# Qatar University

**College of Engineering**

**Department of Computer Science and Engineering**


# Senior Project Report

***Smart City Services Interconnection***


**Project Group Members:**

| | |
|---|---|
| Lameaa Rashid Al-Sulaiti | 201209168 |
| Maryam Nasser Al-Naama | 201209282 |
| Ekram Gamal Al-Qilani | 201105760 |

**Supervisor**: Dr. Abdelaziz Bouras


**2016 - 2017**

This project report is submitted to the Department of Computer Science and Engineering of Qatar University in partial fulfillment of the requirements of the Senior Project course.

# Declaration

This report has not been submitted for any other degree at this or any other University. It is solely the work of us except where cited in the text or the Acknowledgements page. It describes work carried out by us for the capstone design project. We are aware of the university's policy on plagiarism and the associated penalties and we declare that this report is the product of our own work.

Student: Maryam Nasser Al-Naama Date:

Signature:

Student: Lameaa Rashid Al-Sulaiti Date:

Signature:

Student: Ekram Gamal Al-Qilani Date:

Signature:

# Abstract

Our system is a simulation of traffic management through Smart Parking and Smart Traffic nodes. Smart City services and applications are well known in the present days to address problems of high levels of urbanization by providing smart technology solutions. The project is dedicated to simulate the link between smart city services using Internet of Things (IoT) platform. Our system is made up of four different sub-systems each showing or simulating a different aspect or service of smart parking. It is proposed in our system to model the registration process and providing the parking nearest to the seat ticket in a sport or other events (sub-system 1), ease entrance flow to the parking area to the right parking space assigned for a specific vehicle and providing the nearest parking in case of none pre-registration of parking (sub-system 2), simulating the communication between different parking nodes showing requests and responses messages through IoT (sub-system 3), finally the exit flow by connecting the Smart Parking to a Smart Traffic and changing Smart Traffic lights according to number of vehicles to ease the flow(sub-system 4). The first step ~~is~~ ~~assuming~~ assumes a previous ticket registration (of any event) has occurred, ~~and through~~ then the system ~~providing~~ provides the nearest parking space available to the seat number and ~~saving~~ saves the information in the system ~~in purpose of~~ to ~~easing~~ ease the flow of cars and people aiming for the event. The second step is authorizing the cars entering the parking area (by assuming reading cars plate numbers through sensors) and check if previously registered and giving access to authorized vehicles and if not, searching for nearest parking among different zones. The third step is related to the communication of the needed information from smart parking to smart traffic, which will react accordingly to make the flow of the vehicles faster by increasing the duration of longing the green light depend on the number of vehicles. ~~The~~ To establish the communication between the smart parking and smart traffic, the Open Group IoT Standards O-MI (Message Interface) and O-DF (Data Format) is used. Based on previous relationship with Lusail, we have visited one of the parking sites there, which consist of four parking zones that are managed by a common smart parking system. We have seen some functionalities and services of the system and have taken some data to help develop a simulation of smart parking.

In Subsystem 1, we propose....

# Acknowledgment

This research was supervised by Dr. Abdelaziz Bouras. We would like to show our gratitude to Dr. Abdelaziz Bouras and Mr. Ahmad from ictQATAR (The Supreme Council of Information & Communication Technology) for sharing with us their pearls of wisdom in programming knowledge and research experiences. Also, we would like to thank them for helping us throughout this journey to overcome obstacles and give our best effort to deliver a successful project.

# Table of Contents

Declaration ...ii

Abstract ... iii

Acknowledgment ...iv

List of Figures ...vii

List of tables... viii

1. Introduction and Motivation ...9

1.1 Introduction ...9

1.2 Problem Statement ...9

1.3 Problem Significance ...10

1.4 Objectives ...11

2. Case Study...13

3. Background and related work ...14
   3.1. Background...14
   3.2. Related work...14

4. Requirements analysis ...18
   4.1. Functional requirements...18
   4.2 Non-functional requirements...22
   4.3 Ethical requirements...22
   4.4 Assumptions...23

4 Solution Design ...24
   4.2 Overview...24
   4.3 Structural model...27
   4.4 Behavioral model...32
   4.5 Design principles and patterns...36
      4.5.1 Design pattern used ...36
      4.5.2 Design principles applied and benefits...36
   4.6 Database design...38
   4.7 User interface design...39
   4.8 Hardware/software to be used...42
   4.9 Alternative solutions and tradeoffs ...43
      4.9.1 Possible Solutions of IoT and Tradeoffs...43
      4.9.2 Possible Solutions of Exit Flow Simulation and Tradeoffs...44
      4.9.3 Possible Solutions of the System Simulation and Tradeoffs...45

5. Implementation ...46

6. Testing ...49

6.1 Functional Requirement Testing…………………………………………………...…….48

6.1.1 Java Testing…………………..……………………………………..………48

6.1.2 Arena Testing……………………………………………………..………51

6.2 Non-Functional Requirement Testing……………………………………………………55

6.2.1 Performance Testing…………………..…………………………………………..55

6.2.2 Availably Testing…………………………………………………………55

6.2.3 Usability Testing…………………………………………………………56

7. Expected Impact ........................................................................................................58

8. Conclusion ...............................................................................................................59

9. Future work...............................................................................................................60

10. Student reflections .....................................................................................................62

References.......................................................................................................................64

Appendix A – Project Plan ..............................................................................................65

A.1. Project milestones…………………………………………………………65

A.2. Project timeline……………………………………………………………66

A.3. Anticipated risks…………………………………………………………...70

# List of Figures

Figure 1. Al-Gharafa Sports Club parking area……………………………….……….10

Figure 2. Mr. Anuj MADAAM presenting in Lusail City…………………………...12

Figure 3. Visualization of the software used to monitor in Lusail City……….……….…13

Figure 4. The main monitoring room in Lusail City……………………….……….…..13

Figure 5. Use case diagram……………………………………………..……………18

Figure 6. Incremental Diagram of Smart Parking System……………………..……….24

Figure 7. High Level Architecture Diagram of Smart Parking System…………...……….25

Figure 8. UML Class Diagram of Smart Parking System………………….…………..27

Figure 9. UML Class Diagram of Parking Registration System…………….…………28

Figure 10. UML Class Diagram of Real-Time Smart Parking Management System……..29

Figure 11. UML Class Diagram of the O-MI/O-DF tree for the Smart Parking Management System……………………………………………………………………………..30

Figure 12. UML Class Diagram of the Exit Flow Simulation……………….………31

Figure 13. Activity Diagram of Parking Registration System………………….……32

Figure 14. Activity Diagram of Real-Time Smart Parking Management System….….…32

Figure 15. Activity Diagram of Smart Parking Management System Between Two Machines. ……………………………………………………………………..……......33

Figure 16. Activity Diagram of Exit Flow Simulation……………………………………33

Figure 17. Sequence Diagram of Parking Registration System……………………….34

Figure 18. Sequence Diagram of Real-Time Smart Parking Management System………35

Figure 19. Database Design Diagram…………………..………………………..…..38

Figure 20. User Interface of Parking Registration System…………………………..…39

Figure 21. Match View of Smart Parking Management System. (User Interface I)…………40

Figure 22. Parking View of Smart Parking Management System. (User Interface II)……...40

Figure 23. User Interface of Smart Parking Management System between two machines …41

Figure 24. User Interface of Exit Flow Simulation………………………………………42

Figure 25. Probability values of Car Arrivals……………………………………………….52
Figure 26. Numbers of cars exiting per minute in 30 minutes in a static excel file……...….53
Figure 27. Accumulated time per minute in 30 minutes in a static excel file…………..…...53
Figure 28. Numbers of cars exiting per minute in 30 minutes in a static excel file……..…55
Figure 29. Accumulated time per minute in 16 minutes in a dynamic excel file…………..55
Figure 30. Responses time of the system on different trails………………………………..56
Figure 31. ParkingRecord and ParkingSite UML class diagram using DATEX II…………61

## List of tables

Table 1. Differentiation between some related works……………….……………………….17
Table 2. Use Cases summary…………………………………..………………………….19
Table 3. Non-functional Requirements…………………………………………………….22
Table 4. Ethical Requirements……………………………………………………………….22
Table 5. Comparison of possible solutions for IoT and their tradeoffs…………………….43
Table 6. Comparison of possible solutions for Exit Flow Simulation and their tradeoffs…...44
Table 7. Comparison of possible solutions for System Simulation and their tradeoffs……..45
Table 8. Duration Time for Batch car sets……………………………………….………….48
Table 9. Function Testing Summary…………………………………….…………..49
Table 10. Numbers of cars exiting per minute in 30 minutes in a static excel file…………..53
Table 11. Numbers of cars exiting per minute in 16 minutes in a dynamic excel file……….54
Table 12. Usability Testing for Non-Functional Requirements…………………………...…57
Table 13. Anticipated Risks……………………………………………………….…………70

# 1. Introduction and Motivation

## 1.1 Introduction

Smart city is a ~~latest~~ recent urban development vision that is highly needed in the region, which focuses on integrating, analyzing, managing and collecting information and establishing communication between different aspects of a city including learning institutes, transportation systems, hospitals and health, waste management, law enforcement institutes and other city services by using smart technologies such as information and communication technology (ICT) and Internet of Things (IoT) and establishing communication between these aspects and preforming actions based on them. Smart city services main purpose is analyzing and managing communication between smart city aspects in a secure fashion way to accomplish developed city growth. In our system we focus on simulating a small yet highly essential part of smart city, Smart Parking and Smart Traffic. We reflected some main functionality that can be applied in real life parking management controlling systems. As it is highly difficult to deal with sensors and real vehicles to develop a real smart parking system, we have considered simulati~~ng~~on as a very important step in showing the management of parking using advanced technologies in this project. In our project, we have narrowed the problem by focusing on the occur~~ring~~ence of a sport event and showing through our system different scenarios of managing the traffic into, in and out of the parking areas.

perhaps you can rephrase the sentence.

## 1.2 Problem Statement

In most of our modern cities, controlling and easing the flow of cars in traffic jams in highly crowded places is becoming a must. We all face the problem of traffic every day in our lives; it affects our daily life badly by losing time in such a non-productive process. By taking into consideration the increasing population in the upcoming years, it can be predicted that traffic would be even worse. We mean in this project to contribute to finding solutions for traffic by focusing on organizing parking areas and their control and easing the flow of exiting and entering of cars from and to parking areas using advanced technologies. We will be taking in consideration crowded places in this project and how to manage the entering and exiting in an organized manner, avoiding traffic jam that the drivers usually experience~~d~~ in such largely overcrowded spots. We also aim to solve the problem of traffic in the situation when such places should be evacuated in the shortest amount of time possible; such as at an emergency event. in specific, we will be studying the case of one of the stadia in Qatar to solve this problem. As it is known, stadia represent one of the most highly crowded places in a sport event, and entering and exiting a stadium's parking areas ~~are~~ is difficult and sometimes expose the driver's safety to danger. ~~Nevertheless,~~ Additionally, searching for parking space can be come even harder and a waste of time. We propose to connect a Smart Parking and Smart Traffic to help with both situations of exiting from parking areas to the streets in the shortest time possible and entering parking areas to the right parking spot in the easiest way possible. We will be using Internet of Things (IoT) concept to develop the solution by developing the way of communication between Smart Parking nodes themselves and between Smart Parking node and Smart Traffic node to accomplish an easy flow and to perform an action existing plan in a ~~specific~~ short amount of time. We also mean to organize the parking areas by optimizing fasting the process of entering of the cars into the parking areas by early registration of the car plates with the corresponding parking zone nearest to the spectator seat. This allow~~ing~~s access to registered cars only and decreas~~ing~~es the exposure to traffic jams in parking areas and time los~~ing~~t in searching for parking spots.

Through our visit to Lussail city and speaking with Mr. Anuj MADAAN, facility manager in Lusail Marina Car Parks, we discovered that they plan ~~mean~~ to handle such problems by developing Smart Parking system. Moreover, ~~We understand that~~ the problem~~atic~~ that their smart parking system is trying to solve is managing parking areas functionalities that will help organize the flow of traffic and reduce congestion inside parking areas using new technologies to enhance the city of Lussail into a smart city. They are trying to solve the problem of wasting time by searching for parking spaces by developing system that handles reservation of parking spaces and using Internet of Things (IoT) for communication between different parking nodes and Smart Traffic system of the city.

redundant

## 1.3 Problem Significance

The importance of the project can be seen through the lack of efficient traffic control strategies that is obvious in the world today in highly crowded spots and how does it impact negatively the drivers comfort. Qatar has these typical problems, in a recent football match between Barcelona vs. Al Ahli Saudi that took place on Dec,13 2016 in Al-Gharafa Sports Club, it has been declared in a local newspaper about the organization of the parking areas to solve such problem.


**Figure 1. Al-Gharafa Sports Club parking area.** [1]

Such traffic control solution is essential and highly needed in our country to insure comfort and more importantly, driver's safety. The impact of not having such a solution for traffic can cause is bad in the way that some traffic that can last for a long time and it affects the driver badly as it is a non-productive process of waiting or searching for parking. Even worse more, it affects the driver's safety in some situations where there is an emergency evacuation in a crowded parking area. It also exposes drivers to more accidents due to the tight spacing and the stopping-and-going process. Even more, it is bad for the economy of the country itself. What got us interested in contributing to such problem is what we face daily in Qatar streets; ~~of~~ many types of traffic jams and the highly increasing rates of accidents ~~accordingly~~. We felt, as all other people who live in Qatar do, that it is important and highly needed to provide a solution to decrease the traffic and ease the way of moving cars. The project can highly benefit areas like stadia in sport events that are taking place in the upcoming Qatar 2022 world cup. Such traffic issues control is ~~highly~~ need to be addressed needed and is an important component of ongoing economic development and must be a priority in these day events to tackle congestion and bottlenecks. This would benefit the stadia by providing the visitors a comfortable and safe drive through their parking areas and save them the long wasting time process of searching for a parking spot.

This project would affect us a lot in our future career ~~by having a~~ participation ing in an experience of solving a real-life problem that we face every day and manage to accomplish what we think is needed to develop a solution to the addressed problem, that satisfies and meets our expectation.

## 1.4 Objectives

The main purpose of the project is to develop a solution for better handling functionalities of smart parking management system. The system will handle allocating parking spaces according to reservations, and will simulate the communication between Smart Parking and Smart Traffic nodes. The aim is to ease and optimize ~~fasten~~ the flow of vehicles. This will contribute to ease the flow of traffic in the vicinity of the parking areas.

The main objectives of this project are: means to tie something

- Reduce the time of searching for the optimal parking spot.
- Better monitoring and managing parking spaces.
- Control the flow of traffic inside parking areas to insure a safe atmosphere for drivers and avoid accidents
- Provide intelligent management of the Traffic signals according to the flow of cars each direction.

## 2. Case Study

A visit to Lusail City ~~were~~ was planned to explore more about the implementation of Smart City concept. Lusail will provide a high-technology environment for its residents and visitors involving wireless communication networks to offer advanced services. All their smart services management will be centralized in Lusail Command and Control Centre (LCCC). However, one of their smart implemented technologies is their smart parking system.

We met with Mr. Ibrahim Kocagoz, project manager in Smart City, to have an overview of Lusail City development projects and get a general information about how Lusail City is implementing a smart parking management system. He claimed that they have four marinas, each having four zones where each zone has 4 basement floors. To reduce complexity, each zone has a monitoring room with a manager and all parking zones are monitored in a control room in LCCC.

We ~~as well~~ also met with Mr. Anuj MADAAN, facility manager in Lusail Marina Car Parks to have a closer narrower view into the parking monitoring procedure at one zone. He showed us the software they use to monitor real-time occupancy of parking spaces in the control room. In one floor, there are several sensors to check for parking space occupancy and they use Park Assist sensor cameras. One sensor is responsible for four different parking spaces. Also, they are using IoT concept in monitoring the temperature, air handling and fire fighting in the parking zone to avoid any technical issues.

We captured some special moments during our visit:

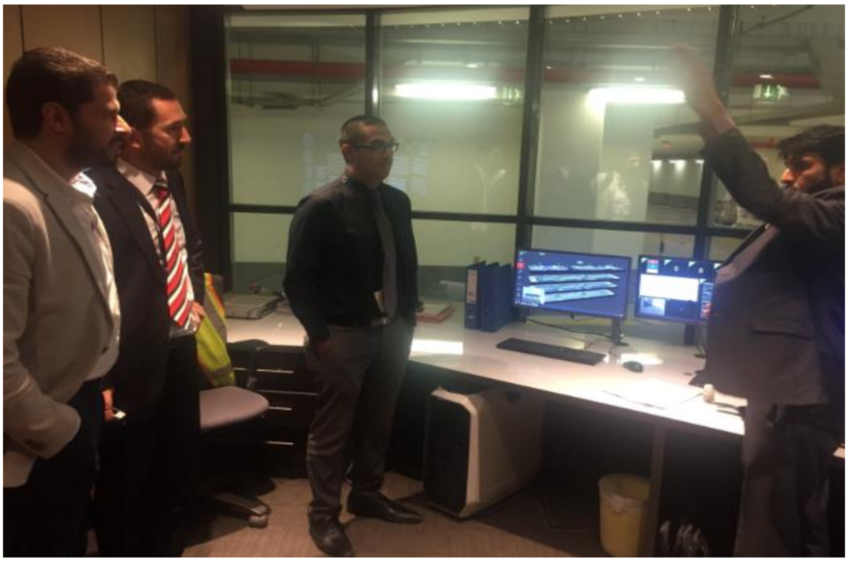


**Figure 2. Mr. Anuj MADAAM presenting to us, Mr. Ibrahim Kocagoz and Mr. Ahmed Hefnawy inside the control room in Zone 3 in Marina 1 in Lusail City.**


**Figure 3. Visualization of the software used to monitor all four basement floors inside the control room in Zone 3 in Lusail City.**

**Figure 4. The main monitoring room which displays all four different zones in Marina 1 in Lusail City.**

The visit to Lusail City was very useful. It helped us recognizing the differences and similarities between Lusail City smart parking system and our proposed solution. Mr. Ibrahim and Mr. Anuj gave us an overview of how IoT and Smart City concepts are implemented physically in real world and most importantly, in Qatar. They supported us to start implementing our smart parking system and develop a new smart parking system that have not been implemented in the country with efficient services and new ideas. Our case study will deal with [1] Open Standards, we will be using the Open Group standards (O-MI/O-DF) for the Internet of Things to enable everything to be connected on the fly to deliver our project faster with less effort and most importantly lower cost. [2] Link with traffic services, our proposed system will have a connection with the traffic lights directly around the stadium to ensure a safer atmosphere for the citizens.

# 3. Background and related work

## 3.1. Background

### 3.1.1 Internet of Things

The background of our work is related to the use of Internet of Things (IoT) frameworks in a context of a Smart City. IoT means the use of intelligently connected systems and devices to control data gathered through sensors in various machines. IoT is believed to spread rapidly in the nearest future and will release a new dimension of services which improve consumers' way of living as well as the productivity of enterprises [2]. A Smart City is a global development vision which combines Internet of Things (IoT) with other Information and Communication Technologies to provide a solution to a city problems and its vision for the future evolutions.

### 3.1.2 Simulation

"Simulation is the imitation of the operation of a real-world process or system over time". [3] The first requirement to demonstrate a simulation is to develop a model. The model will represent the main functions and attributes of the system. The simulation denotes the operation of the system with time while the model embodies the system itself. The use of simulation is to connect Internet of Things with external system. A model? system can simulate a real-time system and get a practical feedback applied to different real places such as stadia, malls and theaters and get a practical feedback. Moreover, simulating systems allow the developer to observe the efficiency of a design before implementing it. Simulation can be a good solution for various real-time problems such traffic congestion in cities and propagation delay in safety and electrical engineering.

## 3.2. Related work

In 2008, IBM introduced the concept of "Smart City" to observe how a city functions in order ~~can be progressed~~ to improve the quality of life. Today, the concept of Smart City has become a global trend in the development of advanced cities in the 21st century [4]. IoT solutions improve a city in many aspects by enhancing public transportation, managing traffic congestion, developing infrastructure and other community services which all cities require. Smart City technology has been developed by many sectors in different countries including transport and traffic management, health, energy, water, and government. We summarize hereunder the main sectors related to our work and some current related works.

### 3.2.1. Smart Energy

As the cost of energy increases rapidly, organizations are developing new ways of saving energy. An efficient energy management system helps optimize energy consumption for heating, ventilation, air conditioning, refrigeration, lighting, fire systems, and security systems, ensuring that energy is used only when needed doors [5]. Smart energy management systems use IoT to provide ideal solutions for energy consumption in buildings and campuses. Each device is connected to the internet to gather variety of information including energy consumption, power factor and configuration. For instance, artificial lighting in the rooms is being positioned in a way to ensure that the total illumination in the room is ergonomically suitable for all occupants. When the sunlight in a room rises, artificial illumination is immediately reduced keeping the total illumination constant at a specific ergonomic level. Moreover, fire sensors can become active in the event of fire where fire exits detect the number of people exiting across different routes. The Energy Management System can utilize this information to manage crowdedness through exit.

