

# **Department of Electrical Engineering PARKING LOT SYSTEM**

**Manuel Maia Lopes**

**213127652**

**Lecturer:**

V. Moyo

Room 4.18, Engineering Building

0214603096

[moyoV@cput.ac.za](mailto:moyoV@cput.ac.za)

January 2016

# Contents

Table of Figures .....	III
List of Tables .....	IV
Glossary .....	1
Introduction .....	3
1. Projects Description .....	4
1.1 CPUT GradeBook .....	4
1.2 PLS (Parking Lot System) .....	4
1.3 IAS (Intelligent Alarm System) .....	5
2 Motivation .....	6
3 Overview .....	7
3.1 Design of the Database .....	8
4 Gant Chart .....	9
5 User Requirement .....	10
5.1 Job Description .....	10
5.2 System Requirements .....	10
5.3 Limits .....	11
6 Planning and Design .....	11
6.1 User Interface .....	12
6.1.1 Server UI .....	12
6.1.2 Client UI .....	12
6.2 UML Use Case Diagram .....	15
6.3 Class Diagram .....	17
6.3.1 High Level Class Diagram .....	17
6.3.2 Detailed Class Diagram for the Client .....	18
6.3.3 Detailed Class Diagram for the Server .....	19
6.4 Activity Diagram .....	20
6.5 Sequence Diagram .....	21
7 Problem Design .....	22
7.1 PLS Scenarios .....	22

7.2	Administrator Scenarios.....	22
7.2.1	Adding New Record.....	23
7.2.2	Deleting a record.....	25
7.3	Security Scenario.....	27
7.3.1	Monitoring the PLS.....	27
7.4	Driver Scenario.....	29
7.4.1	Getting the Ticket.....	29
7.4.2	Pay Ticket .....	31
8	Test Reports .....	33
8.1	White-Box Testing.....	33
8.2	Black-Box Testing .....	35
8.3	Module Testing .....	37
9	Evaluation Report .....	41
9.1	Client Perspective .....	41
9.2	Programmer Perspective .....	41
9.3	Features .....	41
9.4	Problem Areas.....	41
9.5	Future Improvements .....	42
	Bibliography .....	43
	Appendix A: User Manual .....	45
10	Getting Started.....	45
10.1	Using NetBeans application .....	48
10.2	Using the Application.....	49
10.3	More Information .....	54
11	Conclusion.....	55

## Table of Figures

Figure 1 Server Form .....	12
Figure 2 Login Screen.....	12
Figure 3 Sign up Form .....	13
Figure 4 Administrator Form .....	13
Figure 5 Simulation Form (Retrieval and Payment of Ticket) .....	14
Figure 6 Monitor PLS Form .....	14
Figure 7 ArduinoConnect Form.....	15
Figure 8 Overall Use Case Diagram .....	16
Figure 9 Overall class relationship diagram .....	17
Figure 10 Classes that will interact with the user .....	18
Figure 11 Detailed Class Diagram of the other classes.....	19
Figure 12 Overall Detailed Class Diagram of the Server Side .....	19
Figure 13 Overall Activity Diagram .....	20
Figure 14 Sequence Diagram for Login.....	21
Figure 15 Administrator adding a new record .....	23
Figure 16 adding a new record Use case Scenario.....	23
Figure 17 Use case for the saving button .....	24
Figure 18 User interface for Deleting a record .....	25
Figure 19 Use case for Deleting a record.....	25
Figure 20 Activity Diagram for deleting a record.....	26
Figure 21 User Interface for Monitoring PLS.....	27
Figure 22 Use case Diagram for Monitoring the parking system .....	28
Figure 23: Acativity Diagram when monitoring the PLS .....	28
Figure 24 UI of the simulation when getting a ticket.....	29
Figure 25 Use case Diagram for when getting the ticket .....	29
Figure 26 Activity Diagram for when getting a ticket .....	30
Figure 27 UI for when paying the ticket.....	31
Figure 28 Use case diagram for paying ticket.....	31
Figure 29 Activity Diagram for when paying the ticket .....	32

## List of Tables

Table 1 SIGN_UP table .....	8
Table 2 LOGIN TABLE .....	8
Table 3 REPORT Table .....	8
Table 4 PLS TABLE.....	8
Table 5 Admin Table .....	8

## Glossary

**Hardware:** is the collection physical element that constitutes a computer system. [1]

**Software:** is any set of machine-readable instructions that directs a computer's processor to perform a specific operation. [2]

**Grade Book:** is a teacher's online record of their student's lessons, assignments, progress and Grades. [3]

**Embedded System:** Combination of a computer Hardware and Software, either fixed in capability or programmable and is specifically designed for a particular function. [4]

**Blackboard:** is a virtual learning environment and course management system. [5]

**Sensors:** is a transducer whose purpose is to sense or detect some characteristics of its environments. [6]

**Transducer:** is a device that converts one form of energy to another. [7]

**Frequency:** Are a number of occurrences of a repeating event per unit time. [8]

**Radio Signals or Radio Waves:** Is a type of electromagnetic radiation, as are microwaves, infrared radiation, X-Rays and gamma rays. [9]

**Hackers:** is someone who seeks and exploits weaknesses in a computer system or a computer network [10]

**Database:** are specialized structures that allow a computer based system to store, manage and retrieve data very quickly. [11]

**Client/Server:** is a program relationship in which one program (the client) requests a service or resource from another program (the server). [12]

**Arduino:** is an open source computer hardware and software company, project and user community that designs and manufactures microcontroller-based kits for building digital devices and interactive objects that can sense and control the physical world. [13]

## Introduction

Technology keeps improving in a high rate. Every day, people sit together trying to think what to come up next. Some improve what already exist, following the quote “Don’t reinvent the wheel, just realign it” by Anthony J. D’Angelo. Others try to come up with something innovative. Just like them, we were given the opportunity to come up with something new or improve something that already exist, after some years of learning new things.

