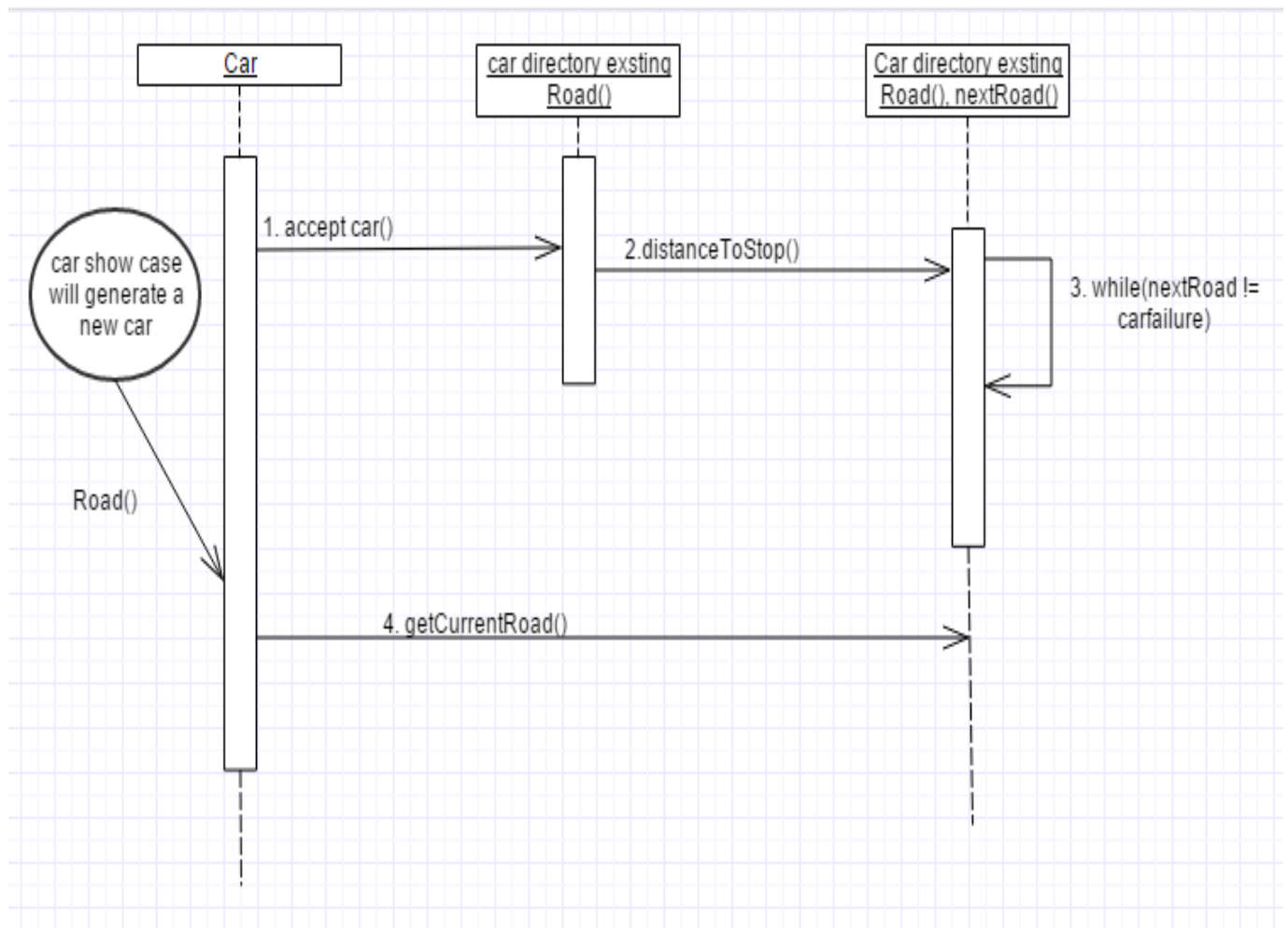
⑤ class diagram for 'Road functionalities':