In [6], Smart Meters are introduced and developed to overcome the issues of traditional electricity meters. The researchers claim that the main objective of developing Smart Meter System is to decrease the energy consumption in the households. Based on hourly consumption of energy, an energy alert will be fired to the consumers using Smart Meter data. Furthermore, smart meters are connected to an online billing which might reduce the timely consumption of consumers. A study was applied on hourly measurement data of more than 15 households to detect energy consumption pattern. Smart Meters will benefit the electricity market as it reduces carbon dioxide emissions along with preserving the energy.

### 3.2.2. Smart Infrastructure

With the growth of worldwide population, water level is decreasing rapidly. The aim of some research works of Smart Water Management is to manage the functionality of water infrastructure more efficiently to fulfill the citizen’s demand to high quality water. Some researchers are performing experiments using advanced signal processing algorithms to detect any pre-failure signals of water pipe burst with vibration sensor [7]. Other experiments are related to gathering data about vibration of pipes and water pressure using specific sensor units. Using the data collected with sensors, simulations are developed to simulate the complex hydraulic conditions in real world utilizing mathematical models or to allow higher level control of water networks.

In [8], a research is developing a Smart Waste Management due to the fast growth in population which increases waste disposal. Moreover, the proposed system is implemented to avoid spread of some harmful diseases which threat human health. The system consists of Smart dustbins which are connected to the Internet to get the real-time information about each bin and hence, a decision is made. The dustbins are interfaced with micro controller system with radio frequency modules and infrared sensors. Infrared sensors will detect the dust level in a dustbin and sends signals to micro controller. The same signal is encoded and sent through radio frequency transmitter and an Internet connection is enabled as well. The information gathered is received and processed in the cloud which shows the status of the waste in the dustbin on the user interface on a web browser.

### 3.2.3. Smart Health

Using IoT technology, machine to machine communication(M2M) is developed for healthcare data to solve serious health related problems. Different IoT healthcare applications were introduced, including glucose level sensing, electrocardiogram monitoring, blood pressure monitoring, body temperature monitoring and oxygen monitoring. Each IoT-based device monitors specific type of health-related information where all are classified as sensor data collecting layer. In [9] a network architecture of such system consists of several network layers such as the data collection layer; the smart medical services layer and the medical resource management layer. The sensor data collection layer involves data sensing, data storage devices, local computing and processing units and wired/wireless transmitting modules. This is connected to different wired/wireless protocols, such as Wi-Fi, Bluetooth, Ethernet, RFID and 3G/4G network. Data collected is available to be accessed by the medical resource management layer which involves the management of medical resources through cloud computing to provide patient privacy protection and security. The smart medical service layer is straight connected to medical facilities such as hospitals, medicine supply chain and emergency centers which can receive~~s~~ all patients' real-time information. Due to the interconnection of many devices, M2M communication is used.

Moreover, the work of [10] is dealing with a new concept for health monitoring within home environment. Minimizing health risks for elder people by monitoring blood pressure, body temperature and mainly the falls for the aging people. Around 80% of the old people who are older than 60 suffers from sudden downfalls leading to some serious health risks and injuries.

The objective of this research is to implement an intelligent floor which detect the fall events, walking patterns and other abnormal behavior to detect an emergency case. The advantage of the proposed system is that older people don't have to activate emergency calls themselves which on many cases ~~in~~ are impossible in the case of fainting. In case of an emergency situation, the system may contact a professional medical doctor or a relative of the user. The field of smart health is growing and applications are being developed everywhere.

### 3.2.4. Smart Mobility

This topic is widely investigated nowadays. As an example, [11] introduces a smart parking system by connecting the hardware and software parts together. The proposed system is providing a graphical view of available parking slots to the user through an Android application. The system as well is consisting of Infrared sensor which detect when a car is parked at any space and hence, the graphical view of the application is updated according to that change. The advantage of using Infrared sensor is that it is continuously sensing the parking slots for any status change at any time. The researchers claim that the contribution of such system is to reduce time, provide intelligent management, increase space utilization and save fuel from the driver's point of view.

good!

Other works have developed simulators to import real-world maps and traffic traces [12]. The real-world map is to show virtual parking lots as one block as well as the path to different parking lots, making it easier for drivers to locate the exact parking location. Furthermore, a real-world traffic traces is used to generate the number of drivers who are approaching a specific parking lot. Based on the simulation developed, conclusions could be drawn to suggest reservation-based smart parking systems to decreases traffic congestion caused by parking searching, besides reducing the amount of traffic volume searching for a parking.

Another system is presented by [13] where information about available parking spaces is stored in a database about a specific geographic area and process to place vehicles at available positions. One single gate covers an area of 1 $km^2$. The gateway gathers information about a vehicle and sends information to the database in a real-time. The occupancy of parking spaces is reported to the drivers through a mobile application. When a car is parked, the sensor will detect the status of the parking and update it in the database. The researcher claims that the proposed system' objectives is to sense occupancy of a vehicle and parking spaces in real-time, guides drivers to available parking, manage traffic to flow more easily through IoT technology.

Each introduced system has a main goal to target, comparing other Smart Parking Systems to what we are simulating is classified as follows:

**Table 1. Differentiation between some related works.**

| | **Osmani A., Gawade A., Nikam M., & Wavare S., 2016** | ***Wang H. & He W., 2011*** | **Basu A., 2014** | ***Proposed Smart City Services Interconnection*** |
|---|---|---|---|---|
| Use of Internet of Things | √ | | √ | √ |
| Reduce Parking Searching Time | √ | √ | √ | √ |
| Make a relation with Traffic Congestion | | √ | √ | √ |
| Use of Simulation | √ | | | √ |
| Provide parking payments | | √ | √ | |

good!

Our system is considering two main targets (1) reducing parking searching time as it scans the plate numbers and allocate specific parking for a spectator, and (2) managing the link with the traffic congestion by communicating with traffic signals to provide a smooth flow. Our work uses both IoT platform and Simulation solutions.

However, the proposed system will not handle parking payments as we assume that parking fees will be handled by an external ticket registration and payment services.

## 4. Requirements analysis

Through our visit to Lussail City to better understand functional and non-functional requirements of real smart parking systems. We have conducted some functional requirements such as allocating parking space and managing exit flow, along with other additional requirements that our system will handle that can be useful and can achieve the goals of this project.

### 4.1. Functional requirements

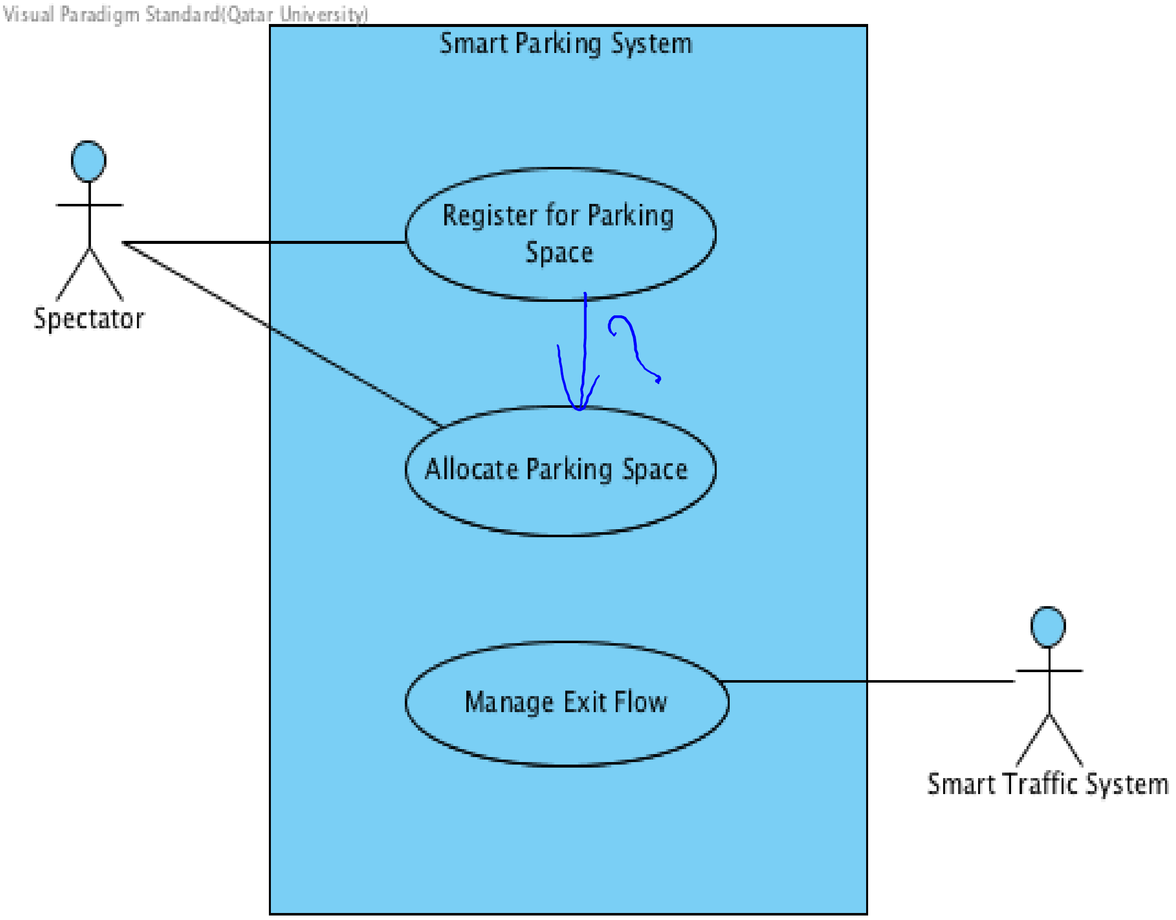



**Figure 5. Use case diagram**

### Table 2. Use Cases summary

| Use case | Brief description |
|---|---|
| Register for A Parking Space | A spectator will be registering for a parking space by providing the system with personal information such as, name, Id, car plate number, match and ticket number. |
| Allocate parking space | After a car is registered to the system by it's plate number, if the user enters the car plate number the system will allocate the driver the most suitable park for it. |
| Manages Exit Flow | In case of having a huge number of vehicles exiting the stadium, the Smart Parking system calculate the number of vehicles exiting each gate heading different directions and hence, sends results to Smart Traffic system which manages the traffic lights signals based on given results. |

- Registration for a parking space:

<table>
<tr><td>Use case Id:<br>UC01</td><td colspan="2">Registering for parking space</td></tr>
<tr><td>Brief Description</td><td colspan="2">A spectator provides personal information along with ticket number, car plate number, parking type and match to register for a parking space.</td></tr>
<tr><td>Primary actors</td><td colspan="2">Spectator</td></tr>
<tr><td colspan="3">Preconditions:<ul><li>Ticket number must be registered in the system.</li></ul></td></tr>
<tr><td colspan="3">Post-conditions:<ul><li>A park space is assigned for the registered spectator.</li></ul></td></tr>
<tr><td colspan="3">Main Success Scenario:</td></tr>
<tr><td colspan="2">Actor Action</td><td>System Response</td></tr>
<tr><td colspan="2">1. Enter name, ID, car plate number, match, parking type and ticket number.</td><td></td></tr>
</table>

| | |
|---|---|
| | 2. Finds the nearest zone and parking space for the spectator based on parking type specified. |
| | 3. Register a parking space for spectator |
| | 5. Show the registered parking space Id to the spectator |
| **Alternative flows:**<br><br>1.a. if the ticket number is not found in the system, an error message will appear to the user. | |

- Allocate parking space:

| **Use case Id:** UC**02** | Allocate parking space |
|---|---|
| **Brief Description** | After a car is registered to the system by its plate number, if the user enters the car plate number the system will allocate the driver the most suitable park for it. |
| **Primary actors** | Spectator |
| **Preconditions:**<br><br>• Car plate number must be registered in the system. | |
| **Post-conditions:**<br><br>• A park will be allocated for the driver. | |
| **Main Success Scenario:** | |
| **Actor Action** | **System Response** |
| 1. Enter car plate number. | 2. Check eligibility to enter. |
| | 3. Updates the car plate number and give access to car. |
| | 4. Allocate the parking for the car nearest to its seat. |
| **Alternative flows:**<br><br>2.a. If the car is not registered, it does not allow access of the car. | |

- Manage Exit Flow

| **Use Case Id:** UC**03** | Manage Exit Flow |
|---|---|
| **Brief Description** | Manages the exit flow of vehicles with Smart Traffic System to avoid traffic jam. |
| **Primary Actors** | Smart Traffic System |
| **Preconditions:**<br>• Huge number of vehicles exiting the stadium. | |
| **Post-conditions:**<br>• Vehicles exited stadium in shortest time possible. | |
| **Main Success Scenario:** | |

| **Actor Action** | **System Response** |
|---|---|
| 1. Spectators exiting the gate in different directions. | |
| | 2. Detect number of vehicles exiting each gate per minute. |
| | 3. Sends information gathered to Traffic System. |
| 4. Smart Traffic System reads traffic information from Smart Parking System. | |
| 5. Smart Traffic System manages traffic lights duration based on information given. | |
| **Alternative Flows: -** | |

## 3.2 Non-functional requirements

**Table 3. Non-functional Requirements.**

| | |
|---|---|
| **Performance** | The system should provide fast response for authorization to the users with a time interval of 10-15 seconds' maximum for 95% requests. |
| **Availability** | The system should be available always under circumstances of high load of cars. |
| **Usability** | The system is clear, concise and easy to use and understand. |

how is this number selected?

## 3.3 Ethical requirements

what about privacy of users' information (name, ID ...)?

| ACM and the IEEE Software Engineering Code of Ethics | Description |
|---|---|
| **PUBLIC** | |
| 1.04. Disclose to appropriate persons or authorities any actual or potential danger to the user, the public, or the environment, that they reasonably believe to be associated with software or related documents. | Our prototype will be presented to the user (Lussail City Smart Parking management system) for validation and reduction of any possible harm (security) to future visitors. |
| **CLIENTS AND EMPLOYER** | |
| 2.02. Not knowingly use software that is obtained or retained either illegally or unethically | A combination of software the system will use such as The Open Group IoT platforms, Eclipse for Java and Arena are guaranteed be ethically payed and downloaded from their official websites. |
| 2.05. Keep private any confidential information gained in their professional work, where such confidentiality is consistent with the public interest and consistent with the law. | Any sort of information gained from meetings and discussions with the stakeholders about will be kept private and secured. |

| PRODUCT | |
|---|---|
| 3.02. Ensure proper and achievable goals and objectives for any project on which they work or propose. | We adopt modular approach based on step by step testing. The use of simulation helps enhancing the system at each step. |
| 3.05. Ensure an appropriate method is used for any project on which they work or propose to work. | Using The Open Group IoT platform is efficient as it enhances a user-friendly interface which require fast training.<br>The Usage of simulation offers users a practical feedback to control the efficiency of the current design. |
| 3.08. Ensure that specifications for software on which they work have been well documented, satisfy the users' requirements and have the appropriate approvals | The system is developed after a meeting with stakeholders (Lussail Smart City head of Project, Mr. Ibrahim and Zone 2 Smart Parking Team ( Mr. Anuj MADAAN, facility manager in Lussail Marina Car Parks and 2 teammates) to gather all their requirements and how they want the system to function. Then, a use case specification is designed to fit their needs. |
| **MANAGEMENT** | |
| 8.02. Improve their ability to create safe, reliable, and useful quality software at reasonable cost and within a reasonable time. | The use of free standard of the open group helps the creation of coherent and interoperable system. A project plan is designed to ensure that the system will be done within a reasonable time duration. |

## 3.4 Assumptions

To guarantee a successful system that achieves all the previously proposed objectives, we assume that:

1. There is an online registration website for reserving the ticket of a sport event.
2. We will have information of tickets or seat numbers and nearest zones to them.
3. We will get the car plate number of the entered car from the scanner or sensors.
4. There is parking manager/administrator to monitor the parking system.

# 4 Solution Design

## 4.2 Overview

The proposed final system will be divided into four different subsystems; Parking registration system, Real-time parking management system, A visualization of a real-time parking management system between two machines and a simulation of an exit flow event.

The first subsystem is a registration for a specified parking space given the ticket's number and position in the stadium. The second subsystem will be a real-time parking management system for arriving spectators with an overall parking analysis such as the occupancy rate at real-time.

The third subsystem will visualize the communication between two machines for a parking management system. The first machine/PC, the parking emulator which will show the assignment of parking spaces for the registered spectators and the other is the parking monitor which will receive the assignment of parking space using a network connectivity at real-time demonstrating the concept of Internet of Things. Finally, a simulation of an exit flow event. Simulating several vehicles leaving the stadium given an overall statistic of the exact number of spectators in the stadium and the connection with the traffic lights.

Combining all the four subsystems will result in our final proposed system which handles the Smart Parking system from beginning to end. Starting from registering for a parking space and allocating parking spaces on a specific event day to handling the exit flow of vehicles at the end of that day.

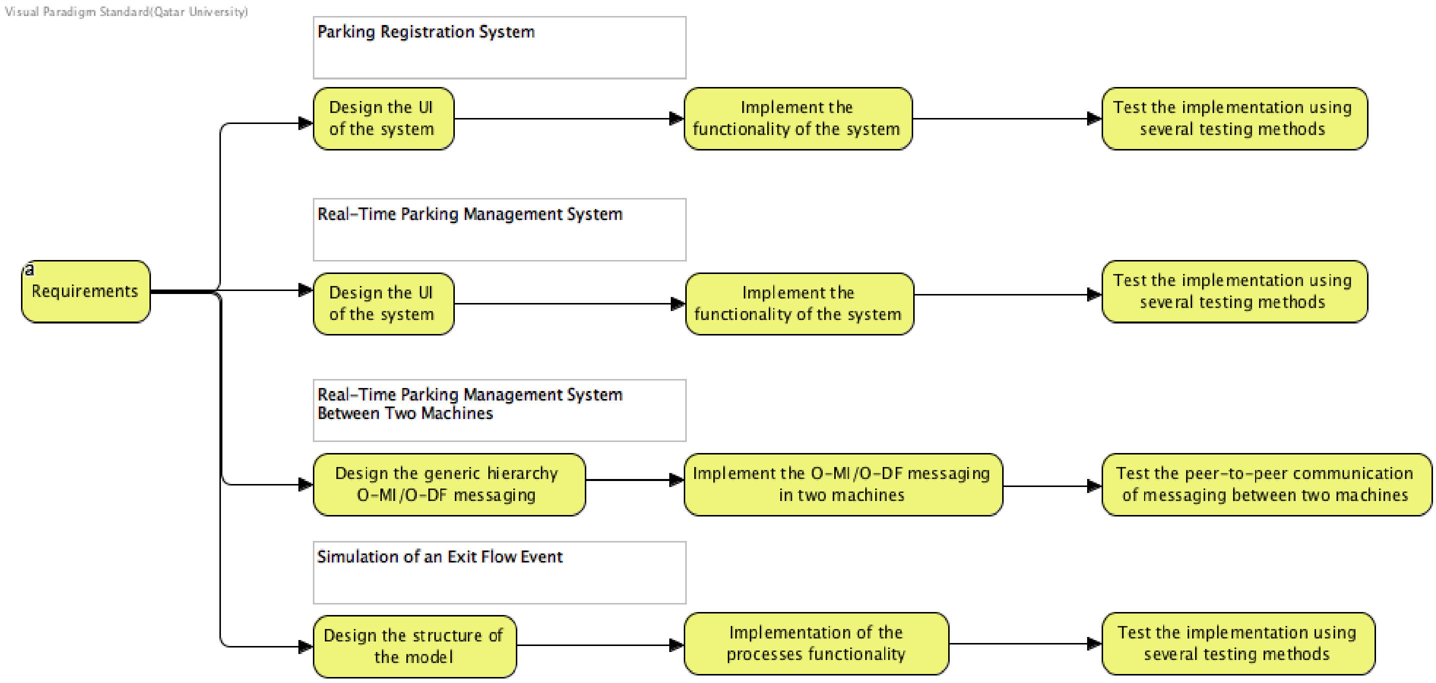



**Figure 6. Incremental Diagram of Smart Parking System**

Since the proposed system considered to be a big project, which involves both hardware and software to simulate a real-world smart system, the incremental model is used to overcome the complexity of the system and break it down into manageable smaller development subsystems. Each subsystem is added incrementally once it finished to produce the final system.

Furthermore, using the incremental model, the work can be generated easily and quickly since all the subsystems can be designed and implemented at the same time which helps in saving a lot of time during the system development life cycle. Moreover, the incremental model supports separation of concerns which makes it easier to test and debug smaller iteration separately. The incremental model is flexible and is chosen to meet with the non-functional requirements of the system in being scalable and modifiable.