Design Project 3 is a subject in where we are supposed to create our own project and present it. While thinking about a suited project, I got a few ideas and had to make a choice of the project that I would present. The ideas are:

- **CPUT Grade Book:** This is an application where the student would be able, not only to register online, but also will be able to share information between the students themselves and the lectures. Getting an update of everything that would happen in his/her department during the year in course.
- **PSL (Parking Lot System):** This is an application that would be able to manage the parking lot of a respective area, as soon as one presses the button, he will get a ticket with an available parking space, and if the parking is full a feedback will be sent.
- **Alarm System:** This will be an embedded system, where, whenever a person enters in the house, the alarm will go off, and as soon a as the alarm goes off, a message will be sent to the owners phone, improving like this the security of the house.



## **1. Projects Description**

### **1.1 CPUT GradeBook**

The idea of creating a CPUT GradeBook came from the blackboard, since it is being used for sharing information between lecturer and student. However, just like all software, there are pros and cons on the blackboard from my perspective and this are:

- Student cannot Register himself from it
- Sometimes fails to login
- Professors often do not post Grades [14]
- Quite Complex to use (need special training) [14]
- Lecturers have to set it up before using it. [14]

The Pros of the CPUT GradeBook compared to the current Blackboard are:

- Fast Login
- No need to go into the CPUT website in order to login
- Able to Register Online
- No need to set it up before using it
- Easier to use

The disadvantages of the CPUT GradeBook are:

- Can't be used to do exams or tests
- If the Server crashes, there will be no use for the application
- Can only be used with the internet or else the user won't be able to sign in

### **1.2 PLS (Parking Lot System)**

Not knowing where to park your car even though you arrived at the shopping mall, Bank or any other place can be very frustrating, therefore, the PLS would be of great help.

In one's point of view, The PLS would be a very vague idea, however, thinking about an intelligent parking lot would be very interesting and would increase the time consumption whenever looking for a parking space.

These are the Advantages of the PLS:

- Reduce the time consumption when looking for a parking space
- Less Stressful
- Inform the user of the parking availability

In case the parking lot is full, a feedback of a full parking System will be seen. However, a certain amount of time will be given to that ticket holder to enter and exit the parking lot system, or else a fine will be given to the holder that would exceed a fair amount, avoiding like this people entering the parking lot for free.

The disadvantages of the PLS are:

- If one of the sensors fail, there will some unreliable data [15]
- It can have an interference with other sensors [15]
- These sensors are only suitable for indoors usage [15]

### **1.3 IAS (Intelligent Alarm System)**

Getting your house robbed got safer with the creation of an alarm system which goes off as soon as the person goes in, since there are sensors on the alarm which detects the activity and sends whatever information gathered to the alarm system control panel.

However, there are some disadvantages when using a simple Alarm System:

- If forgotten to set the alarm on, and decided to go out, there will be a chance of leaving the house, or company unsecured.
- If forgotten to set it on, and on the way to a shop or any other place, the user will have to return at his house in order to set it on.

- If not home, and the house gets robbed, there will be no way of being informed that someone just entered the house.

The difference between a normal Alarm System and the IAS is that the IAS will cover for most of a normal Alarm System disadvantages:

- Set the alarm on and off via an application on his/her phone or computer
- Receive a message from any place in the world of what is happening in his/her own house

The IAS, is an Alarm System which will be able to do everything that a normal Alarm can do with some improvements as it was described above, improving like this the security of the his/her house.

The disadvantages of the IAS are:

- It can be very costly [16]
- It can have some interferences from devices using the same frequency [16]
- The battery of the Control Panel can become weak or die and cause the uselessness of the components [16]
- It is more vulnerable for hackers due to the radio signals [16]

Although there are some disadvantages on this system, the evolution of the technology makes it possible to solve many of these issues such as the battery problem that nowadays, it can last for the all year, and some installation process can help to avoid radio interferences, even reducing the probability of a burglar to access and disarm the control panel. [16]

## **2 Motivation**

I have decided to take the 2<sup>nd</sup> project or PLS (Parking Lot System), because in terms of difficulty stands in a level which will require lots of research and time which is quite challenging. However, the difficulty wasn't the only thing that actually inspired me, the fact that I'll be using hardware and software at the same time, in other words, knowledge that I have been acquiring

from the last 2 years to make 1 project only looked better than applying knowledge that had acquired in 1 year.