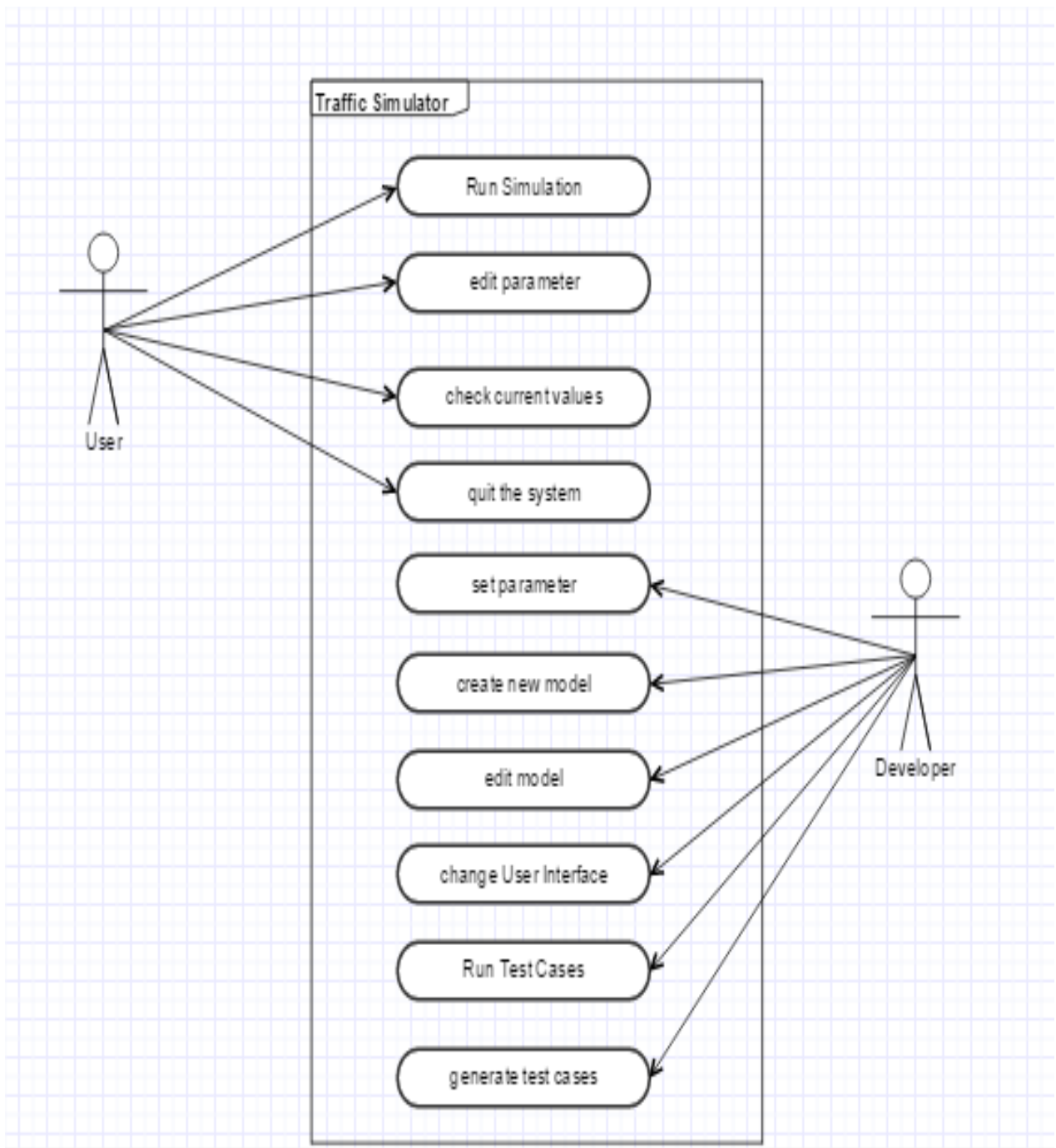
(6) classes designed for Light functionalities :-



### 3.2 Sequence Diagram



### 3.3 Use Case Diagram



## Chapter -4 Appendix

### 4.1 Notes on Pattern

Firstly, I implemented “Template” pattern to my project or I can say that I used Template pattern on some portion of my project. For instance, I used that to implement roads and intersections with signals. To use Template pattern, I just simply made an interface which declares all the directions of roads and with that I made a class as an abstract that implements that interface with all the methods declared in it.

Secondly, I used “State” pattern because implementing all the traffic lights with their states and attributes are easier using state pattern.

Thirdly, I used “Static Factory” pattern to create new objects like to add a car or bunch of cars on a traffic loop.

Along with all these, I implemented “Command” pattern for especially Agent class and MP class.

### 4.2 Success & Failure

According to me, I think if there is another few days like four to five days to work on this project, I can try to add more functionalities with better implementations. Still there are some functionalities and features are left to be developed by me, such as implementation of traffic lights in all four directions, two way road segment instead of just a one way traffic, to create obstacles or hurdles using various sign boards and many more. For example, I will create a sign board of “STOP” signal and all the cars will stop at some certain distance after watching that sign at any intersection. I was planning to add Rail Road-RR intersection but somehow I was not able to implement it. I think I should start much earlier to implement the various features of traffic simulator rather than working on design and user interface of simulator. Furthermore, I took much time to understand the given code with their connection to each other and thinking about how can I used patterns in this section or which pattern would be used to make this task easy and all that. I think realizing that I should watch all the video clips and google-group question forum of our class discussions and implementing it would be easy for me to start and understand this project.



Moreover, implementation of traffic lights and their features with time limitations was a quite difficult task for me in this work. Another problem which took my little time was to reset all the parameters while user press *Choice 15*:- “Reset simulation and return to the main menu”. I was trying to reset all the parameters manually in to the “*SimRunnerMain.java*” class but somehow it didn’t work. After a long discussion with brain I came up with excellent solution and I did declare a method named “*reset ()*” in class “*MP.java*”. The “*reset ()*” method will call after pressing “*Reset Parameters*” i.e. *choice 15*, will reset all the changed parameters to default parameters. But after all things done, while I was running that reset simulation it followed the reset method but the size of simulation was changed to smaller automatically. So I left it to implement finally. As I mentioned earlier many features are remain unimplemented but this project mainly emphasis on design patterns and their knowledge that’s why I used much time behind them.

The project work on “Traffic Simulator” help me to understand different design patterns in Java with their roles along with their practical implementations as well. By implementing this scenario, I found I am ready to work with design patterns on any application. The best way to learn and understand these patterns that, you should design your application or code using these patterns.

Furthermore, this project work helped me to improve my Model-View-Controller (MVC) skill in a very well furnished manner. It helped to manage and handle a large amount of code with understanding of “Enumerations”, “Interfaces”, “Methods” and all that. After completing this, I can see my OO paradigm concepts and fundamentals at some higher level.