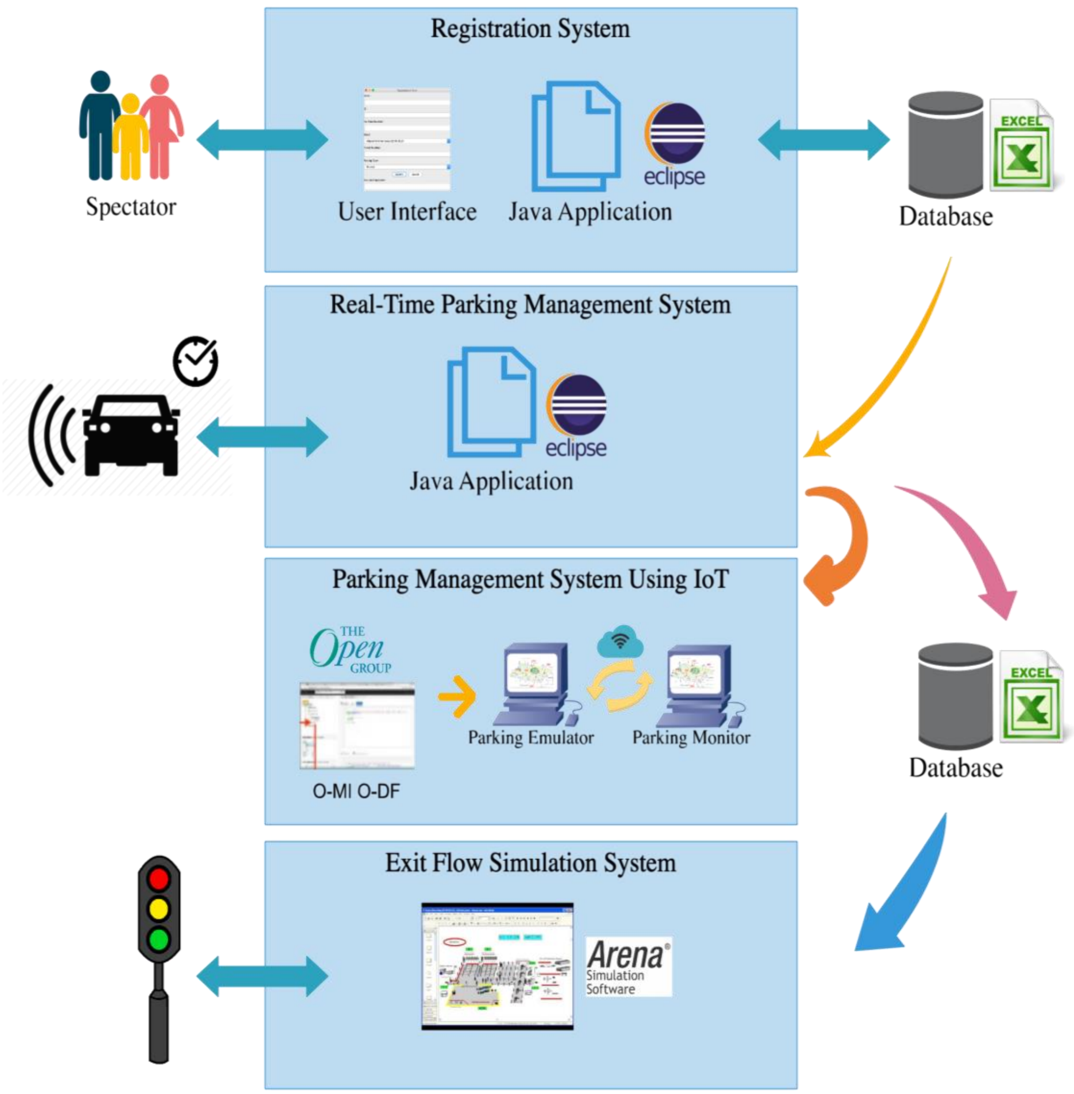



**Figure 7. High Level Architecture Diagram of Smart Parking System**

The high-level architecture diagram shows the 4 subsystems and the communication between their components.

- **Parking Registration System**:
  Interact directly with the spectator through a user-friendly interface and writes the spectators' registration information to the database.

- **Real-Time Parking Management System:**
  - Scans the car plate number of the arriving spectator in real-time and get authorization of the car from the registration database using Java Application.

  If the car is authorized, it will allocate a specific parking space for the car. If it is not, it will guide him to register for a new parking space.
  - A new database file is generated containing information about the attended spectators such as car plate number and arrival/departure times.

- **Parking Management System Using IoT:**
  Using O-MI/O-DF messaging, a peer-to-peer parking management is developed. One machine, Parking Emulator will assign the arriving spectators' car plate number into different parking spaces which then using Internet will be sent and shown in another machine which will be monitoring and taking decisions based on this assignment.

- **Exit Flow Simulation:**
  Gets the number of exiting cars which are coming to the traffic from the database and will manage red and green lights accordingly to make sure that the cars will flow faster.

## 4.3 Structural model

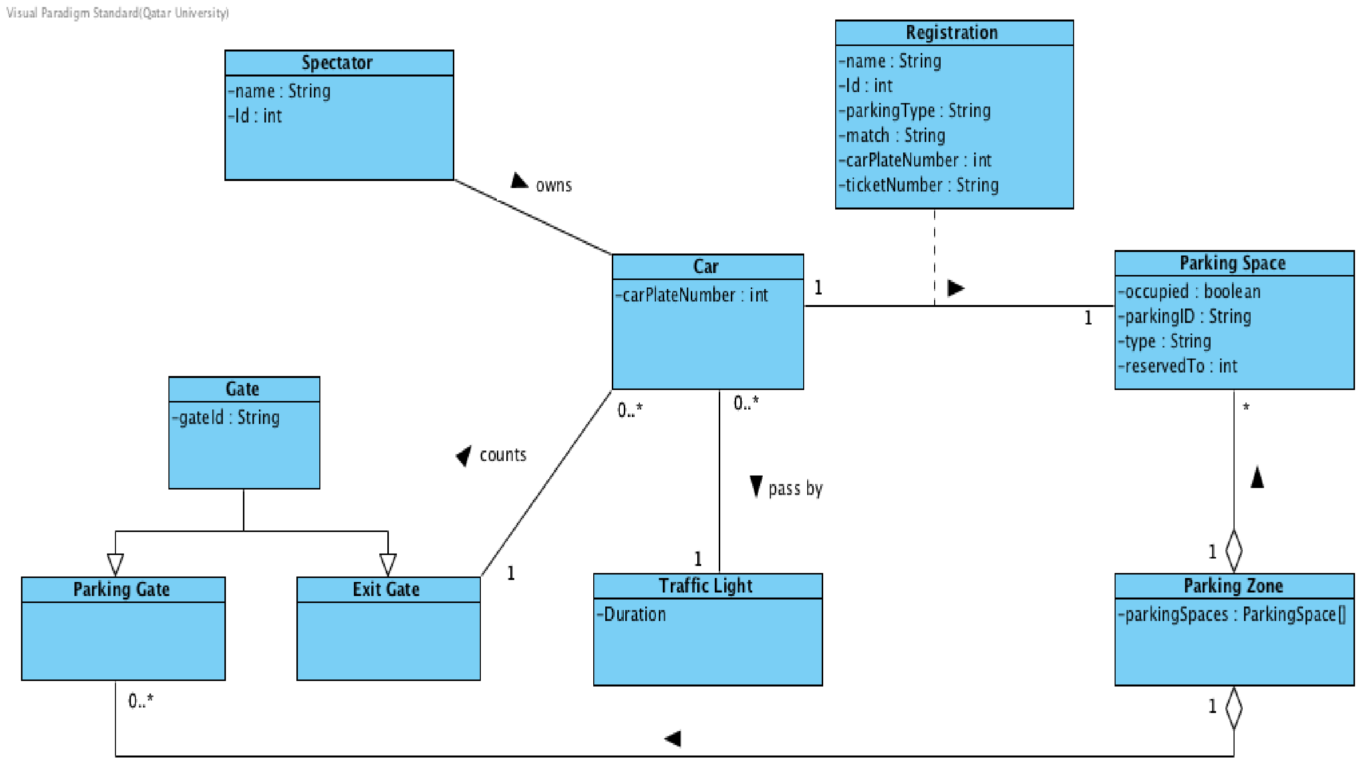



**Figure 8. UML Class Diagram of Smart Parking System**

The UML Class diagram shows the main entities of the final system with all the necessary associations between them. The system's main user is the spectator who uses the registration system to register for a parking in an upcoming match with his or her car plate number. Registration is an association class which saves all the private data of the spectator. Each spectator is only allowed to register for one parking space within his car plate number. A parking zone contains multiple parking spaces and parking gate which will give authorization for authorized cars only. In the case of an exit flow event, the traffic light will respond based on the number of spectators leaving the exit gate.

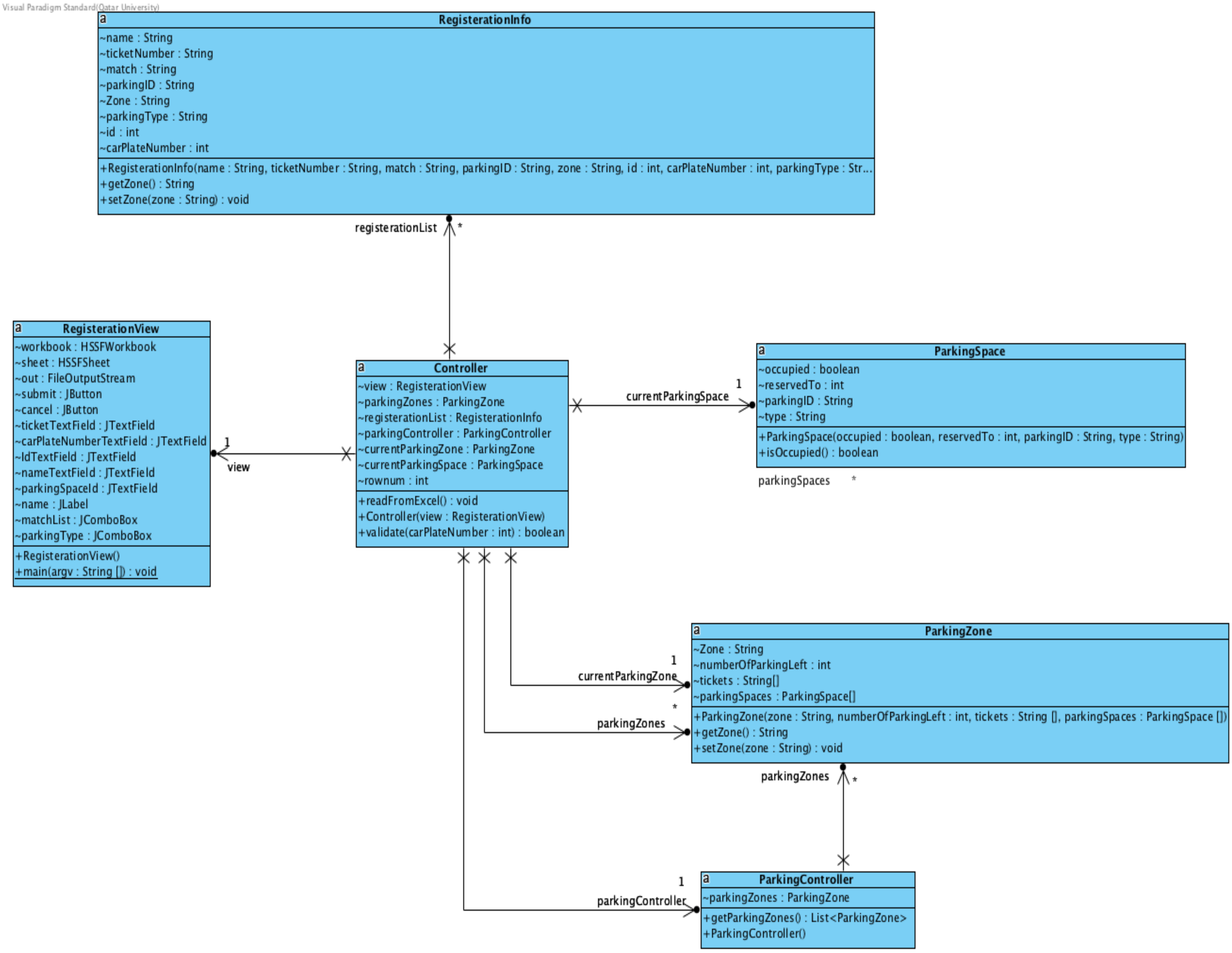



**Figure 9. UML Class Diagram of Parking Registration System**

Indirection pattern is used and more particularly, the model-view-controller pattern. The diagram introduces two controllers, *Controller and ParkingController* as a mediator between different components. Each interconnect with the representation of the data (view) and the data component itself (model) such as *ParkingZone* and *ParkingSpace*. The indirection pattern supports low-coupling and high cohesion. Moreover, the creator pattern is used in the diagram as some classes are responsible for creating other classes. *ParkingZone* class is responsible for creating multiple parking spaces by means of *ParkingSpace* class. In addition, giving the *Controller* class the responsibility to access other classes' functions as it has all the information needed reflects the use of Information Expert principle.

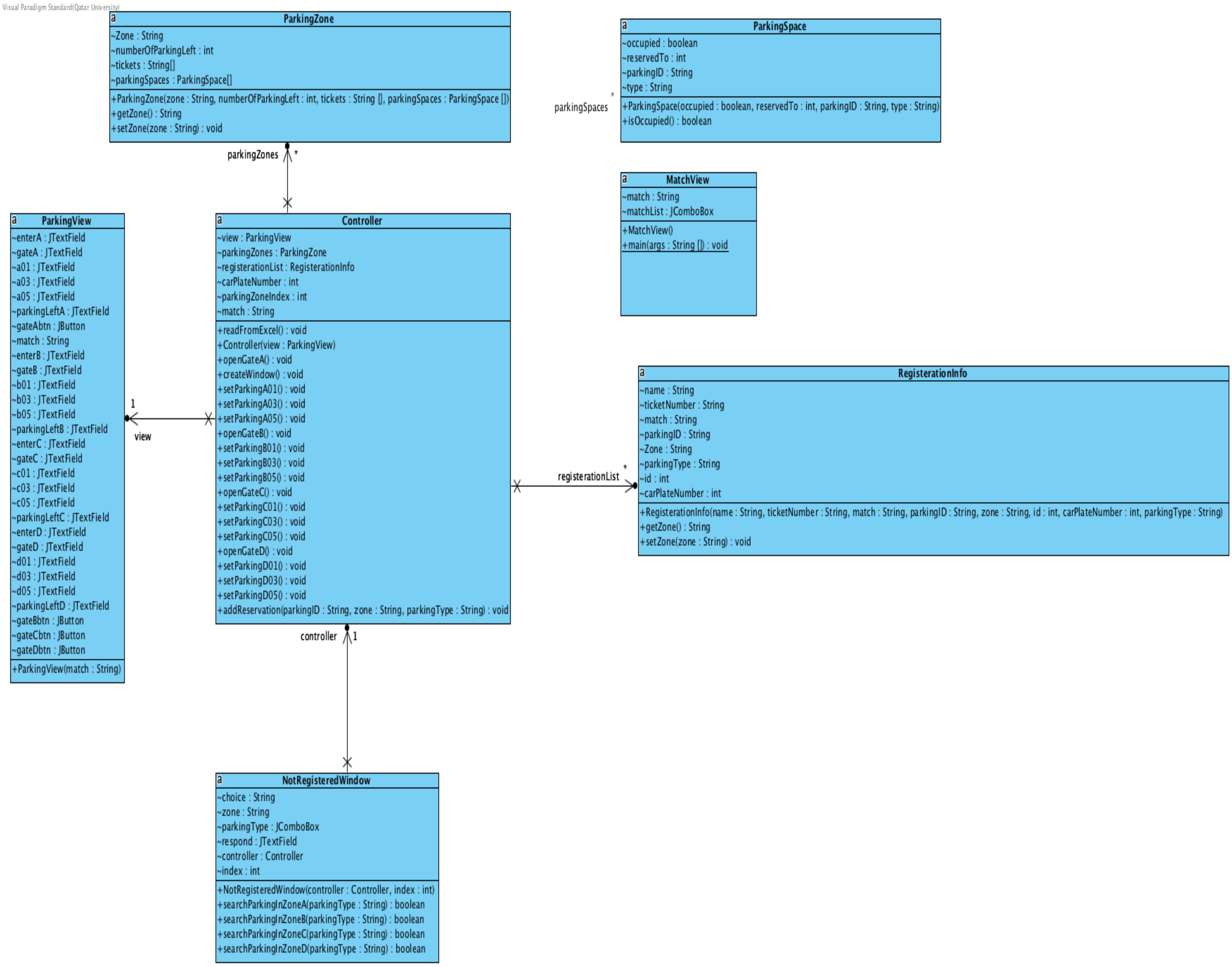



**Figure 10. UML Class Diagram of Real-Time Smart Parking Management System**

?? describe the other system

The system considered to be a continuation of the previous system as it uses some of the classes within the Registration system inheriting the effectiveness and efficiency of the designing principles implemented previously. Nevertheless, encapsulation is used in some of the system classes such as *ParkingZone* and *RegisterationInfo* to prevent unauthorized spectators from accessing the internal data about the parking zones and registration information. However, it gives the controller only the authority to access private information and reduces the overall complexity of the system. The system is designed using model-view-controller design pattern which enhance the readability and maintainability of the system targeting the non-functional requirements mentioned in previous stages of the development process.

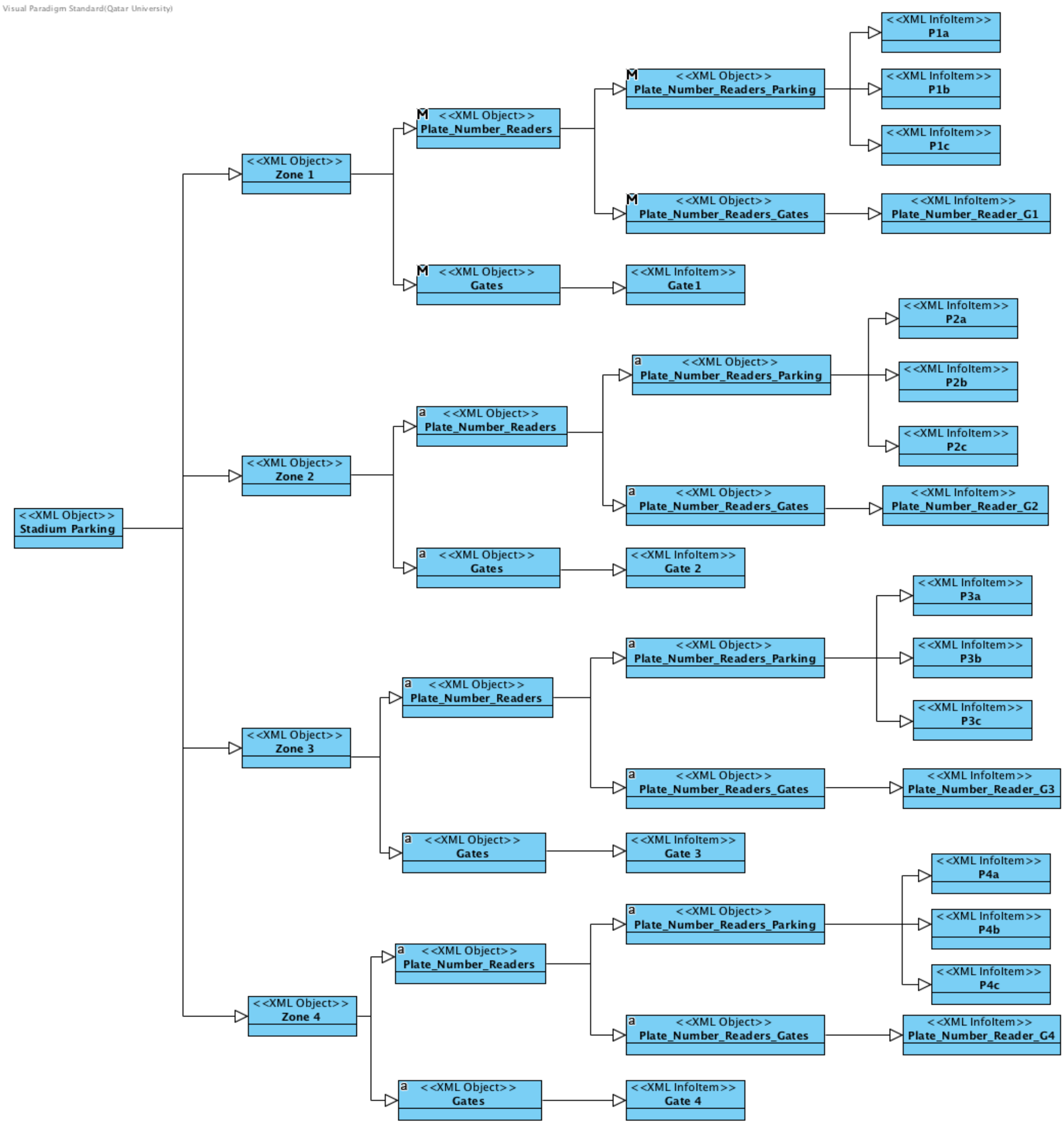



**Figure 11. UML Class Diagram of the O-MI/O-DF tree for the Smart Parking Management System.**

The UML diagram shows a representation of XML schema generated using O-MI/O-DF messaging standards. It is represented as hierarchical tree structure with a root node and different subtrees consist of a parent node and children. The nodes are either *XML Object* or *XML InfoItem* which are in XML instance document are markup tags. In the world of Internet of things and O-MI/O-DF messaging standards, objects can be seen as directories and InfoItem represents a sensor to be used to exchange data through a network connectivity. The root node is Smart Parking which includes four different Zones. In each Zone, we have three different sensors for each parking space and a gate sensor to read car plate numbers at real-time. O-MI nodes can send data to different components at any time which aligns with our smart parking system's non-functional requirement in being available always at any circumstances. Also, O-MI is very flexible as it can be used for transporting payloads in any format (one of the most commonly used is XML) and thus, it can effectively make the system more modifiable which requires low coupling and high cohesion.