Following the requirement for the project, the chosen project will need knowledge of the following:

- Database
- Programming in Java and Arduino
- Networks
- Electronics
- Digits

The Interface was requested to be built in java, therefore, the Control Panel and some simulation parts will be designed and presented in Java. The authorized personnel to use the application will be stored in a Database as well as the time in and time out of the parking space being occupied, which we will be using MySQL in order to store the information needed.

The Network part will come in when the user that will take control of the Control Panel will sign in, since there will be only one database, and there must be one that has full control over it, the user will have to make a remote connection, in other words a Client/Server Connection.

### **3 Overview**

The project to be built is the PLS (Parking Lot System), where the System will be a guide for the drivers in order to make it easier for them to find the parking space. The user will get the ticket and with the ticket a number of this/her parking place. As soon as the ticket is taken, that ticket will come out again as long as there is not someone using that place. A sensor will be used to take the information of whether the space is being used or not

In order to do the project, I divided the time in 5 ways:

- Searching for more Knowledge especially about sensors
- Buy the Components
- Documentation

- Do the Project

The way that the time was divided will be shown in more details on the Gant Chart.

For the Hardware part, it is thought to use Arduino and the IR sensors, and a model for the hardware representation. A 3x3x3 led cube will be added just for show.

### 3.1 Design of the Database

Four Tables will be used in this project:

staff_ID	Username	Password	Conf_Pass

Table 1 SIGN\_UP table

Staff_ID	Username	Password

Table 2 LOGIN TABLE

ID	STAFF_ID	PARK_SPACE	DATE_TO_CONTROL	TIME_IN	TIME_OUT

Table 3 REPORT Table

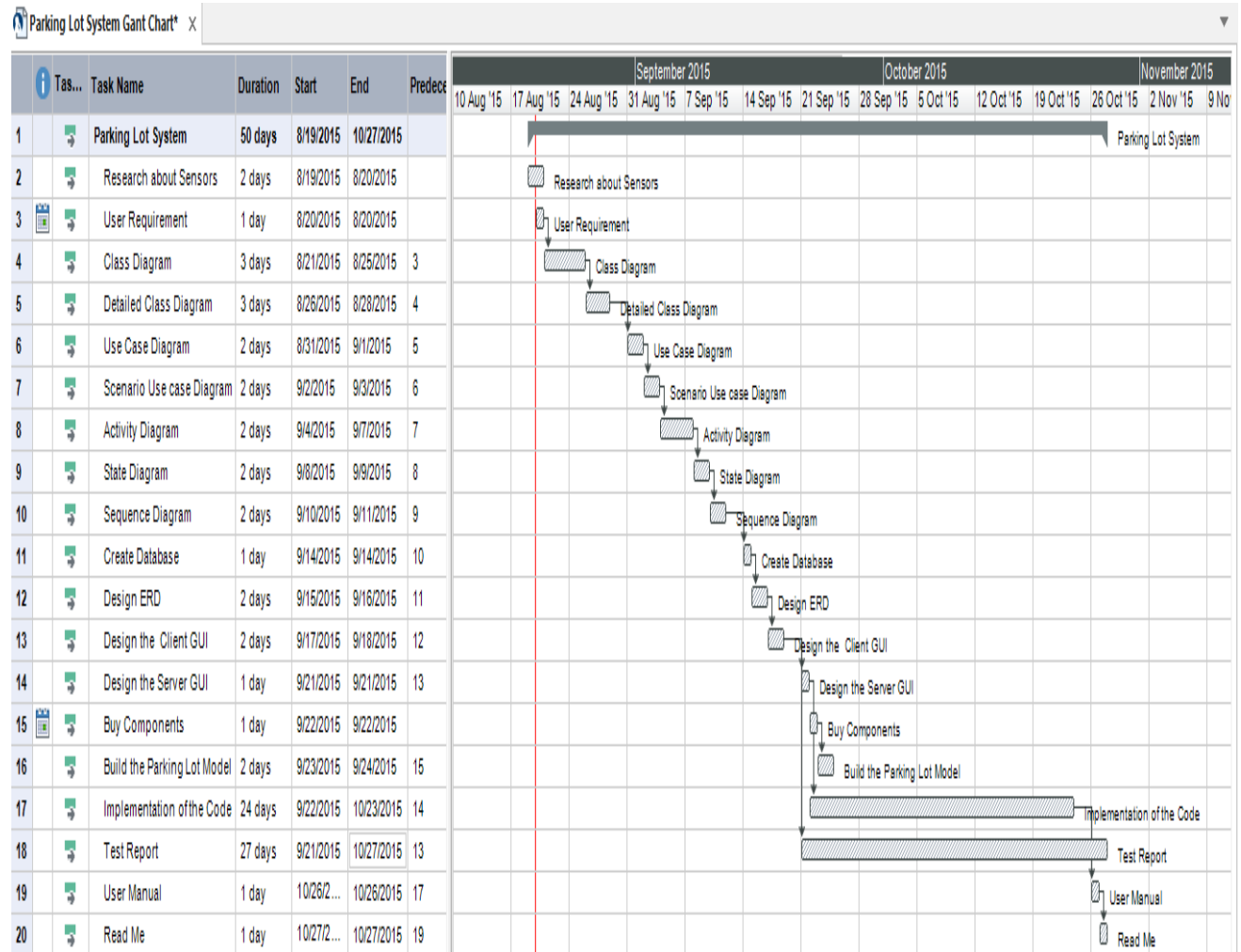
PARKING_ID	PARK_SPACE	PARK_AVAILABILITY

Table 4 PLS TABLE

STAFF_ID	First_Name	LAST_NAME	GE	DATE_OF_BIRTH	ID_NUMBER	PHONE_NUMBER	ADDRESS	R	STREET	CITY	CITY	POSITION	DATE_HIRED

Table 5 Admin Table

## 4 Gant Chart



## **5 User Requirement**

### **5.1 Job Description**

It is required for the developer to create a parking lot simulation. The data used for the system must come from the sensors and displayed on a Java application. The application must be able to perform the following:

- Register the Control Panel Officer (Security)
- Simulate a ticket withdrawal
- Get the Data acquired from the sensors after the car was parked
- Update the availability of the parking spaces
- Simulate the Payment of the ticket

A simple model for the parking lot must be created in order to store a minimum of 6 cars. Before parking the car, a ticket must be given to the drivers. After the driver park the car, using sensors, the application will have to put the parking space as unavailable and at the same time, and as soon as the driver goes out of the sensors range, in the Control Panel will have to say that the space is available. However, before the driver goes out, a simulation showing the amount of money the driver is paying in order to go out will have to be shown.

In case the parking lot is full, a feedback from the ticket withdrawal machine will have to be seen saying “The Parking Lot is full”.

The information must be accessed, edited or updated directly from the database using a Server/Client connection.

### **5.2 System Requirements**

The System will run on a standard personal computer using Windows 7 or later as the operation system and it has to be able to run a java application.

### **5.3 Limits**

- It will only make a simulation for 6 parking lot spaces.
- A java application will be used to simulate the feedback and the ticket machines.
- It will not produce a report for the all application including the Administration part

## **6 Planning and Design**

Before starting any work, a good planning and analysis of how the system will work will make the work much easier whenever we get to the design and implementation phase. This is the phase where we can see how the application will work theoretically, in other words, how the system will work behind the scenes of what the user will see.

For a better understanding of the System, graphical structure where used to explain in details how the system will function, to show the bigger picture of the system [1], not only in general, but also in some specific cases like how the communication between server and client is executed, how some information come from the Arduino to the Java application.



## 6.1 User Interface

### 6.1.1 Server UI

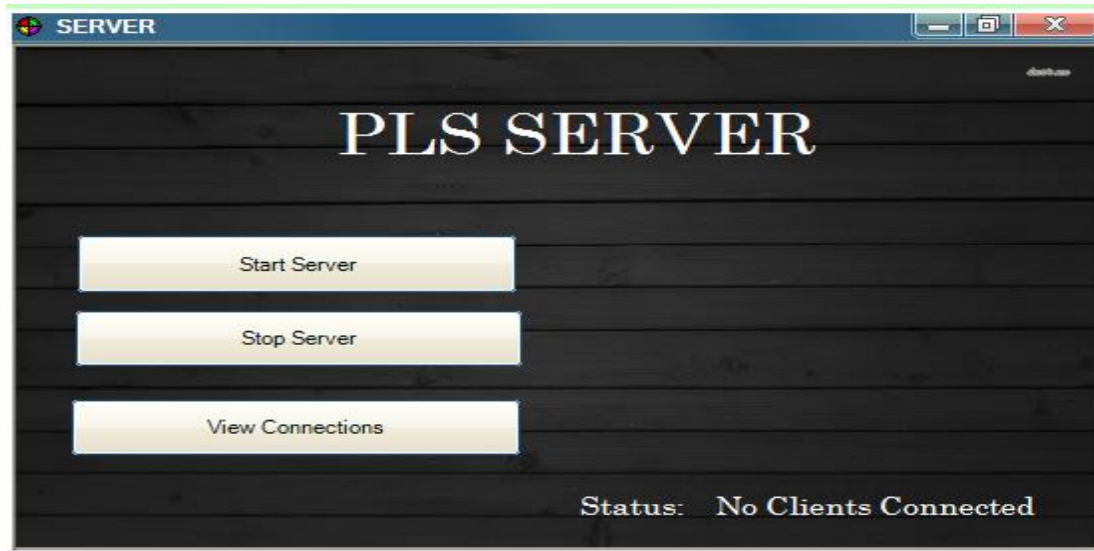


Figure 1 Server Form

### 6.1.2 Client UI



Figure 2 Login Screen



The screenshot displays a software window titled "PARKING LOT SYSTEM" with a menu bar (File, Configure, View, Help) and a toolbar. The main title "THE PARKING LOT SYSTEM" is centered at the top. Below the title, there are two tabs: "Simulation" and "Monitor PLS". The "Simulation" tab is active, showing two main panels: "Get Ticket" and "Pay Ticket".

The "Get Ticket" panel contains a "GET" button and a "Ticket" section. The "Ticket" section has a label "Available Spaces" above three empty rectangular input fields. The "Pay Ticket" panel contains a "PAY" button and a "Paying Machine" section. The "Paying Machine" section has a small white rectangular area above two empty rectangular input fields.

Figure 5 Simulation Form (Retrieval and Payment of Ticket)

The screenshot displays the same software window as Figure 5, but the "Monitor PLS" tab is active. The main content area is divided into two large vertical sections, each containing three horizontal green bars. The left section has three bars of a lighter green shade, while the right section has three bars of a darker green shade. The title bar and menu bar remain the same.

Figure 6 Monitor PLS Form



Figure 7 ArduinoConnect Form

## 6.2 UML Use Case Diagram

The Diagrams bellow shows the interaction between the users (Administrator, Security and Driver) whenever they are using the application, and notice that only the Security and Administrator have to login before using the application.

The figure 1 shows the overall picture of the respective functions of each user in this system. Where the main purpose of the Administrator is only to manage and view the records. In other words, the administrator will be able to perform the basic commands (Add, Delete, and Modify) records and some complex ones such as filtering and more. The Security will only be able to monitor the Parking Lot, that is, will be able to see which parking space is available and which one is not available (not to mention, that in the security room, will have to have cameras for safety reason), and at the end of the day, he will be able to print the report of the people in and out of the parking, helping like this, the control of people that goes in the shopping or other center per day. It is to be seen that the Administrator and the Security, have to login before doing any other operation. The figure also shows the operation that the drivers will execute, where the driver will be able to get the ticket, and park wherever he wants as long as it is a free space, and will also be able to pay the ticket as soon as he decides to live the facility.

Figure 1 will be better explained throughout the report, where some of the scenario descriptions of the conditions that have to be met in order to achieve the actions referred on the diagram, to be more precise, it will explain the conditions (pre and pro) and the steps necessary to do a certain action like “Get Ticket” (from Driver perspective).