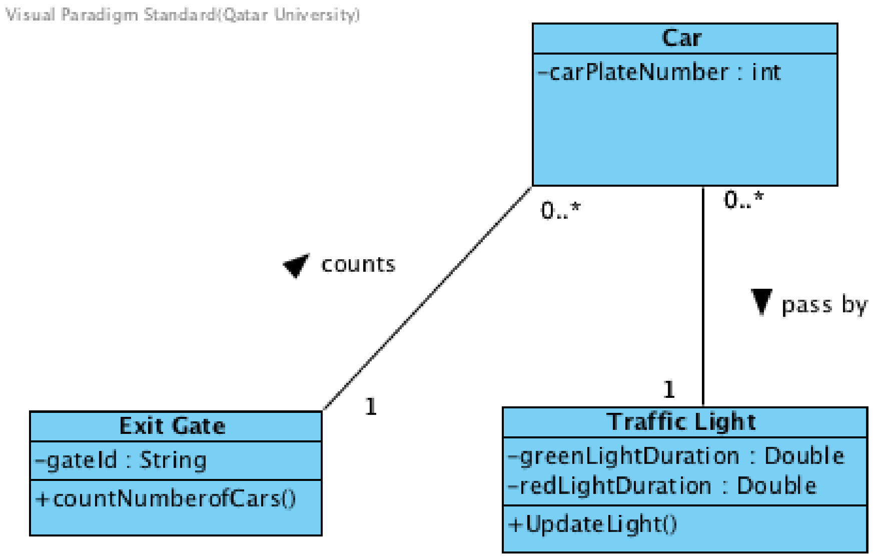



**Figure 12. UML Class Diagram of the Exit Flow Simulation.**

The exit flow simulation will have three main entities that are connected. At an exit event, spectators will be exiting the stadium from an Exit Gate which will count number of cars and sends it to the traffic light to update light based on given statistics. If more cars are exiting, longer will be the duration of the green light to smooth the flow and avoid traffic congestion.

## 4.4 Behavioral model

We developed both activity diagrams and sequence diagrams for our subsystems. Activity diagrams were developed to describe the dynamic aspects of the system and captures the flow of the system between its main activities. Besides that, for each subsystem, we developed a sequence diagram which describe a more specific flow of the system in terms of method calls and return responses.

### 4.4.1 Activity Diagrams

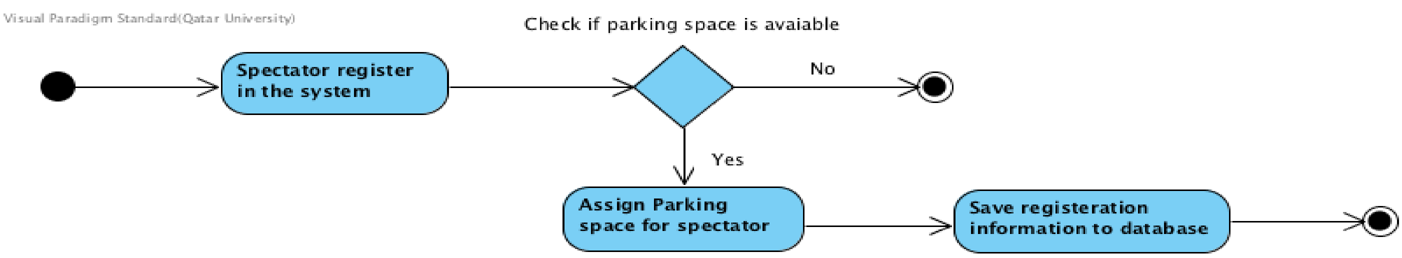



**Figure 13. Activity Diagram of Parking Registration System.**

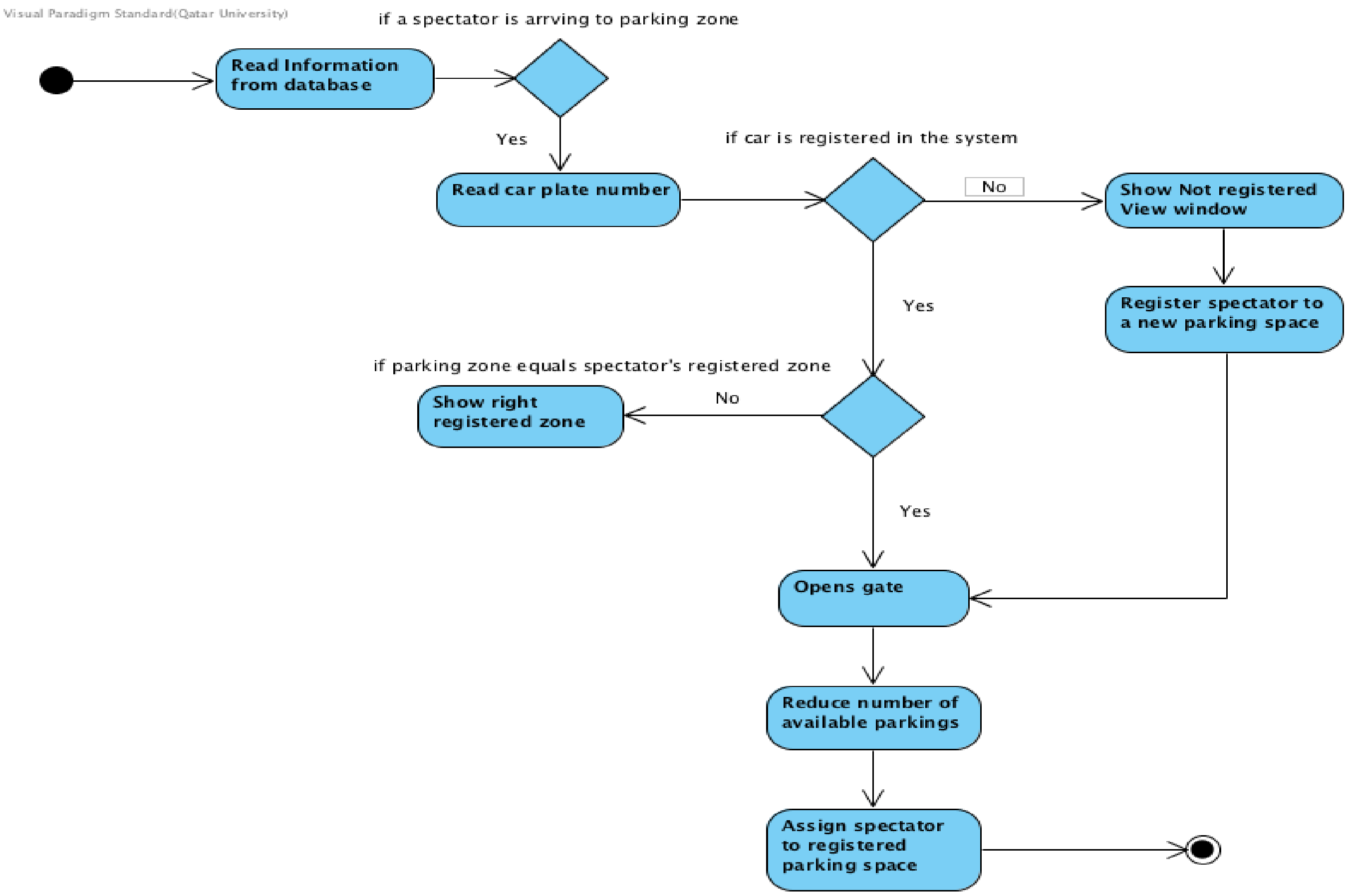



**Figure 14. Activity Diagram of Real-Time Smart Parking Management System.**

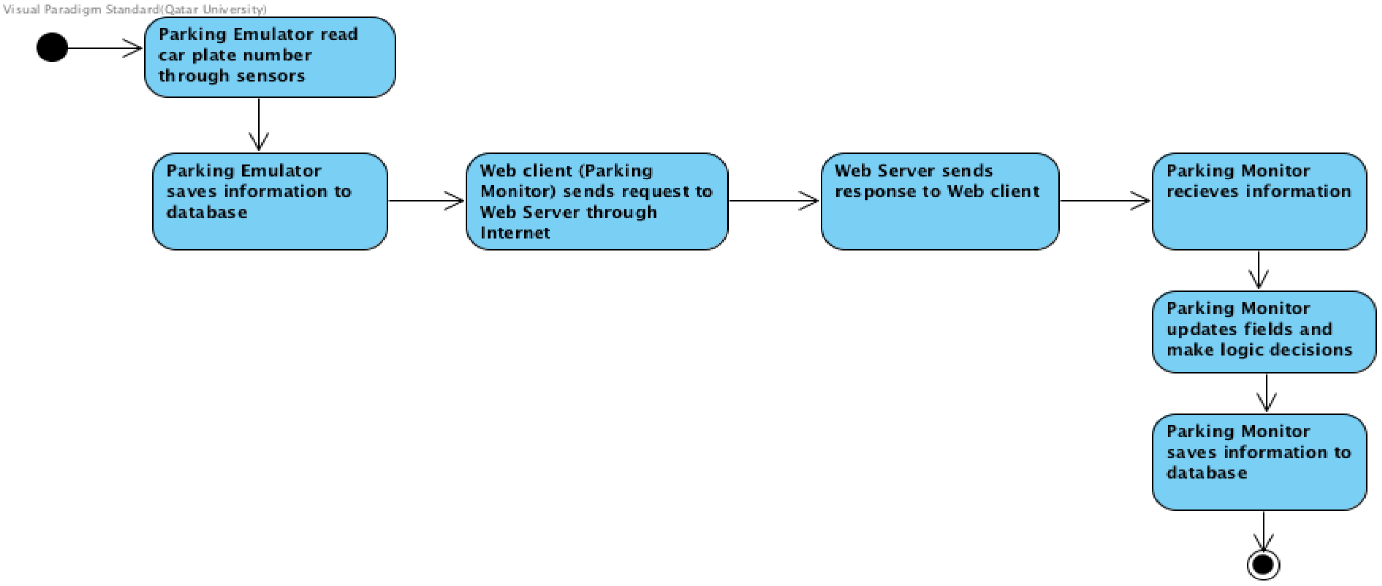



Figure 15. Activity Diagram of Smart Parking Management System Between Two Machines.

what if the light is already green?

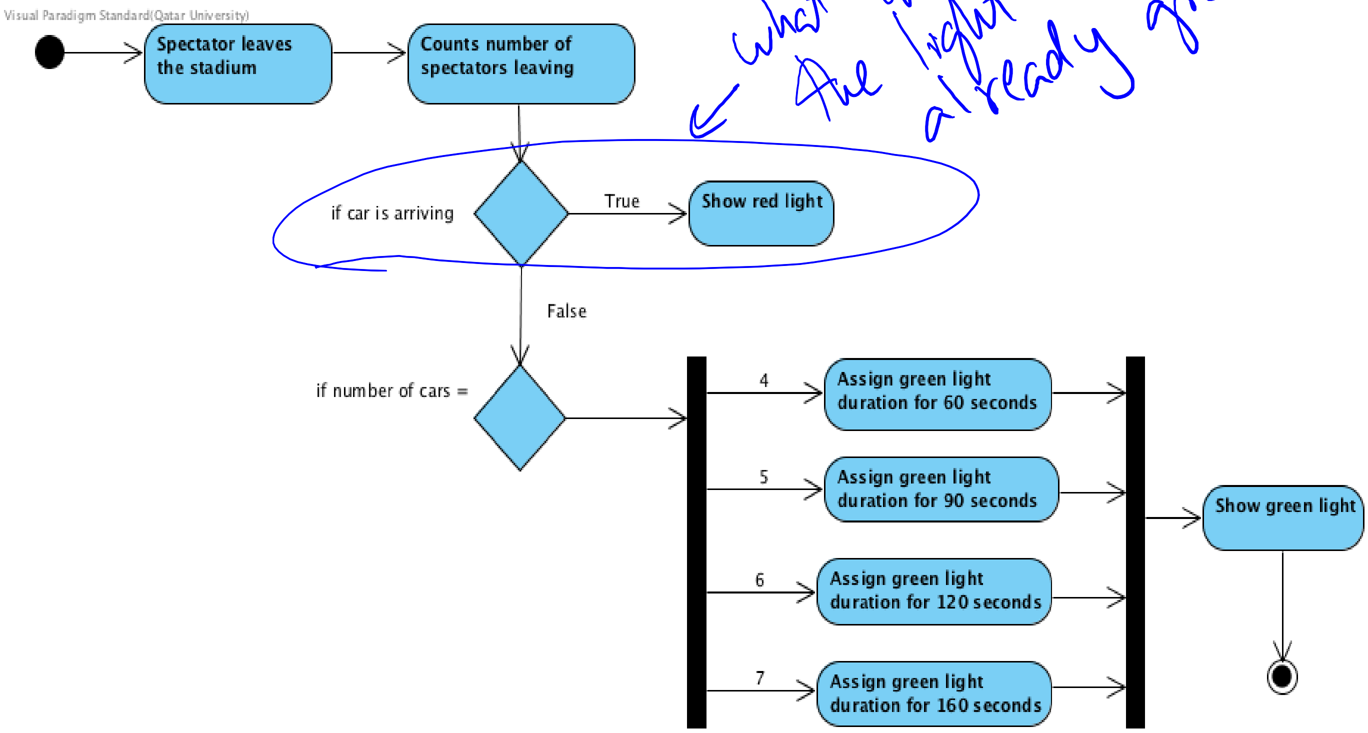



Figure 16. Activity Diagram of Exit Flow Simulation.

### 4.4.2 Sequence Diagrams

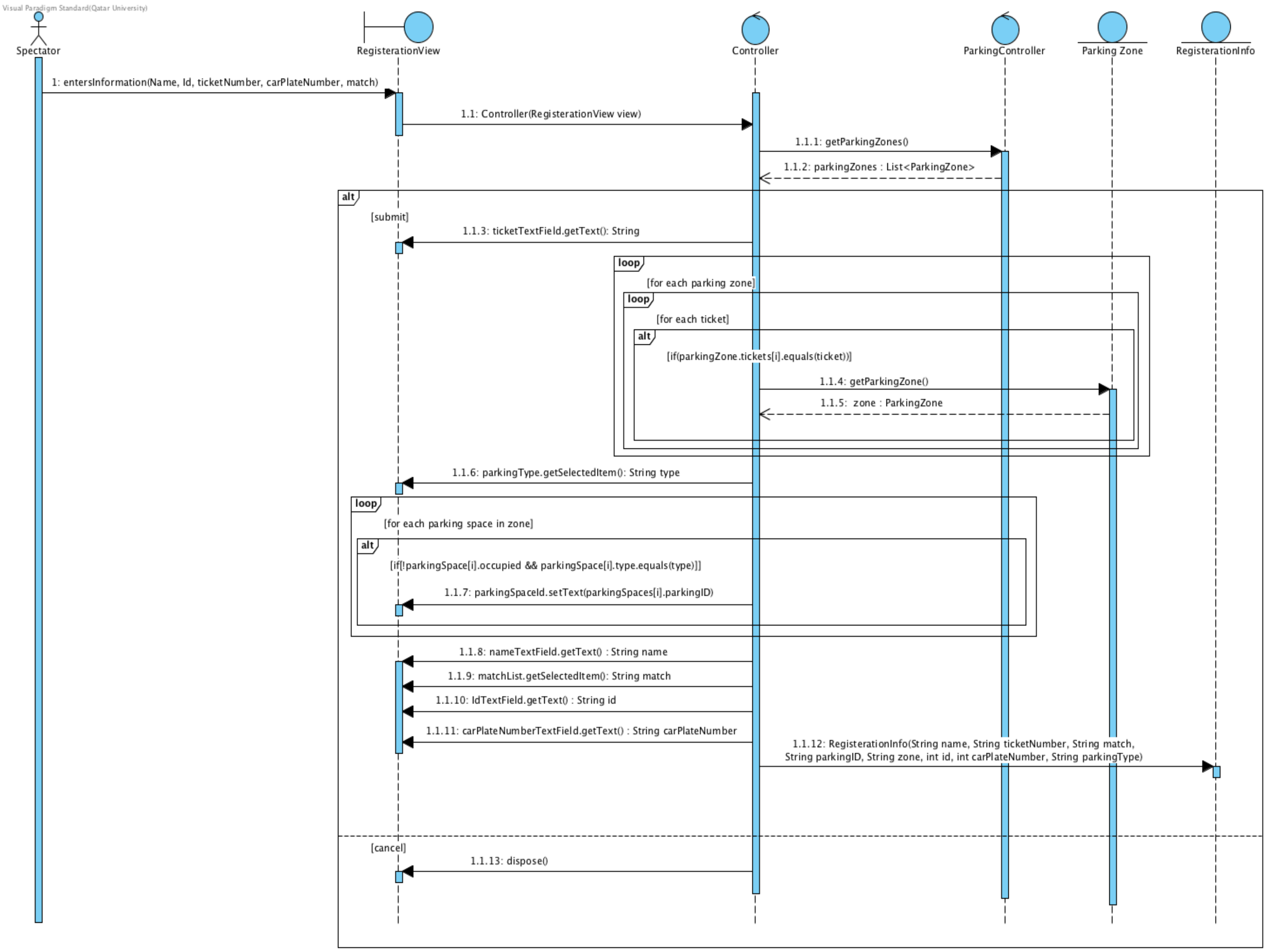



**Figure 17. Sequence Diagram of Parking Registration System**

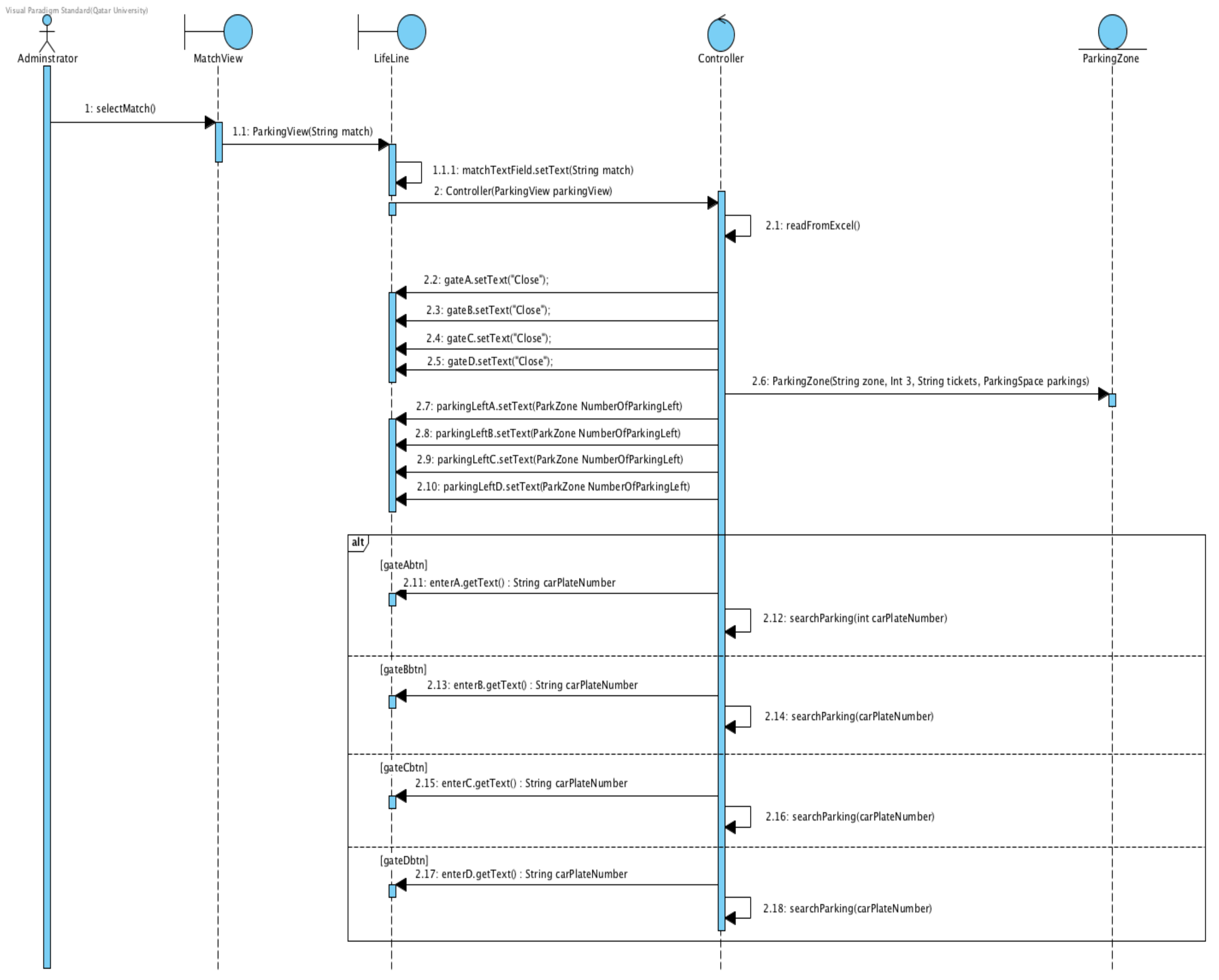



**Figure 18. Sequence Diagram of Real-Time Smart Parking Management System.**

## 4.5 Design principles and patterns

### 4.5.1 Design pattern used

We have used **Model View Controller (MVC)** pattern in our system. We chose to implement MVC pattern because it is more suitable for what were trying to implement. MVC is mostly useful for Object Oriented programmed systems that have Graphical User Interface (GUI) applications. It fits our system, as we needed to simulate the parking system using a GUI for user simplicity. We also needed to separate the concerns of each component in the system for simplicity and to ease our process of implementing the system as well as for better understanding of the problem. In our system, we needed to separate the model, which are the classes that represent the data to be stored (which are ParkingSpace, ParkingZone and RegisterationInfo) from the view, which is basically the user interface (In our system, the ParkingView and the RegisterationView) form the controller classes, which does all the computation and processing between the model and taking the inputs and showing the output from and to the view (in our system Controller and ParkingController). MVC design pattern is also most appropriate for our system as it gives the benefit of low dependency between the user interface, which may need to go through changes in the future to meet end users satisfaction or may need to go through updates more than any other component, from the other components of the system.

Also, the IoT sub-system we developed uses **Observer** pattern in the sense that one O-MI node, which is the Parking Monitor in our system, add itself as an observer that reflects and acts upon the other O-MI node's events, which is Parking Emulator in our case. It happens through changing the user interface to reflect the events of the other OM-I node. It is developed to have an Observer pattern, as it is useful for most IoT application so that products communicate with each other directly.

### 4.5.2 Design principles applied and benefits

- **Separation of Concerns***:*

Our system highly supports separation of concerns principle in the way that the components with different concerns are implemented separately in classes. For instance, the user interfaces are separated in classes from classes that preforms the computation and deals with the user action preformed, which are also separated from classes that holds the data or have the model of the data to be stored. Each class in our implementation holds a different concern, for instance, in our registration system; the user interface is separated in a class from the controller, which preforms the computation due to the action listeners of the user actions. Also, the parking controller class is separated from the controller as it has a different concern of holding the data and computations only specific to the parking areas. Moreover, the methods in each class are separated in concern from other methods in the class in preforming a very specific task of is own.

- **Re-usability and Flexibility:**

The system is broken down to small functions that each preforms a small task that works together to accomplish the tasks of the system as a whole. For instance, in the controller classes the methods are each broken-down to preform the smallest function possible, this benefits our system a lot in improving the understandability, re-usability, flexibility and the reducing of code duplication of the methods and the system as a whole. It also helps through out the implementation of the system due to the independencies of the components, as the changing of one method does not require the effort of changing a huge part of the system.

- **High Cohesion and Low Coupling:**

Also, applying such principle making our system high cohesion in the way that each method implementation is highly connected together and focused, and each class functions in a single class are functionally related and preform methods that complete a specific job. For instance, breaking down the controller in our registration system to two controllers (ParkingController and Controller) improved the system in a way that it narrowed the focus of each of them making them highly cohesive cohesion and low coupling, it also improves and eases the modifiability of the computation part of the system in the way that modifying computations of the parking does not effect the computation of the registration system as a whole as they are separated classes, nor does it require the change of one class in the occasion of changing or modifying one of them. Low coupling is also applied to other parts of the system. For instance, in the model part of the system, ~~the~~ we have an instance class that is ParkingSpace which holds the information related to a single class and a ParkingZone which holds an array of ParkingSpace instances along with other attributes, separating those two model classes increased the independencies in a way that modifying or adding attributes to a single parking space does not have to affect the other nor require the parking zone model class to be modified as a whole.

## 4.6 Database design

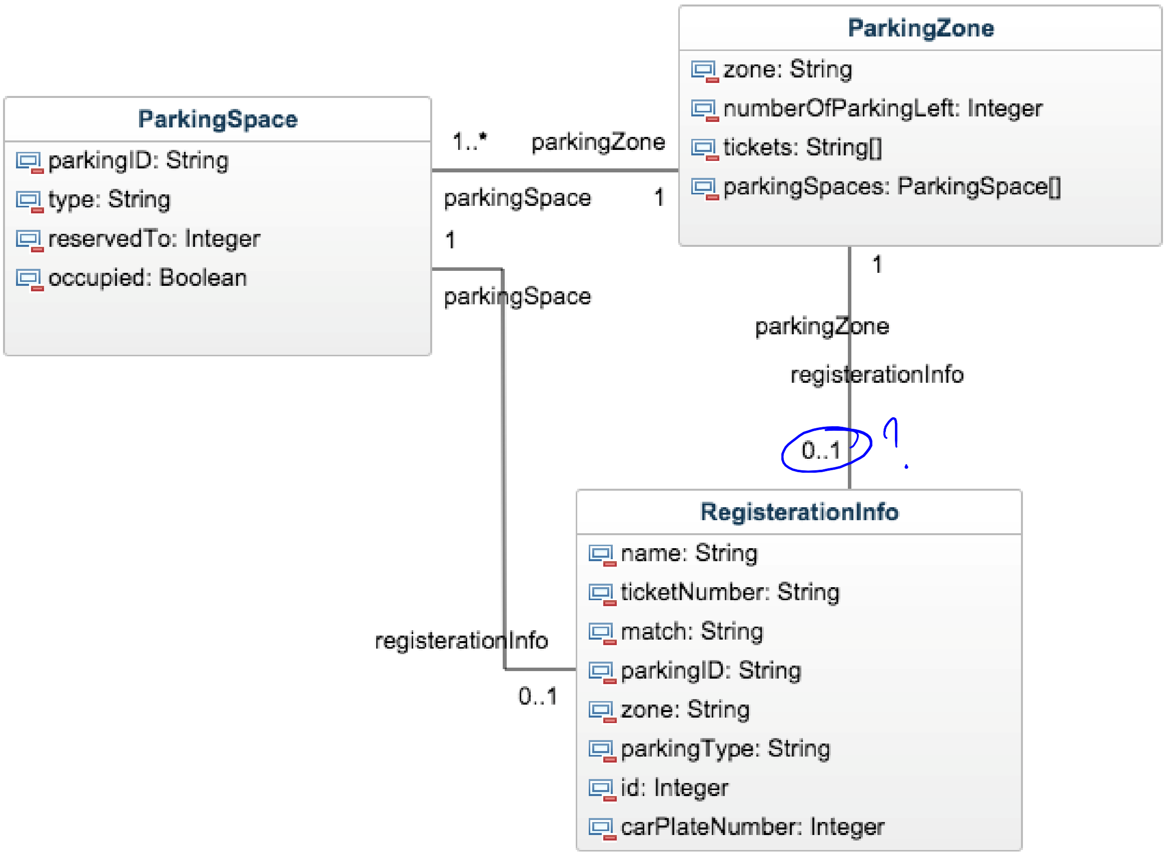



**Figure 19. Database Design Diagram.**

In our project, we simulated the database as an excel file. The major entities of the database are ParkingSpace, ParkingZone and RegisterationInfo.

The ParkingZone holds attributes we are interested in in a parking zone such as the zone attribute, for instance, zone A, B, C or D. The zone attributes in the ParkingZone class is the primary key as it should be unique in all zones. Also, we have other attributes such as numberOfParkingLeft representing number of parking spaces left in a zone and an array of tickets that have the seats number nearest to this specific parking zone. Finally, the last attribute parkingSpaces, which is an array of object ParkingSpace, which is a class that hold information about every single parking space in a zone. ParkingZone class hold information in some attributes that does not change throughout the running time of the system such as the zone and tickets, they are instantiated to represent a specific parking area and they can be changed when implementing the system on a different parking area only.

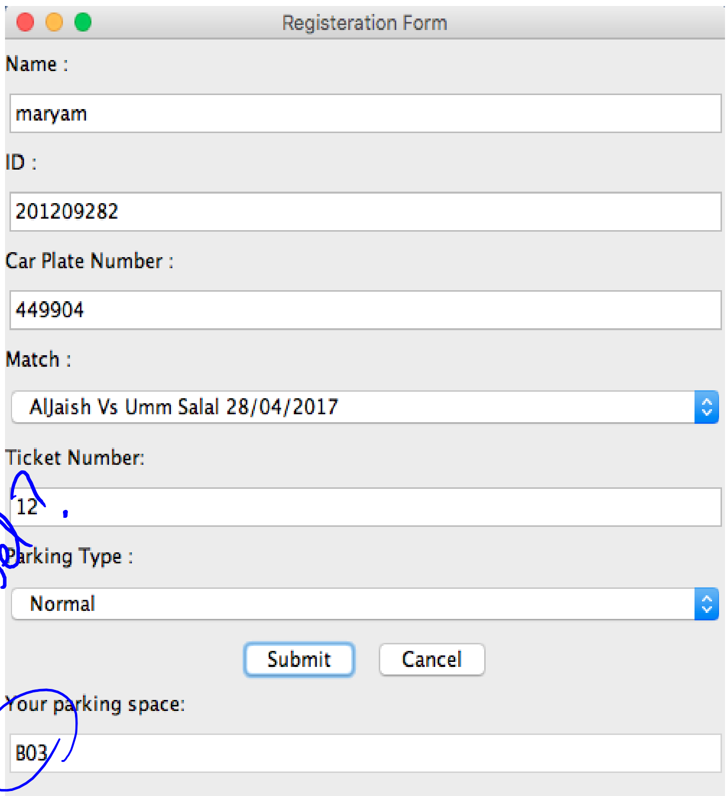
The ParkingSpace holds data about every single parking space such as parkingID, which is the primary key, type of the parking, whether it's a normal parking, handicapped, parking or parking for electric cars. Also, is has a reservedTo attribute which is assigned a car plate number whenever a spectator reserves this parking space. Finally, it has occupied attribute that holds true if this parking space is occupied at the moment. Although there is no attribute that links a parkingSpace to a zone, the link of these two classes can be seen through the array of ParkingSpace in the ParkingZone class.

Finally, we have the RegisterationInfo class which holds data after registration phase. This class holds data about spectator such as name, ID and carPlateNumber, as well as other information about the match that this spectator is registered for parking on such as ticketNumber and match along with other attributes about parking such as parkingID, zone, parkingType. This class can be linked to the other two classes through parkingID and zone.

## 4.7 User interface design

### 4.7.1 User interface design for Parking Registration System

The user interface implemented in the Parking Registration System is user-friendly. It is very clear and straight-forward so that any spectator at any age can use it even elderly spectators. There is no way for confusion or frustration. Moreover, the UI of the Parking Registration system is familiar. Parking Registration System is very formal; it appears to users like something they have encountered before which will make them know~~s~~ how the system behave even before using it. The User Interface of the Parking Registration System has multiple text fields which reads the user's input such as his name, ID, car plate number, match, parking type and ticket number along with two buttons. Once a user interacts with the buttons, the system will start functioning by reading all the inputs provided by the user. Then, the system will responsively interact back to the user by sending his specified zone and parking ID. The Parking Registration System will validate the car plate number and ticket number and show a dialog box in case of unauthorized actions or duplication. The user interface has high efficiency as it interacts with the user very fast with less effort.




Since this is not changed by the user, maybe use (JLabel)

Figure 20. User Interface of Parking Registration System.

### 4.7.2 User interface design for Parking Registration System

Smart Parking Management System has two different interfaces one followed by the other. The system will be monitored by an administrator who might be trained before on how to use it, however, the interface is still simple and clear using unambiguous language familiar to users. The first view shows a dropdown indicating all upcoming matches, once the administrator selects a match, another interface will be shown. The second interface has multiple elements yet it is very attractive and concise as it uses colors to indicate different parking spaces. A stadium view with all zones, parking spaces, gates and parking occupancy signs will be shown. The system is following the feedback principle of user interface design; it will keep the users informed of actions or interpretations efficiently and quickly.


**Figure 21. Match View of Smart Parking Management System. (User Interface I)**

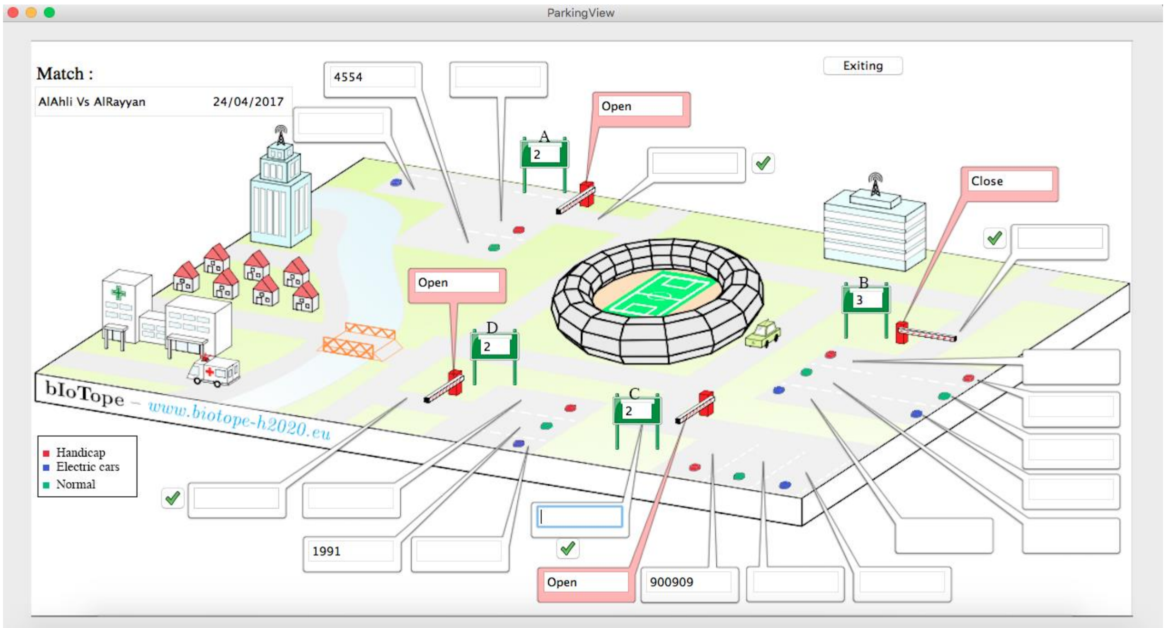



**Figure 22. Parking View of Smart Parking Management System. (User Interface II)**

### 4.7.3 User Interface of Smart Parking Management System Between Two Machines using IoT

The user interface of this system is very simple and it is the same for both machines. Reusing the same interface reduces the need for users to rethink, remember and train targeting the reuse principle of user interface design. If there is a strong connectivity (Internet/Wi-Fi), the communication between the interfaces of both systems in two machines will perform very fast. In the parking emulator machine, the stadium view is shown with filled text fields indicating the assigned car plate number for each parking space. In the parking monitoring side, an extra button is shown which when pressed, the system using Internet starts receiving responses from parking emulator and show it back to a different user.

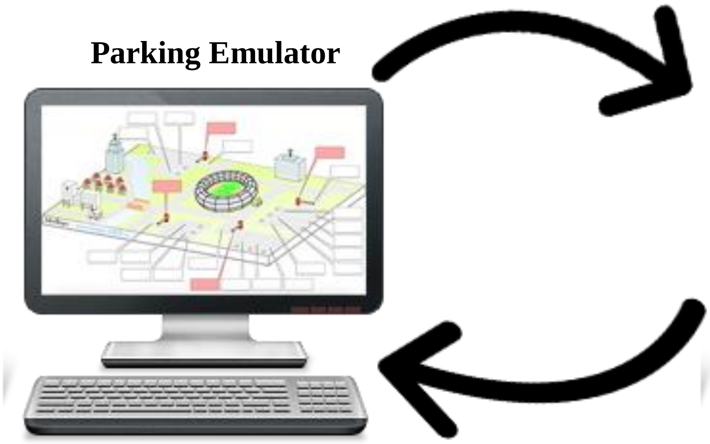



Figure 23. User Interface of Smart Parking Management System between two machines.

### 4.7.4 User Interface of Exit Flow Simulation.

The exit flow simulation interface doesn't have any direct interaction with the user however, it is very attractive and colorful to enable audiences to observe the connection between cause and effect in real time in an easy understandable way. The simulation interface shows a view of cars' queue and connection with traffic lights.

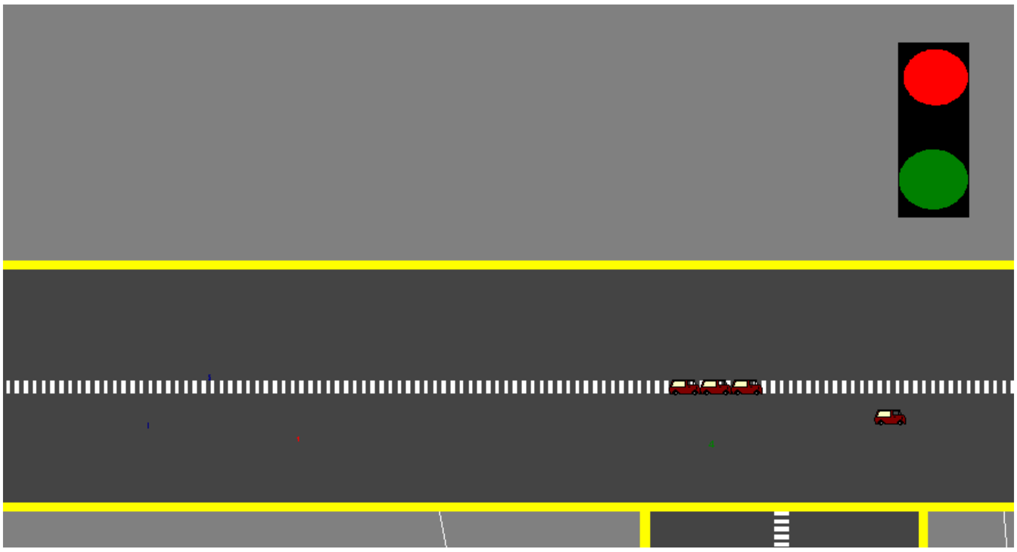


Figure 24. User Interface of Exit Flow Simulation.

## 4.8 Hardware/software to be used

**Hardware:**

- Two Laptops:
  to simulate the communication between the nodes.

**Software:**

- The Open Group Internet of Things standards (O-MI/O-DF IoT):
  to implement the communication using O-MI and O-DF. O-MI is the messaging interface containing the time-to-live value, message type (whether a request or response message) and message format. The O-DF is the message format containing the structure of the message.
- Eclipse for Java:
  To implement the computational logic of the system preformed according to the message received.
- Microsoft Excel:
  To simulate the database by holding the data used by the system.
- Arena:
  To simulate the communication with the Traffic Signals system.

## 4.9 Alternative solutions and tradeoffs

### 4.9.1 Possible Solutions of IoT and Tradeoffs

Developing Smart City Systems can be implemented using different IoT platforms. One of the platforms used is Labeeb$^{TM}$ IoT Internet of Things, which is the first locally IoT platform in Qatar developed by Qatar Mobility Innovations Center. Another one is WAVIoT which provides as easy access to all data gathered from IoT devices and sensors and is specifically designed for analysis, data storage and processing. The last one is The Open Group Internet of Things. Although all platforms target international markets, we chose to work with Open Group Internet of Things (O-MI and O-DF) due to some advantages, the following table summarizes the good and bad points of each system according to identified indicators:

**Table 5. Comparison of possible solutions for IoT and their tradeoffs.**

| | ***Labeeb$^{TM}$ IoT* [1]** | *WAVIoT* | ***Open Group IoT*** |
|---|---|---|---|
| Open Standards | No | No | Yes |
| User friendly interfaces | Correct | Correct | Enhanced |
| User experience | Needs training | Needs training | Needs ~~fast~~ less training |
| Simulation | Not provided in the basic version | Not provided in the basic version | Not provided in the basic version but the connection with the simulation exist |
| Real-time monitoring of parking occupancy | Yes | Yes | No |

We chose to work with Open Group IoT since its advantageous to our system over the other platforms especially that it is an open standard and publicly available. However, other platforms may have the advantage of real-time monitoring of parking occupancy, but our system may not need this feature since it is designed to perform this task.

[1] QMIC Qatar. *Https://Www.Youtube.Com/Watch?V=2G8a1ah7_QY*. 2016. Print

### 4.9.2 Possible Solutions of Exit Flow Simulation and Tradeoffs

Developing the simulation of the exit flow of cars by simulating the smart traffic and how it reacts to various numbers of cars exiting to ease the flow can be established using different ways in Arena. It can be simulated using generating data with probability, which is basically generating the number of cars exiting the stadium randomly using probability. Another way is, by connecting the two phases of the project and taking the number of cars exiting simulated in the previous phase and tests it in this phase after or during running time of the previous phase. Each of these ways have some advantages over the other:

hard to understand what this means

**Table 6. Comparison of possible solutions for Exit Flow Simulation and their tradeoffs.**

| | Generating data using probability | Using the data from previous phase after running time of it | Using the data from previous phase during running time of it |
|---|---|---|---|
| May cover more various numbers | √ | | |
| Enhance connection between different components of system | | √ | √ |
| Real-time simulation | | | √ |

However, we chose to implement the simulation using the third way, as it is more advantageous over the others because it enhances the connection of the different parts of the system and provides a better visualization and understandability as it provides real-time simulation. It may have the disadvantage of not covering more various numbers because numbers are entered manually, however, this can be avoided by using probability for testing the system and using a different range of numbers while simulating to audience.