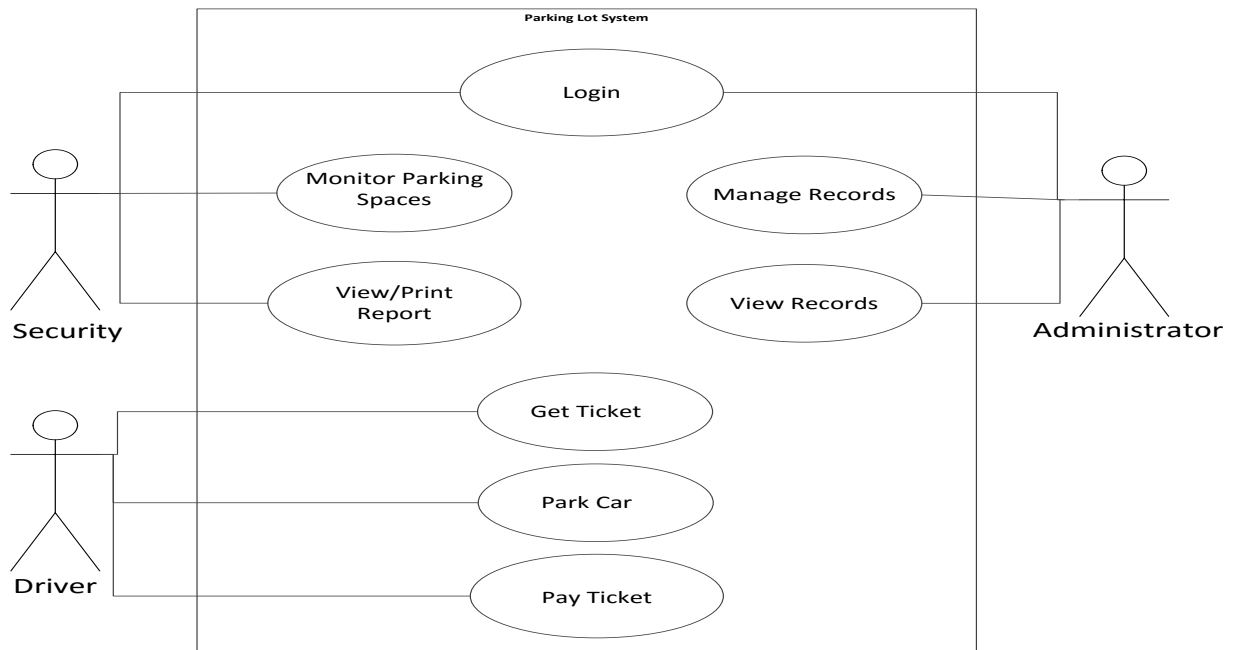


Figure 8 Overall Use Case Diagram

## 6.3 Class Diagram

### 6.3.1 High Level Class Diagram

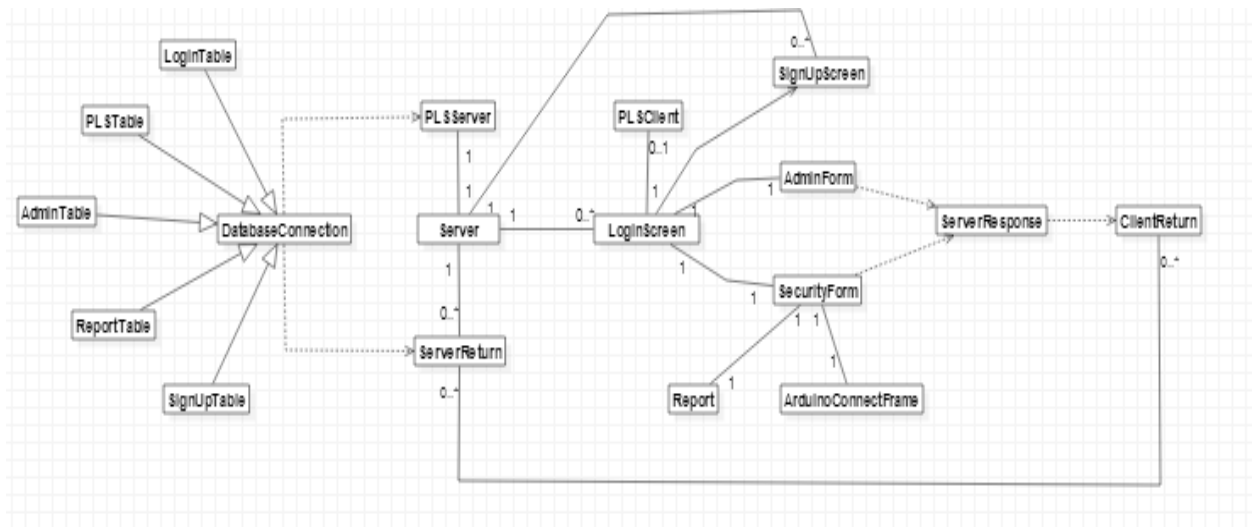


Figure 9 Overall class relationship diagram

The Parking Lot System (PLS) is an application with two sides, where one side will not work properly without the other, especially on the client side. As it can be seen on the diagram, LoginScreen class can exist without the Server class, and vice versa, however, in order to access other classes, LoginScreen will have to “Request” for permission on the server, that is, check whether the input value exists on the database giving like this access on the other forms, according to the credentials.

Therefore, all application goes around four main classes “Server, LoginScreen, ClientReturn and ServerReturn”, because if the communication between is Server and LoginScreen is successfully, then maintaining the communication between them will depend on the ClientReturn and ServerReturn which receives and sends other requests to access the database.