### 4.9.3 Possible Solutions of the System Simulation and Tradeoffs

Our system can be developed using different approaches to simulate different Smart Parking and Smart Traffic services. It can be implemented as a single system handling and managing simulations: Registration System (Before the event), Simulation of real-time smart parking system (At the time of event) and Simulation of smart traffic. It can also be implemented on different sub-systems, each focusing on related functionalities of the system. Each approach has advantages and disadvantages shown in the table:

**Table 7. Comparison of possible solutions for System Simulation and their tradeoffs.**

| | *Implementing Smart Parking and Smart Traffic System as a single system* | *Implementing Smart Parking and Smart Traffic System as sub-systems* |
|---|---|---|
| **Realistic** | Less realistic, it can be seen that implementing registration as a combined system with smart parking system and smart traffic system can be unrealistic simulation of real-life situation. | More realistic, implementing registration as a separated system, yet has a communication with smart parking system is more realistic simulation real world and more understandable to audience. |
| **Complexity** | More complexity, as handling different functionalities that are separated in real-life but combined in simulation is far from real and can be very complicated as it has low separation of concerns and hard to be implemented. | Less complexity, more user-friendly and understandability. |
| **Coupling between components** | Coupling between different components is very high, and modifiability of one system can affect other systems resulting in less flexibility. | Low coupling between components. Thus, modifying one component is easier and may not affect the other sub-systems as they are separated. |
| **Cost** | Less cost because one system. | Costs more because implemented as many systems. |

# 5. Implementation

- Tools and framework used:

Through our implementation we have used different tools to try to simulate smart parking and it services in an event of a sport event.

- **<u>Phase I (Before the event):</u>** Registration Management.

  We have used **<u>Java</u>** to create a Graphical User Interface to simulate the registration that is going to happen before sport event day. Our system is supposed to take the ticket number and assumes to have previous information about the arrangement of the stadium and parking area so it will assign nearest available parking to that ticket to accomplish the objective of organizing and reducing traffic and searching time.

  The implementation of this part is basically implementing the *View* which is the GUI separated from the *Controller* which handles the user actions whenever they submit the registration information, as well as, validating the input such as car plate number by checking if it has previously registered. Also, the *Controller* communicates with *ParkingController*, which does all computations regarding parking and have all information about it saved as *ParkingZone* and *ParkingSpaces* instances. Finally, the final information is being saved as objects of *RegisterationInfo* class in the **<u>Excel</u>**. We have used **<u>Excel</u>** to simulate the database of the system.

- **<u>Phase II (At the event time):</u>** Entry Flow Management.

  - Communication

    In this layer, we have used **<u>O-MI/ODF</u>** Platform to represent and simulate the communication between parking nodes in real-life. The tree for the message exchanged between the nodes is a nested XML message that contains information about parking. The Data Format (*O-DF*) of the message is made up of the Zones (whether the Zone1, Zone2, Zone3, Zone4) inside it the Plate_Number_Readers which represent the sensor of the gate in real-life situation along with Plate_Number_Readers_Parking which can holds the ID of a single parking space. The Message Interface (*O-MI*) of the message contains values such as ttl value (time-to-live) and type of message (i.e. response, request) and other header attribute. These message attributes contain the information needed to be exchanged between parking nodes and acted upon or taking actions depend on them, which is represented in the computation layer.

- Computation

  After the communication layer should be the computation layer, which in here we have implemented an interface using **Java** to simulate the logic of the system and what computation it does on the data received from the message in the previous layer. We have implemented this part by creating a GUI that has the *ParkingView*, which represent the parking zones in real-time and how does the computation part preforms. For instance opening gates for registered vehicles by assuming sensors reading car plate number and communicating needed information in O-DF/O-MI message for gate, and gate responding accordingly. Also, guiding cars to different gates if they are coming to the wrong zone. Evermore, if a vehicle is not registered in the system the system will communicate the needed information in order to check for the nearest parking available at the moment. In order to show such functionalities, in our implementation, we have separated the *ParkingView* from *Controller,* which reads the vehicles car plate number entering gates and does the needed computation to simulate the functionalities discussed above through checking information in model class, which are *RegisterationInfo*, *ParkingZone* and *ParkingSpace*.

- **PHASE III Phase (After the event):** Exit Management.

We have simulated the exit of vehicles from different gates using **Arena**. Arena will be simulating smart traffic, as the gates will communicate the needed information of number of vehicles exiting per minute to the smart traffic and the smart traffic will adjust the lighting accordingly. For instance, if number of cars is greater that 4, the green light will stay for more than 30 seconds to accomplish the objective of easing the traffic and reduce the waiting time.

For all cases the cars are grouped into batch sets, Average Duration for each batch arrival from the parking gate to traffic light is assumed to be around 1 minute (58.2800 sec seconds ), and the probability is distributed such that 50% of the cars arrived to the traffic light will have Red light, while the remaining 50% will have Green light. The waiting time in red light is 30 seconds and does not depend on how many cars arrived to the traffic. However, the duration time in green light depends on the number of cars coming to the traffic light as shown in the table below:

How were these values selected?

**Table 8. Duration Time for Batch car sets.**

| Batch car sets | Red light duration | Green light duration |
|---|---|---|
| **Less than 4** | 30 seconds | 30 seconds |
| **Equal 4** | 30 seconds | 60 seconds |
| **Equal 5** | 30 seconds | 90 seconds |
| **Equal 6** | 30 seconds | 120 seconds |
| **Equal 7** | 30 seconds | 150 seconds |

- Some of the challenges encountered:

  During implementation, we faced some challenges in trying to make simulation closer to the real life as much as possible. In order to achieve this, we went through hard time deciding how functionalities should work exactly to be closer to how it should work in real smart parking systems and how the system as whole work together. For instance, we implement the registration sub-system separately from the other system at it happens before the event whereas the other sub-system simulates real-time parking management at the time of the sport event. We decided that implementing them this way would ease the idea of how they should be in real-time more for the users. Also, we worked on saving data in Excel and getting information from it rather than working with text files as it is closer to working with databases and it enhance the idea of working with databases in real smart parking systems even though it was more challenging and hard to work with them as it requires installing external jars and libraries.

  Even more, other challenges we faced through our implementation was working with IoT Platform as it is a new technology and the resources was limited. Also, it was challenging deciding how much information O-MI/O-DF message should contain, or how big the xml tree of the message should be. Also, it was challenging to try to link both the communication and computation parts together in an easy way that users would understand simply.

# 6. Testing

## 6.1. Functional requirements testing

### 6.1.1. Java Testing

What is the error?

**Table 9. Function Testing Summary**

| Test Case | Met/Not Met | Test Results | How to improve |
|---|---|---|---|
| **TC01 – Normal scenario** | Met | The user information is saved to the database and nearest parking is assigned for user and shown on display. | Although the task successfully passed, there is a minor error that does not affect the functionality in any way. However, it can be overcome for better implementation |
| **TC01 – Validating car plate number.** | Met | A message shown stating that car has already reserved parking space. | - |
| **TC01 – Validating ticket number.** | Met | A message shown stating that the ticket number is invalid. | - |
| **TC02 – Normal scenario** | Met | The gate reads "Open" and the car plate number of the car entering is shown on the parking space reserved for that car. | Even though the task is accomplished successfully, the gate still reads Open after the car has entered which can be improved to better enhance the image of the real system to the audience. |
| **TC02 – Entering to invalid zone** | Met | The gate reads "Go to Zone X" where X is the zone ID of the parking assigned for that car. | Show the directions to go there? |
| **TC03 – Normal scenario** | Met | A window is displayed for the user to select parking type needed for the moment and then the nearest parking available is shown for the user. | - |
| **TC03 – Full parking** | Met | A message is shown for the user stating that all parking spaces in all zones are reserved or occupied at the moment. | - |

- Test Case 1

| **TC01** |
|---|
| **Title:**<br>**Registration –** Register successfully to the system to reserve parking. |
| **Description:**<br>User should be able to register successfully to the system and get back nearest parking number to the ticket the user reserved for the sport event.<br>Precondition: The user should have already reserved a ticket for a sport event.<br>Assumption: Each ticket number has a parking space near to it. |
| **Test Steps:**<br>1. User enters all information to the system including ticket number.<br>2. Click submit button<br>3. |
| **Expected Result:**<br>1. User should get a parking ID reserved for him/her.<br>2. User Information should be stored in the system database. |

- Test Case 2

| **TC02** |
|---|
| **Title:**<br>**Parking Entrance –** Registered car plate number should be authorized to enter zone assigned for it and directed to a parking space reserved for it. |
| **Description:**<br>Registered car plate number should be authorized to enter the zone specified for it successfully. And should be given the parking space assigned for it.<br>Precondition: Car plate number already registered for a parking and given a parking ID number. |
| **Test Steps:**<br>1. Enter car plate number in entering gate space.<br>2. Click submit button |
| **Expected Result:**<br>Gate should read open and car plate number should appear in the right parking space reserved for it. |

- Test Case 3

| **<u>TC03</u>** |
|---|
| **<u>Title:</u>**<br>**Parking Entrance with Unregistered Car Plate Number –** Unregistered car should be assigned to nearest parking space. |
| **<u>Description:</u>**<br>If a car that did not reserve a parking space came to enter the zone at the event time, the system should display the nearest available parking space for that car to guide it to it. |
| **<u>Test Steps:</u>**<br>1. Enter car plate number that is not saved in the database.<br>2. Click submit button. |
| **<u>Expected Result:</u>**<br>System should display for the user the nearest parking space available among the 4 zones. |

### 6.1.2. Arena Testing

You need to describe what you intend to find out from each experiment before you present the results.

Case 1: Probability Based Exit Car Generation:

Assign a random car exiting per minute from parking gate with the Probability values of car arrivals computed using discrete distribution. 100 minutes are roughly needed to all cars are exiting from the parking gate and handle the 100 cars. The average total time for the entity within the system does not exceed 1 minute; which is considered the time for the entity which went till the end and was transferred to traffic light.

not clear

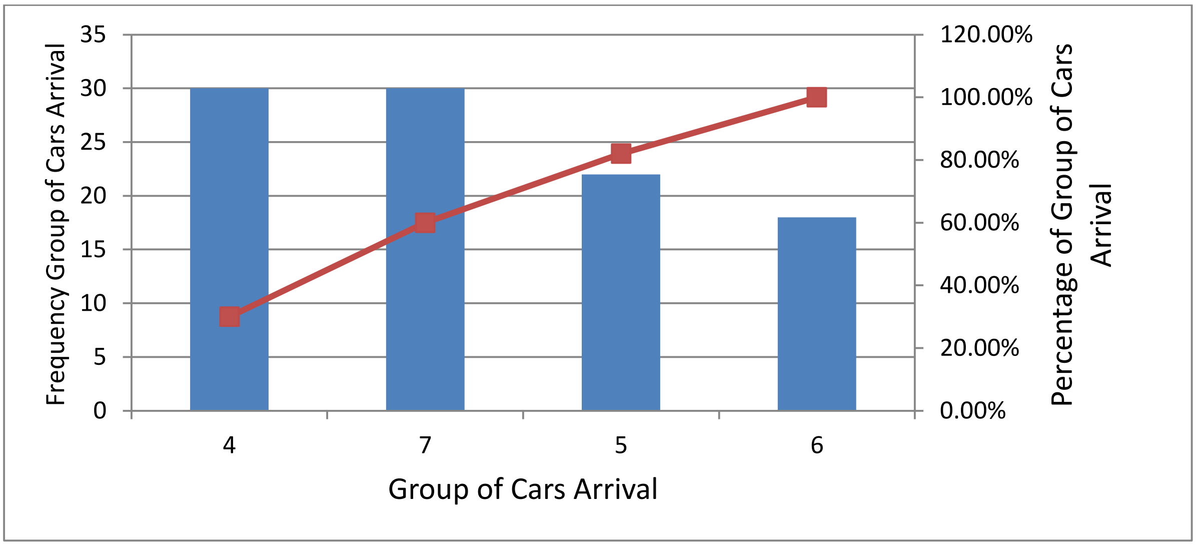



**Figure 25. Probability values of Car Arrivals.**

see next page

Case 2: Static Data Based Exit Car Generation:

In this case, we have the simulation for only the 30 first minutes. We randomly added numbers of vehicles to a static excel file, each line represents 1 minute, the static excel file have 30 lines (each line represent the number of cars that went out per minute: first line represents the number of cars in first minute, second line for the second minute … etc.)

The table and graph below shows the car numbers exiting per minute in 30 minutes:

The distribution does not seem random

**Table 10. Numbers of cars exiting per minute in 30 minutes in a static excel file.**

| Time per minute | Number of cars |
|---|---|
| 1 | 2 |
| 2 | 2 |
| 3 | 2 |
| 4 | 4 |
| 5 | 4 |
| 6 | 4 |
| 7 | 4 |
| 8 | 5 |
| 9 | 5 |
| 10 | 5 |
| 11 | 5 |
| 12 | 6 |
| 13 | 6 |
| 14 | 7 |
| 15 | 7 |
| 16 | 7 |
| 17 | 7 |
| 18 | 7 |
| 19 | 7 |
| 20 | 6 |
| 21 | 6 |
| 22 | 6 |
| 23 | 5 |
| 24 | 4 |
| 25 | 3 |
| 26 | 3 |
| 27 | 2 |
| 28 | 2 |
| 29 | 1 |
| 30 | 1 |

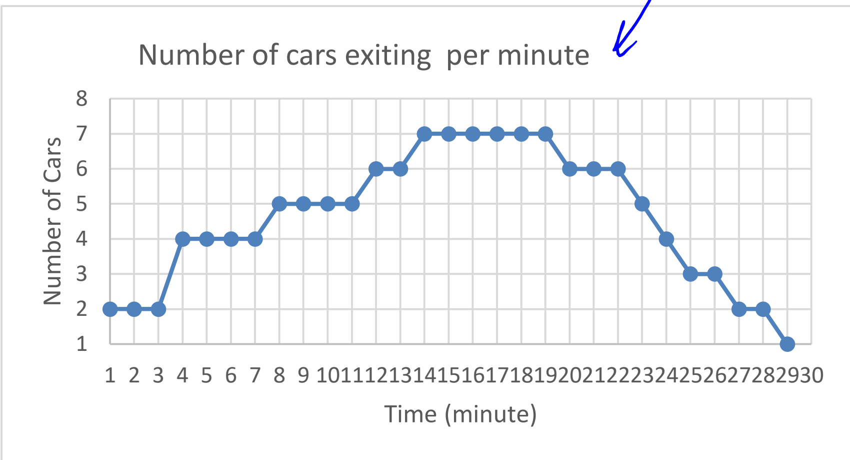



**Figure 26. Numbers of cars exiting per minute in 30 minutes in a static excel file.**

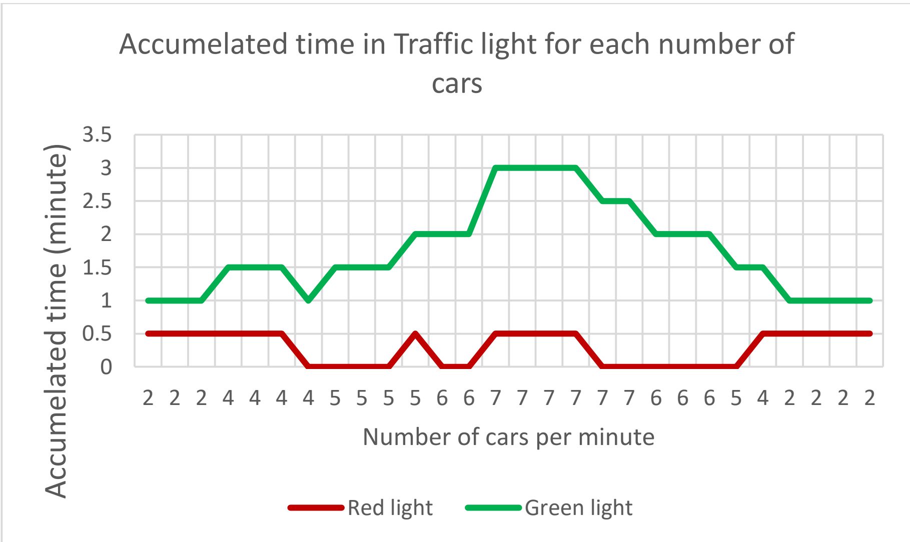



**Figure 27. Accumulated time per minute in 30 minutes in a static excel file.**

Case 3: Dynamic Data Based Exit Car Generation:

In this case, we read from the excel file having only one dynamic cell (which means once the data has been added in the cell using the Smart Parking Management System in real-time, the Arena reads the value in this cell per minute). So, this is close to the real-time distribution of car generation from parking gate.

Arena generates report automatically and records have been taken as follows:

**Table 11. Numbers of cars exiting per minute in 16 minutes in a dynamic excel file.**

| Number of Cars per minute | Accumulated Time for Red light | Accumulated Time for Green light | Average Accumulated time on Traffic Light for each Number of Cars |
|---|---|---|---|
| **7** | 0.5 | 2.5 | 1.5 |
| **5** | 0.5 | 1.5 | 1 |
| **6** | 0.5 | 2 | 1.25 |
| **5** | 0.5 | 1.5 | 1 |
| **7** | 0.5 | 2.5 | 1.5 |
| **6** | 0.5 | 2 | 1.25 |
| **5** | 0 | 1.5 | 0.75 |
| **5** | 0 | 1.5 | 0.75 |
| **7** | 0 | 2.5 | 1.25 |
| **6** | 0 | 2 | 1 |
| **7** | 0.5 | 2.5 | 1.5 |
| **4** | 0 | 1 | 0.5 |
| **7** | 0 | 2.5 | 1.25 |
| **6** | 0.5 | 2 | 1.25 |
| **5** | 0.5 | 1.5 | 1 |
| **5** | 0 | 1.5 | 0.75 |

The result of testing multiple cars exiting during 16 minutes:

- Total number of cars / minute = 93 cars
- Total Accumulated Time for Red light = 4.5 m
- Total Accumulated Time for Green light = 30.5 m
- Total Average time on Traffic Light = 17.5 m

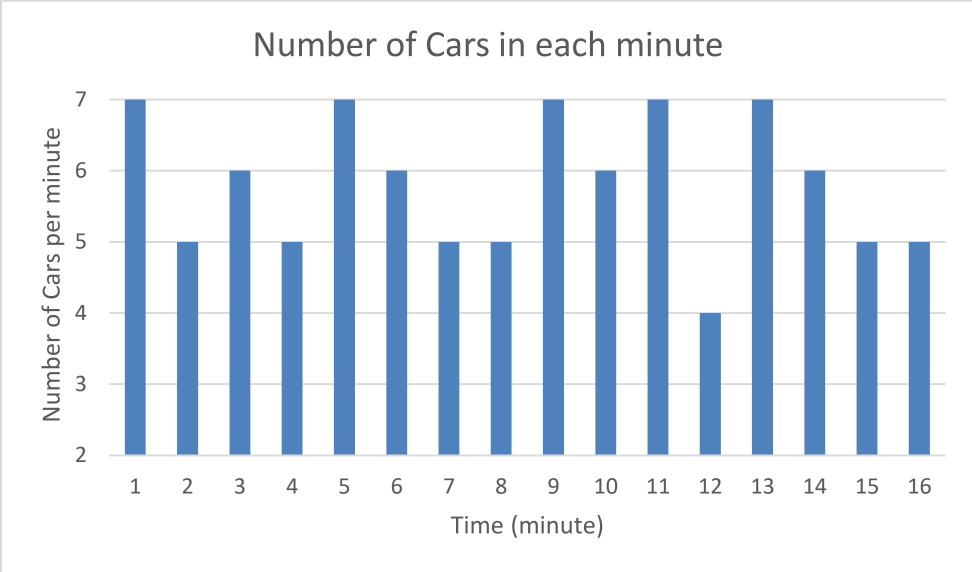



**Figure 28. Numbers of cars exiting per minute in 30 minutes in a static excel file.**

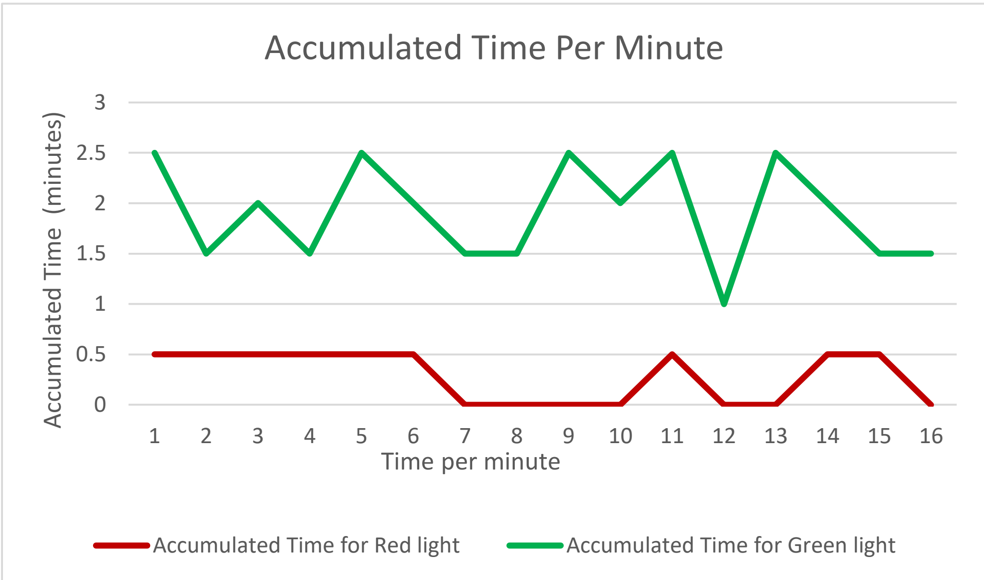



**Figure 29. Accumulated time per minute in 16 minutes in a dynamic excel file.**

The result shows that the total time for 93 cars in Green light is around 30.5 minutes, while in Red light takes about 4.5 minutes. The average accumulated time taken on a traffic light when it is both red and green is 17.5 which is a short range when having 93 cars leaving. This reflects the aim of the proposed solution in reducing the traffic congestion by extending the duration of green light while keeping the red light for short time only to ensure safe atmosphere as shown in Figure X. We can notice from the table that when we have 7 cars exiting, the green light duration stays for longer than when having 4, 5 or 6 vehicles.

?

### 6.2. Non-functional requirements testing:

#### 6.2.1. Performance – ***Not Met***

We have used the function System.nanoTime() to be more accurate about the time taken for the system to respond. It appears that the system took above 10-15 ~~minutes~~ seconds to respond more than 95% of the time. Which contradicts with what we have listed as a non-functional requirement.

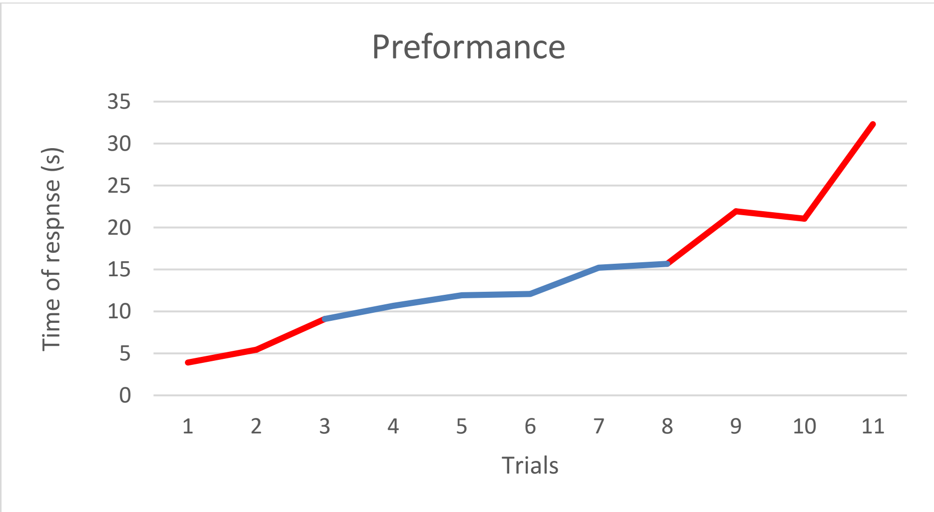



**Figure 30. Responses time of the system on different trails.**

<u>Ways to improve:</u>

The performance of the system can be improved through decreasing the complexity of some functions in order to decrease the response time.

should state some example/candidate functions

#### 6.2.2 Availability - ***Met***

We have tested the availability of the system under high load of applicants or data stored by registering 10, 20, 40 and 60 applicants. Although the response time increases with the increase of data, the system still handles the situation very well, which was required in the non-functional requirements.

### 6.2.3 Usability - ***Met***

We have tested usability of the system by giving it to users of different backgrounds in using technologies and get feedback on user-experience. Also, we have tested the understandability of the simulation and how close is it to real-life experience by explaining it to audience and getting feedback.

**Table 12. Usability Testing for Non-Functional Requirements.**

| Level of Knowledge in Modern Technology | No. Of users | User-experience |
|---|---|---|
| **Low** | 2 | Although there were some difficulties, the picture of how the real parking management system is going to be through our simulation was delivered well to the users. Users understood the main functionalities of the system. However, in using the system users experienced hard time dealing with it. |
| **Satisfactory** | 4 | Users understood the simulation and how it represent real life situations and find it easy to work with system. |

<u>Ways to improve:</u>

In implementing the system in real-life, users who are going to monitor the parking needs to be trained to use the system.

## 7. Expected Impact

The proposed Smart Parking project is expected to have the following impacts:

- **Reduces Parking Searching Time**:
  Parking Registration System will guarantee that spectator have a specific parking space on the day of the match which reduces the time searching for a parking space. Reducing parking searching time will make the spectator happier.

- **Reduces Traffic Congestion**:
  - At Arrival Time:
    Each spectator has a registered parking space in a specific parking zone and only authorized cars can enter a parking zone. Therefore, the impact of having cars congestions in different parking zones will be reduced which will make an individual more comfortable and less anxious.

  - At Exiting Time:
    The connection with traffic lights based on a given number of cars will prevent traffic congestion from happening and smooth vehicles flow outside the stadium providing a safe environment for drivers. Therefore, avoiding vehicles and people accidents.

- **Usage of new Technology:**
  The concept of Internet of Things is a brand-new topic that has been used in many different research projects all around the world. However, developing systems with such a powerful technology will help in changing people’s lifestyle of living and improve the society in many aspects. In addition, it is aligned with Qatar National Vision 2030 which inspire the Qatari society to develop economically, environmentally and most important socially. Moreover, the concept of Smart City has been implemented in many districts in different civilized cities. Qatar nowadays is implementing Smart City modern concept in innovative projects to empower businesses through new services.

# 8. Conclusion

According to the objectives we were hoping to establish through this project, we have highly achieved the standards by developing ways of communication and simulating computations between different parking nodes in order to control traffic. We have developed solutions to accomplish objectives of reducing the time of searching for parking spots and reducing traffic inside parking areas through developing ways of communication by using IoT and implementing the simulation of logic and computation that can be done to find the optimal parking spot for a spectator and thus, reduce the traffic congestion inside parking areas. Even more, we have simulated ways and developed solutions to achieve the objective of easing the flow of exit of cars and we have produced reports that show some statistics that proved that these solutions could truly reduce the traffic and waiting time of cars in smart traffic lights and these proves can benefit in representing the audience that this simulation can have real results in real-life. The strength of our implementation can be seen through the usage of Open Group IoT (O-MI/O-DF), which is the standard tool for IoT and the standardization in the smart parking solutions has adventurous over other smart city solutions in a way that it avoid the current problems of interoperability, and it is better integration of the smart parking system as it also more adaptable solution and can be implemented any where. Also, our system is an implementation of what really the system should do in different sub-systems, which gives the advantage of visualizing and understandability before implementing the real system and gives the advantage of changing of some aspects that may not be suitable for audience easily. It is also to our advantage that we have seen how really Smart Parking systems can work on real-life (Through our visit to Lussail City), and what functionalities might be added that are not already there to handle parking traffic more efficiently (i.e. locating parking spots according to seat numbers) or other ways that can enhance the implementation of the system. For instance, Lussail City Smart Parking system are not using a standard solution of IoT that we thought might cause future complication and thus, we switched to more standardized way of implementation in our simulation. However, there are some shortcomings in our project that can be improved to enhance the simulation. For instance, by using DATEX IoT, which is a standard that is developed for information exchange between traffic management nodes using IoT, we can achieve a better communication tool as it is more specifically related to traffic and parking and it has more attributes that are already implemented such as parking floors, parking services, number of parking spaces and many other attributes and therefore, not requiring effort on developing tree of IoT message exchanged.

# 9. Future work

- **Improving the implementation of the system:**
  - We could add more features to the real-time parking management system such as adding basements floors to increase the number of parking spaces within a zone so that the system would be more realistic since most of parking lots in Qatar are based on parking basements. Also, within each zone, we could add more services to be provided such as register parking spaces for police cars, internal security force “Lekhwiya” and ambulances to handle any emergency situations during big events.

  - Involve the implementation of Internet of Things more into the system to save more time in monitoring as IoT systems deliver data faster and more accurately.

- **Scaling the system:**
  In our proposed solution, we trained and tested the system with few data, however, we would like to make the system more scalable to handle larger number of spectators especially when we would like to introduce the proposed system to large stadiums in Qatar that handles over 5000 spectators in 2022 FIFA World Cup. Therefore, we would like to scale the data in both our registration system to handle larger number of data and in parking management system to allocate parking spaces at real-time when several spectators arrive. Moreover, scale the simulation of exit flow to visualize the connection with traffic lights with large number of vehicles exiting.

- **Extending the implementation using DATEX II:**
  Another way that would value our solution is extending it using DATEX II exchange standard. DATEX II is developed as a traffic data exchange mechanism. It’s beneficial to our system because it is specialized for road and traffic related events such as exchanging traffic status if it is heavy, congested or free flow. Also, it can be used in exchanging dynamic and status data about parking spaces such as occupancy information about a parking space, number of vacant and occupied parking spaces with a zone and exiting/arriving vehicle count. It is a very effective tool yet might need advanced training to be used.

Text is not clear

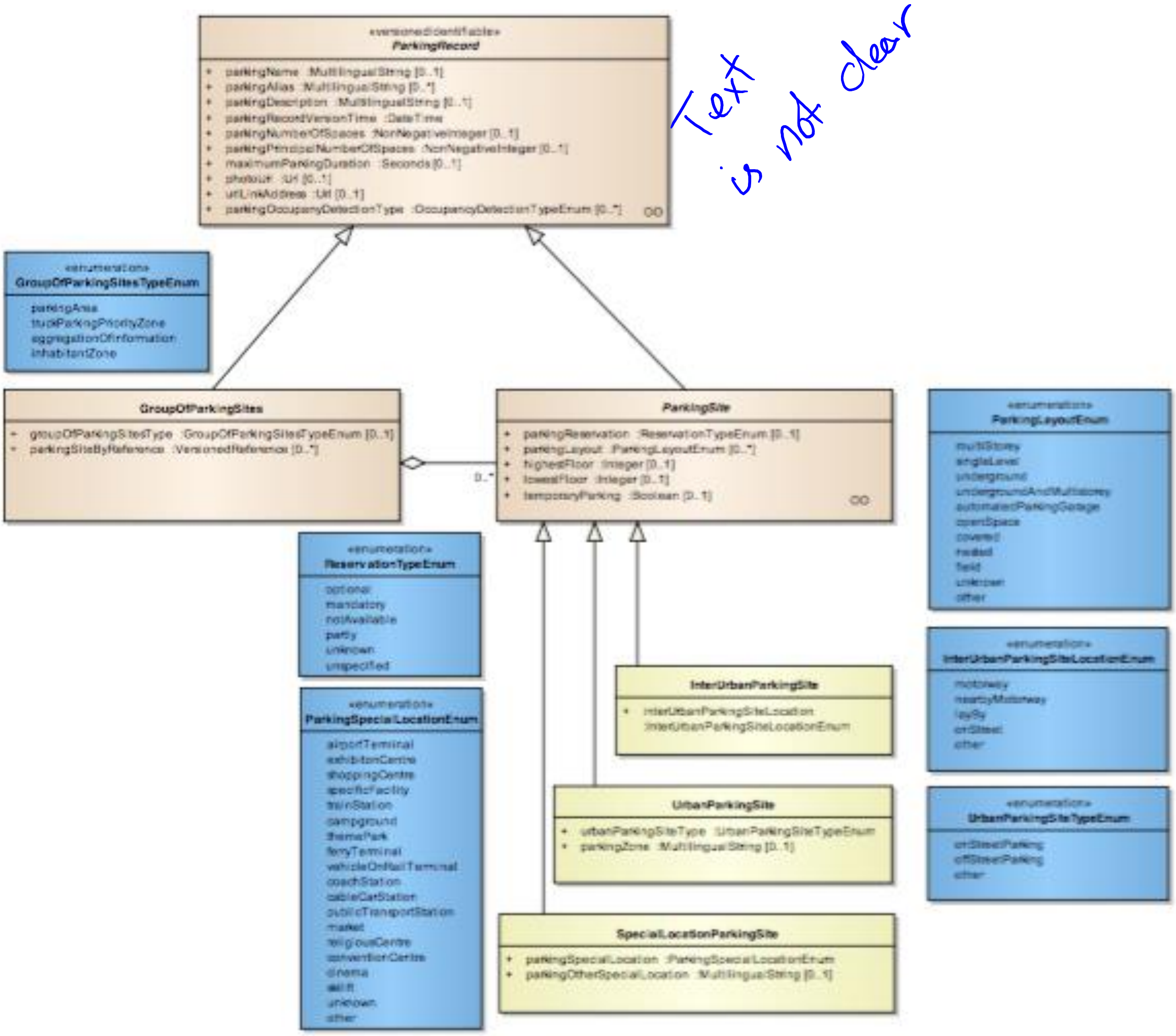



**Figure 31. ParkingRecord and ParkingSite UML class diagram using DATEX II.**

## 10. Student reflections

- **Ekram Gamal Al-Qilani:**

This project was a profitable and interesting experience to me. It demonstrated to me generally accepted methods to build up a simulation and convey to a real world. I have learned a lot of from this project, which helped me to improve my skills. We have been working at it for very nearly a year, and all the planning and implementing then testing were so perfect as should have been obvious the aftereffects of this hard work. It was essential to tune in, think, outline, then after that comes the implementation. The implementation part was the most unsafe and troublesome part with limited time. Along these lines, I have figured out how to compose simulation with code, and how connect with outside databases. In addition to that, getting so somewhere down in the testing like the organization graphs. I have enjoyed being a part of this important project and knowing the value that it can add to the Qatar 2022 makes me so proud.

- **Lameaa Rashid Alsulaiti:**

Throughout this project, I have learned many valuable lessons and developed myself in many different aspects. Mostly I have gained new technical skills as a sub-system of our project was developed using Internet of Things, which is a new technology nowadays and an advanced one to our level of study. It was interesting and valuable to gain knowledge on such new topics that was not covered during my study years in University, and I think it will benefit me on my future career. Also, I have come to learn more about simulation through this project, as I did not take a simulation course whatsoever. Other than technical skills, this project benefits me in the sense of enhancing my problem solving and researching skills as if was a self-learning course. We were encouraged to search and explore more about the field and build our knowledge, although we sought some help from our professor, manly we did most of the work of searching, self-learning and solving problems we face by ourselves. Even more, working on this project trained me on group working or team working. I gained some skills of leading the team and assigning tasks to my team members and myself as well as communicates and learns from each other through exchanging information. We as group learned the weakness and strength of each other and work a way to divide the work according to that, along with many other interpersonal skills that I learned from this project. Even more, we have faced many problems of short timing or not being able to assign a perfect time to meet with each other, however, we have over come these problems and learned how to work under pressure and meet deadlines of milestones. Furthermore, I learned the importance of planning and preparation and design before going forward for each step. I think that those lessons are valuable for my future work life as it emphasis some of the important skills needed anywhere for a good working performance.

- **Maryam Nasser Al-Naama:**

During the lifecycle of developing our proposed project, there were many lessons at each phase which taught me a lot. Our project is not one of the easiest, we faced multiple problems in understanding, implementing and testing it. At each milestone, I gained new skills in solving each problem faced and start improving my technical skills more. Although we worked with Internet of Things which was totally a new concept to me, I tried my best to fully understand it and implement it to our system with the help of our supervisor Dr. Abdelaziz Bouras and Mr. Ahmed Hefnawy from ictQatar who shared with us their pearl of knowledge and supported us to continue developing. The new technical skills and new concepts learnt, will efficiently help me in my future career in building big projects and be part of successful organizations. Also, I learned to always be skeptical in the work I do to ensure that it is qualified and would add a lot to the project. Planning and decision making were very highly recommended to be done at each milestone of developing our project to ensure that we are in the right track. We faced problems in not finishing milestones in their specified time, we overcome such problem with trying to always seek for help from others such as other instructors and TAs. We asked Dr. Samer to get help in installing Arena for simulation and Mr. Ahmed Hefnawy for installing Open Group standards for IoT. In any project, there is always different opinions and ideas from our supervisor or us as a team, therefore, adapting to any changes is highly needed to overcome obstacles and further develop a powerful project. In addition, working as a group, helped me in knowing how to gain interpersonal skills to cooperate with different personalities, leading others to do their assigned work and helping each other in case of any difficulties faced. As a group, we faced problems in finding a suitable timing to meet. However, a key lesson that I learnt from working in a team work is to be able to cope with people with different personalities and qualifications and always listen to different opinions and ideas to be able to continue till the end. Developing a successful project is not easy, you must fail in order to success because failing is the first attempt in learning.

# References


[1] *Match of Champions* [Digital image]. (2016). Retrieved December 21, 2016, from https://pbs.twimg.com/media/CzgQXBPXUAQY99z.jpg:large

[2] Salotagi, S., & Mundas, S. (2016). *User Innovation on Internet of Things. International Journal of Trend in Research and Development, 3(4). 473-477.*

[3] J. Banks, J. Carson, B. Nelson, D. Nicol (2010). Discrete-Event System Simulation. Upper Saddle River, NJ: Prentice Hall.

[4] The Government of the Hong Kong Special Administrative Region. (2015). *RESEARCH REPORT ON SMART CITY* (Rep.). Hong Kong.

[5] Bhardwaj, A. (2015). *Leveraging the Internet of Things and Analytics for Smart Energy Management*. TATA Consultancy Services, USA.

[6] Paraveen, V., & Seelam, S. M. (2013). *Smart Metering for Smart Electricity Consumption* (Rep.). Blekinge Institute of Technology, Sweden.

[7] Takahiro, K., Naofumi, S., Masatake, T., Shin, T., Sumio, M., & Stoianov, I. (2012). NEC Technial Journal. *Smart Water Management Technology with Intelligent Sensing and ICT for the Integrated Water Systems, 9*(1), 103-106.

[8] Parbu, V., & P. (2016, February). *IoT Based Waste Management for Smart City*. *4*(2), 1267-1274. doi:10.15680/IJIRCCE.2016. 0402029

[9] K. Natarajan., B. Prasath., & P. Kokila., (2016).*The Energy Efficiency Clustering in Wireless Sensor Network, 6(3), 37-42.*

[10] Klack, L., Möllering, C., Ziefle, M., & Schmitz-Rode, T. (n.d.). *Future Care Floor: A sensitive floor for movement monitoring and fall detection in home environments* (Rep.). RWTH Aachen University, Aachen, Germany.

[11] Osmani, A., Gawade, A., Nikam, M., & Wavare, S. (2016). *Research paper on Smart City Parking System. 2(3), 2998-3000*

[12] Wang, H., & He, W. (2011, July). *A Reservation-based Smart Parking System*. University of Nebraska - Lincoln. NE, USA.

[13] Basu, A. (2014). *SMART PARKING* (Rep.). Retrieved December 10, 2016, from http://www.happiestminds.com/whitepapers/smart-parking.pdf

# Appendix A – Project Plan

## A.1. Project milestones

To guarantee a successful deliverable of an efficient and effective proposed system, major milestones are planned throughout the lifecycle of developing the system. Each milestone accomplishes detailed deliverable tasks within a specific time duration. An iterative and incremental approach is achieved:

**1. Planning:**

In this stage, a meeting with the stakeholders is required to have a brief overview of how the system should be implemented by gathering all the important requirements. A visit to Lusail City is scheduled to validate the proposed system on an example of Smart Parking of Lusail City and possibly enhance the requirements.

**2. Requirements Analysis:**

After gathering all the needed requirements, a discussion is mandatory with the stakeholders to enhance the analysis of functional and non-functional requirements of the system. A use case diagram and use case specifications are developed to represent the main functions the system will achieve.

**3. System Design:**

Developing the architecture of the system depending on the communication between different components, modules and interfaces of the system. Specifying the design pattern used. The architecture of the system should be the best design suiting the functionality of the system.

**4. Implementation:**

In this stage, an execution of the previously developed design should be achieved. Implementing the computational login of the system using certain software based on the architecture of the system.

**5. Testing and Evaluation:**

Testing the code implemented using several using standard software engineering testing methods such as Unit testing, Data set testing and User-acceptance testing. The aim of this stage is to check if the proposed system meets the clients' requirements.

**6. Deployment and Maintenance:**

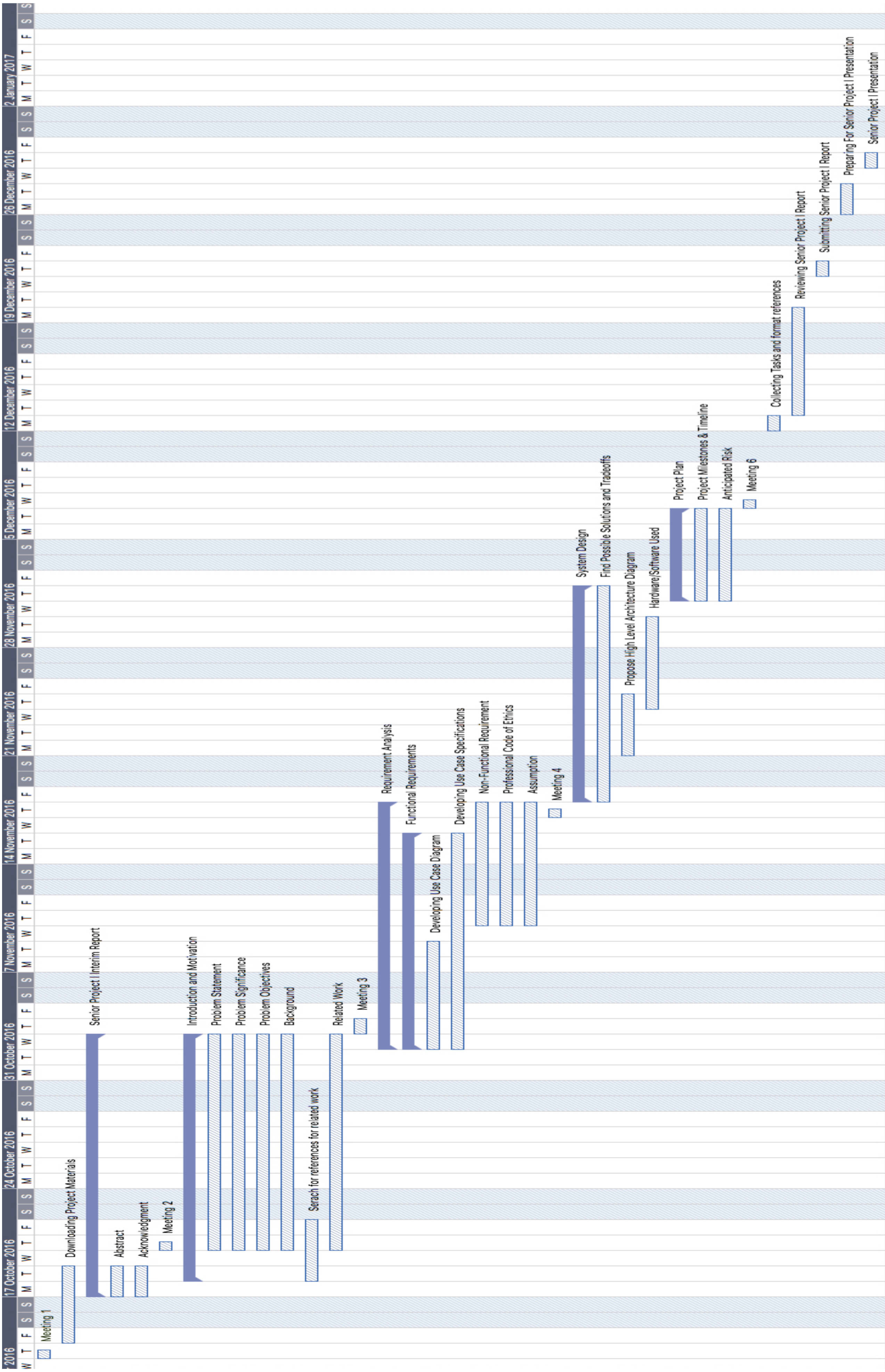
After evaluating the system against client's requirements in a customer environment and an implementation errors are found, the system will undergo a maintenance stage releasing a better version filling the needs of the user.

### A.2. Project timeline

- **Senior Project I – from Oct to Dec, 2016**

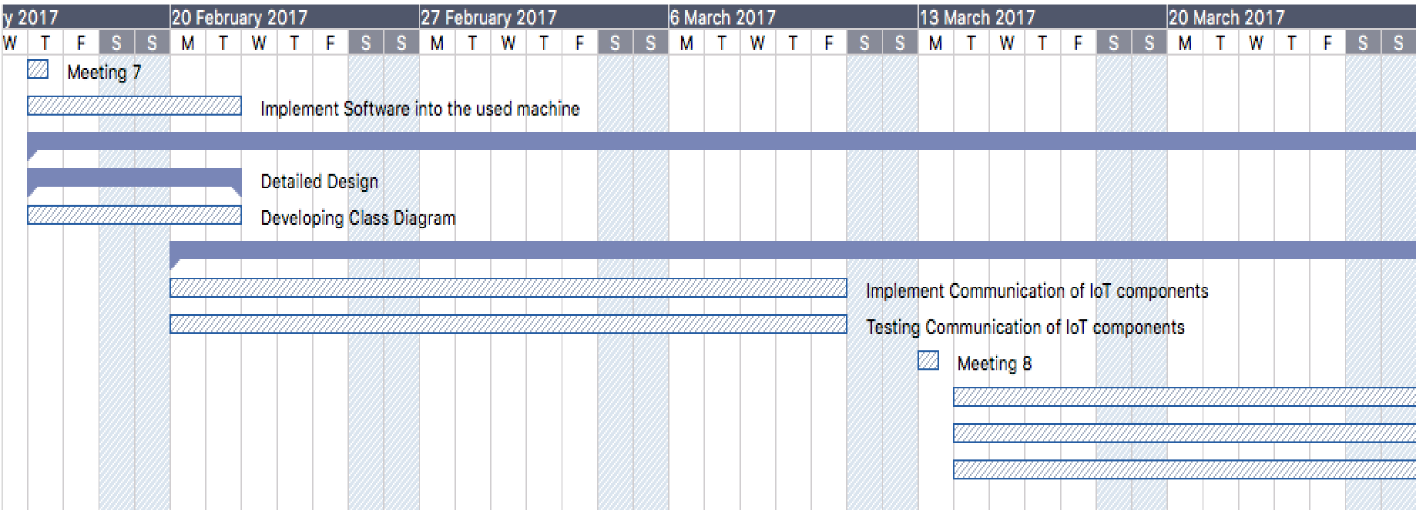
The tasks are spread between group members. Each will be handling specific tasks throughout the two semesters. On other hand, some stages of developing the system which requires more effort such as implementation and testing phases, it will be a group task where all the group members will work together to deliver a successful system.

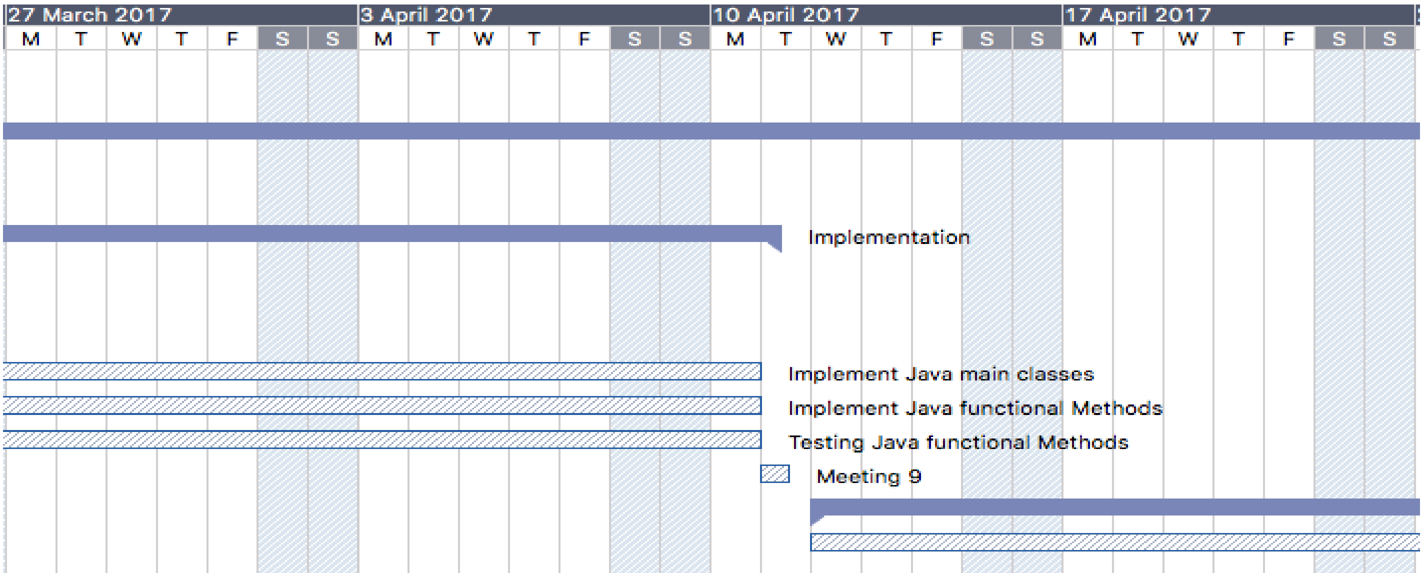
| # | Task Name | Duration | Start | End |
|---|---|---|---|---|
| 1 | Meeting 1 | 2 hrs | 13 Oct 2016 | 13 Oct 2016 |
| 2 | Downloading Project Materials | 3 days | 14 Oct 2016 | 18 Oct 2016 |
| 3 | **Senior Project I Interim Report** | 13 days | 17 Oct 2016 | 2 Nov 2016 |
| 4 | Abstract | 2 days | 17 Oct 2016 | 18 Oct 2016 |
| 5 | Acknowledgment | 2 days | 17 Oct 2016 | 18 Oct 2016 |
| 6 | Meeting 2 | 2 hrs | 20 Oct 2016 | 20 Oct 2016 |
| 7 | ▼**Introduction and Motivation** | 12 days | 18 Oct 2016 | 2 Nov 2016 |
| 8 | Problem Statement | 10 days | 20 Oct 2016 | 2 Nov 2016 |
| 9 | Problem Significance | 10 days | 20 Oct 2016 | 2 Nov 2016 |
| 10 | Problem Objectives | 10 days | 20 Oct 2016 | 2 Nov 2016 |
| 11 | Background | 10 days | 20 Oct 2016 | 2 Nov 2016 |
| 12 | Serach for references for related work | 4 days | 18 Oct 2016 | 21 Oct 2016 |
| 13 | Related Work | 10 days | 20 Oct 2016 | 2 Nov 2016 |
| 14 | Meeting 3 | 1 day? | 3 Nov 2016 | 3 Nov 2016 |
| 15 | **Requirement Analysis** | 12 days | 2 Nov 2016 | 17 Nov 2016 |
| 16 | ▼**Functional Requirements** | 10 days | 2 Nov 2016 | 15 Nov 2016 |
| 17 | Developing Use Case Diagram | 5 days | 2 Nov 2016 | 8 Nov 2016 |
| 18 | Developing Use Case Specifications | 10 days | 2 Nov 2016 | 15 Nov 2016 |
| 19 | Non-Functional Requirement | 6 days | 10 Nov 2016 | 17 Nov 2016 |
| 20 | Professional Code of Ethics | 6 days | 10 Nov 2016 | 17 Nov 2016 |
| 21 | Assumption | 6 days | 10 Nov 2016 | 17 Nov 2016 |
| 22 | Meeting 4 | 2 hrs | 17 Nov 2016 | 17 Nov 2016 |
| 23 | **System Design** | 10 days | 18 Nov 2016 | 1 Dec 2016 |
| 24 | Find Possible Solutions and Tradeoffs | 10 days | 18 Nov 2016 | 1 Dec 2016 |
| 25 | Propose High Level Architecture Diagram | 4 days | 21 Nov 2016 | 24 Nov 2016 |
| 26 | Hardware/Software Used | 4 days | 24 Nov 2016 | 29 Nov 2016 |
| 27 | **Project Plan** | 4 days | 1 Dec 2016 | 6 Dec 2016 |
| 28 | Project Milestones & Timeline | 4 days | 1 Dec 2016 | 6 Dec 2016 |
| 29 | Anticipated Risk | 4 days | 1 Dec 2016 | 6 Dec 2016 |
| 30 | Meeting 6 | 2 hrs | 7 Dec 2016 | 7 Dec 2016 |
| 31 | Collecting Tasks and format references | 1 day? | 12 Dec 2016 | 12 Dec 2016 |
| 32 | Reviewing Senior Project I Report | 5 days | 13 Dec 2016 | 19 Dec 2016 |
| 33 | Submitting Senior Project I Report | 1 day? | 22 Dec 2016 | 22 Dec 2016 |
| 34 | Preparing For Senior Project I Presentation | 2 days | 26 Dec 2016 | 27 Dec 2016 |
| 35 | Senior Project I Presentation | 1 day? | 29 Dec 2016 | 29 Dec 2016 |

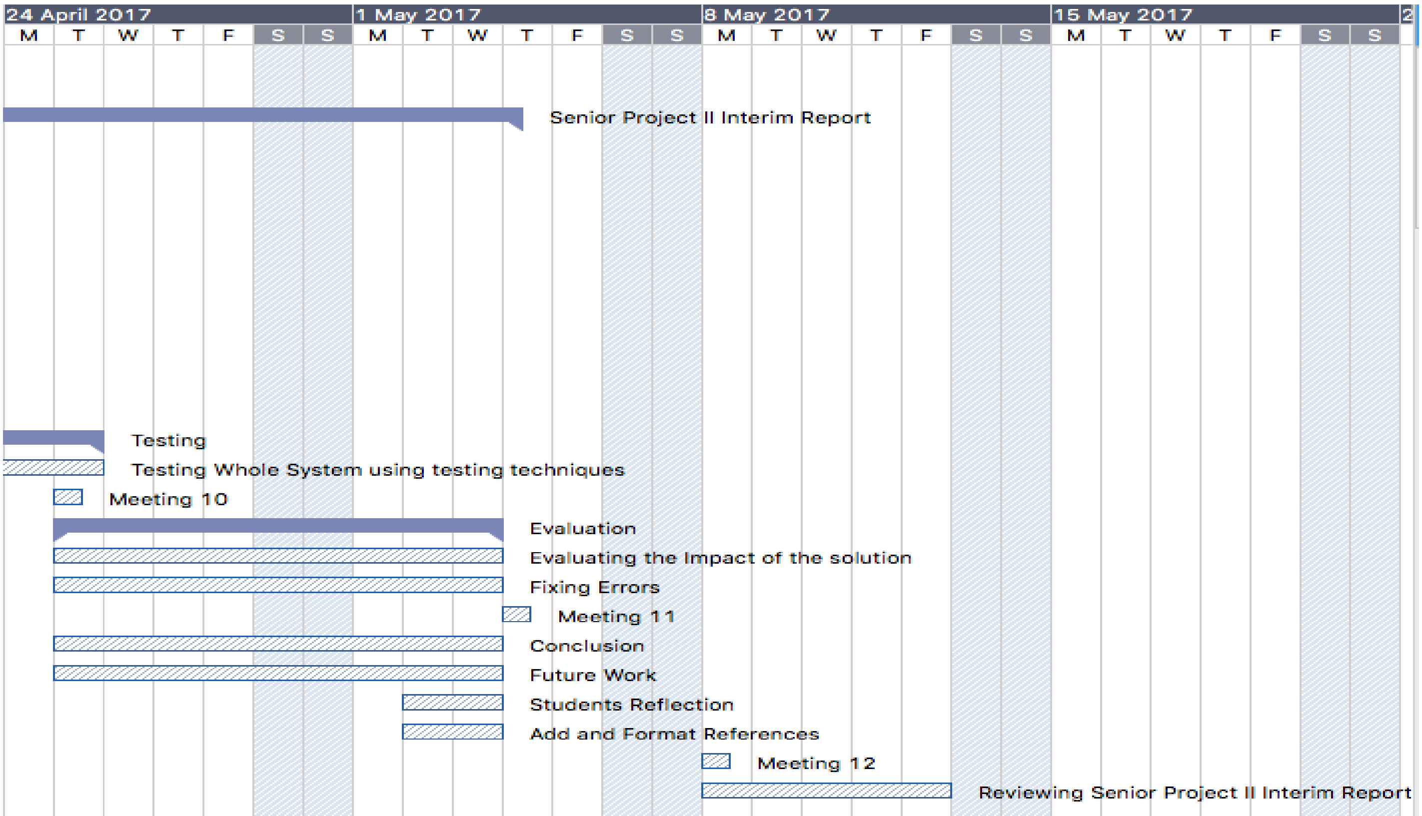

2016
17 October 2016
24 October 2016
31 October 2016
7 November 2016
14 November 2016
21 November 2016
28 November 2016
5 December 2016
12 December 2016
19 December 2016
26 December 2016
2 January 2017
Meeting 1
Downloading Project Materials
Senior Project I Interim Report
Abstract
Acknowledgment
Meeting 2
Introduction and Motivation
Problem Statement
Problem Significance
Problem Objectives
Background
Serach for references for related work
Related Work
Meeting 3
Requirement Analysis
Functional Requirements
Developing Use Case Diagram
Developing Use Case Specifications
Non-Functional Requirement
Professional Code of Ethics
Assumption
Meeting 4
System Design
Find Possible Solutions and Tradeoffs
Propose High Level Architecture Diagram
Hardware/Software Used
Project Plan
Project Milestones & Timeline
Anticipated Risk
Meeting 6
Collecting Tasks and format references
Reviewing Senior Project I Report
Submitting Senior Project I Report
Preparing For Senior Project I Presentation
Senior Project I Presentation

- **Senior Project II – from Feb to Jun, 2017**

| # | Task Name | Duration | Start | End |
|---|---|---|---|---|
| 1 | Meeting 7 | 2 hrs | 16 Feb 2017 | 16 Feb 2017 |
| 2 | Implement Software into the used machine | 4 days | 16 Feb 2017 | 21 Feb 2017 |
| 3 | **Senior Project II Interim Report** | 55.25 days | 16 Feb 2017 | 4 May 2017 |
| 4 | ▼**Detailed Design** | 4 days | 16 Feb 2017 | 21 Feb 2017 |
| 5 | Developing Class Diagram | 4 days | 16 Feb 2017 | 21 Feb 2017 |
| 6 | ▼**Implementation** | 36.25 days | 20 Feb 2017 | 11 Apr 2017 |
| 7 | Implement Communication of IoT components | 15 days | 20 Feb 2017 | 10 Mar 2017 |
| 8 | Testing Communication of IoT components | 15 days | 20 Feb 2017 | 10 Mar 2017 |
| 9 | Meeting 8 | 2 hrs | 13 Mar 2017 | 13 Mar 2017 |
| 10 | Implement Java main classes | 20 days | 14 Mar 2017 | 10 Apr 2017 |
| 11 | Implement Java functional Methods | 20 days | 14 Mar 2017 | 10 Apr 2017 |
| 12 | Testing Java functional Methods | 20 days | 14 Mar 2017 | 10 Apr 2017 |
| 13 | Meeting 9 | 2 hrs | 11 Apr 2017 | 11 Apr 2017 |
| 14 | ▼**Testing** | 10 days | 12 Apr 2017 | 25 Apr 2017 |
| 15 | Testing Whole System using testing techniques | 10 days | 12 Apr 2017 | 25 Apr 2017 |
| 16 | Meeting 10 | 2 hrs | 25 Apr 2017 | 25 Apr 2017 |
| 17 | ▼**Evaluation** | 7 days | 25 Apr 2017 | 3 May 2017 |
| 18 | Evaluating the Impact of the solution | 7 days | 25 Apr 2017 | 3 May 2017 |
| 19 | Fixing Errors | 7 days | 25 Apr 2017 | 3 May 2017 |
| 20 | Meeting 11 | 2 hrs | 4 May 2017 | 4 May 2017 |
| 21 | Conclusion | 7 days | 25 Apr 2017 | 3 May 2017 |
| 22 | Future Work | 7 days | 25 Apr 2017 | 3 May 2017 |
| 23 | Students Reflection | 2 days | 2 May 2017 | 3 May 2017 |
| 24 | Add and Format References | 2 days | 2 May 2017 | 3 May 2017 |
| 25 | Meeting 12 | 2 hrs | 8 May 2017 | 8 May 2017 |
| 26 | Reviewing Senior Project II Interim Report | 5 days | 8 May 2017 | 12 May 2017 |
| 27 | Submitting Senior Project II Interim Report | 1 day? | N/A | |
| 28 | Preparing for Senior Project II Final Presentation | 5 days | N/A | |
| 29 | Meeting 13 | 2 hrs | N/A | |
| 30 | Senior Project II Final Presentation | 1 day? | N/A | |


y 2017
20 February 2017
27 February 2017
6 March 2017
13 March 2017
20 March 2017
Meeting 7
Implement Software into the used machine
Detailed Design
Developing Class Diagram
Implement Communication of IoT components
Testing Communication of IoT components
Meeting 8


27 March 2017
3 April 2017
10 April 2017
17 April 2017
Implementation
Implement Java main classes
Implement Java functional Methods
Testing Java functional Methods
Meeting 9


24 April 2017
1 May 2017
8 May 2017
15 May 2017
Senior Project II Interim Report
Testing
Testing Whole System using testing techniques
Meeting 10
Evaluation
Evaluating the Impact of the solution
Fixing Errors
Meeting 11
Conclusion
Future Work
Students Reflection
Add and Format References
Meeting 12
Reviewing Senior Project II Interim Report

### A.3. Anticipated risks

**Table 13. Anticipated Risks**

| Risks | Approaches to minimize them |
|---|---|
| **Lack of experience regarding principles of IoT that may lead to delays in the project.** | Study the new concept of IoT and get familiar with O-DF and O-MI standards before the time of implementation and try to seek help and find alternative solutions to overcome complex tasks. |
| **A semester may not be sufficient for implementing and testing each unit, which may lead to delay in finishing the whole project.** | Define each process and the time it needs and when to start and define a deadline to finish in a project timeline. Then making sure the whole project will finish before the deadline with consideration of possible delays |
| **Other project non-related work may keep members of the project busy from giving needed effort for the project. Also, one member may not contribute as fair as the others in the project.** | Make sure to specify some tasks to each of the project members and agree on a deadline to accomplish tasks. |
| **Poor estimation of the time a process need to be accomplished or poorly design some parts of the system that makes it harder to implement and thus lead to delays in the project.** | Take into consideration a delay time for each task so that we would manage to complete the task even if the planning was poorly done for this specific part. |