### 6.3.2 Detailed Class Diagram for the Client

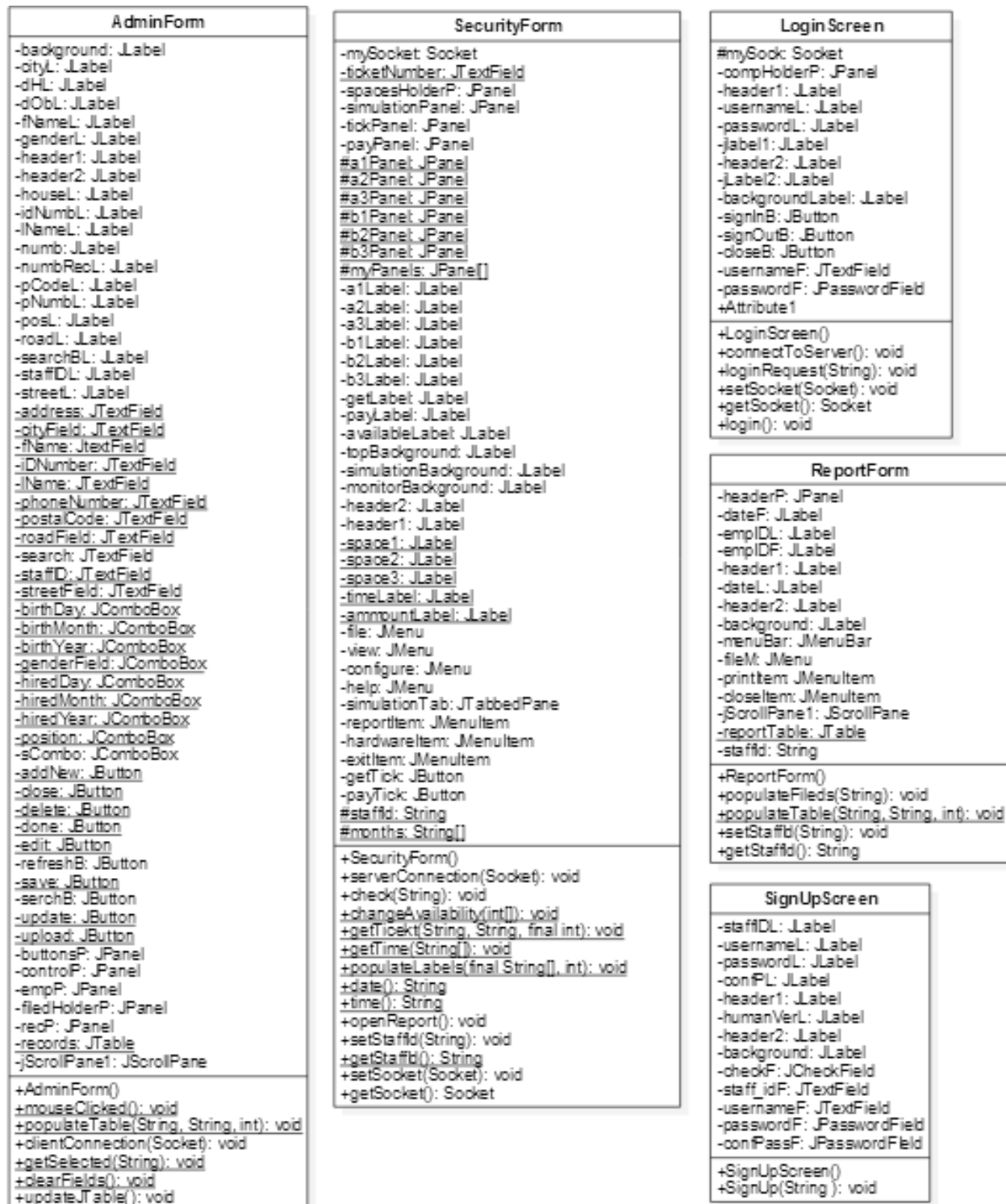


Figure 10 Classes that will interact with the user

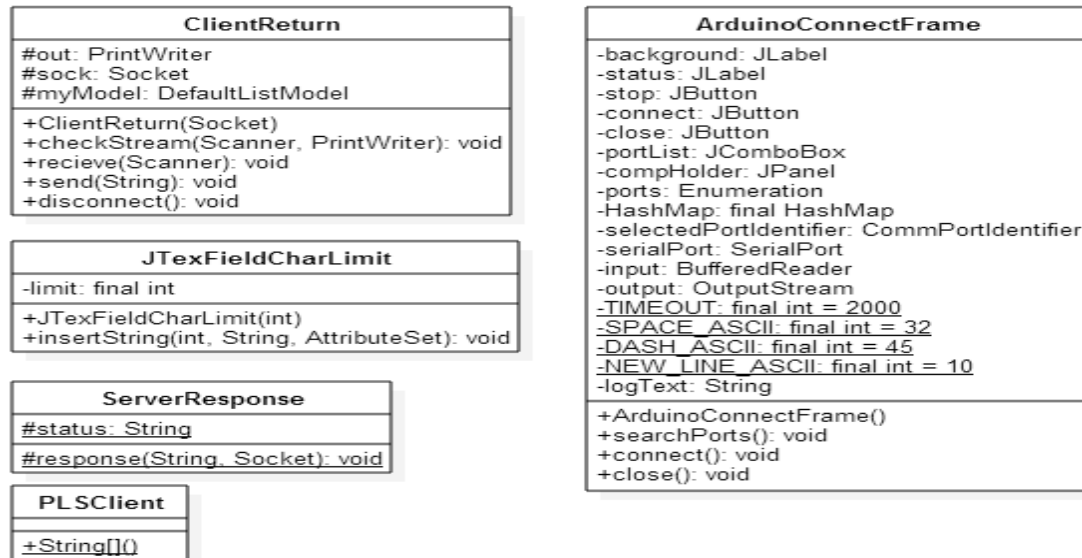


Figure 11 Detailed Class Diagram of the other classes

### 6.3.3 Detailed Class Diagram for the Server

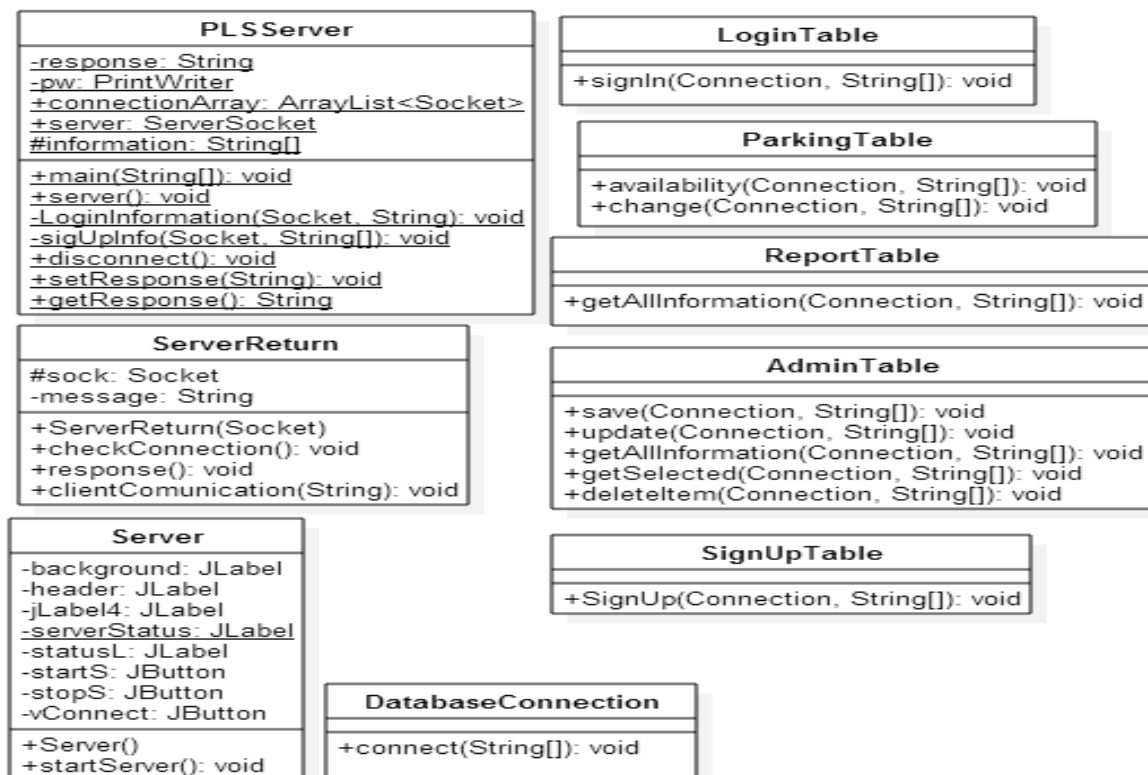


Figure 12 Overall Detailed Class Diagram of the Server Side



## 6.4 Activity Diagram

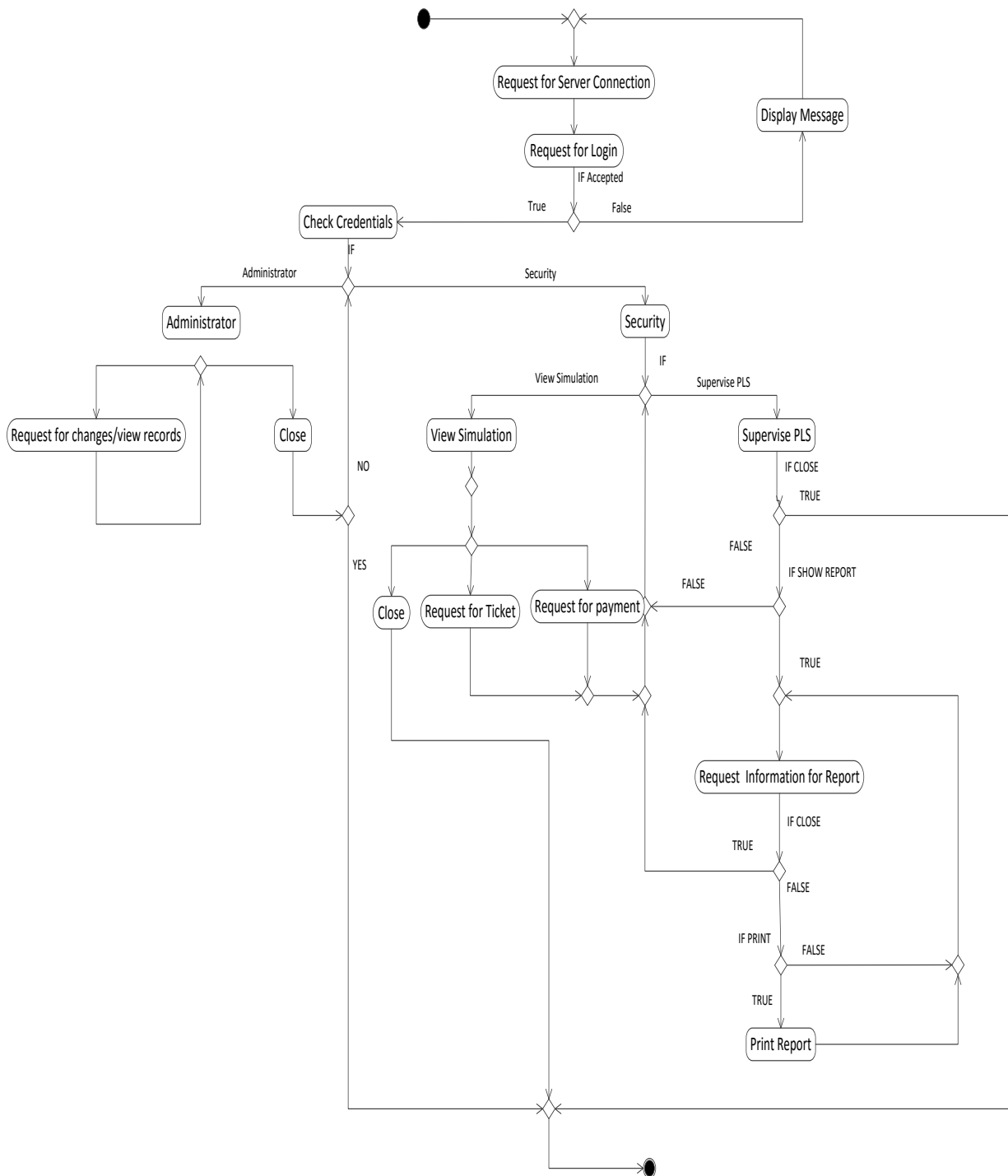


Figure 13 Overall Activity Diagram

The activity diagram above shows how the system will work from both perspectives, whether the user is the Administrator or a Security.

In this diagram, it shows how the system will work on the client side, showing that the client is always requesting the Server for information in order to be used. Because the overall information is in the database, the simulation makes it look like the drivers will also have to login in order to park their cars. But what actually happens is that, the user comes in a time, assuming that the security is there, therefore, in order for the system to be continuously running on the parking system, the Security must be logged in, or else the system will be down, and no movement of tickets will be made.

## 6.5 Sequence Diagram

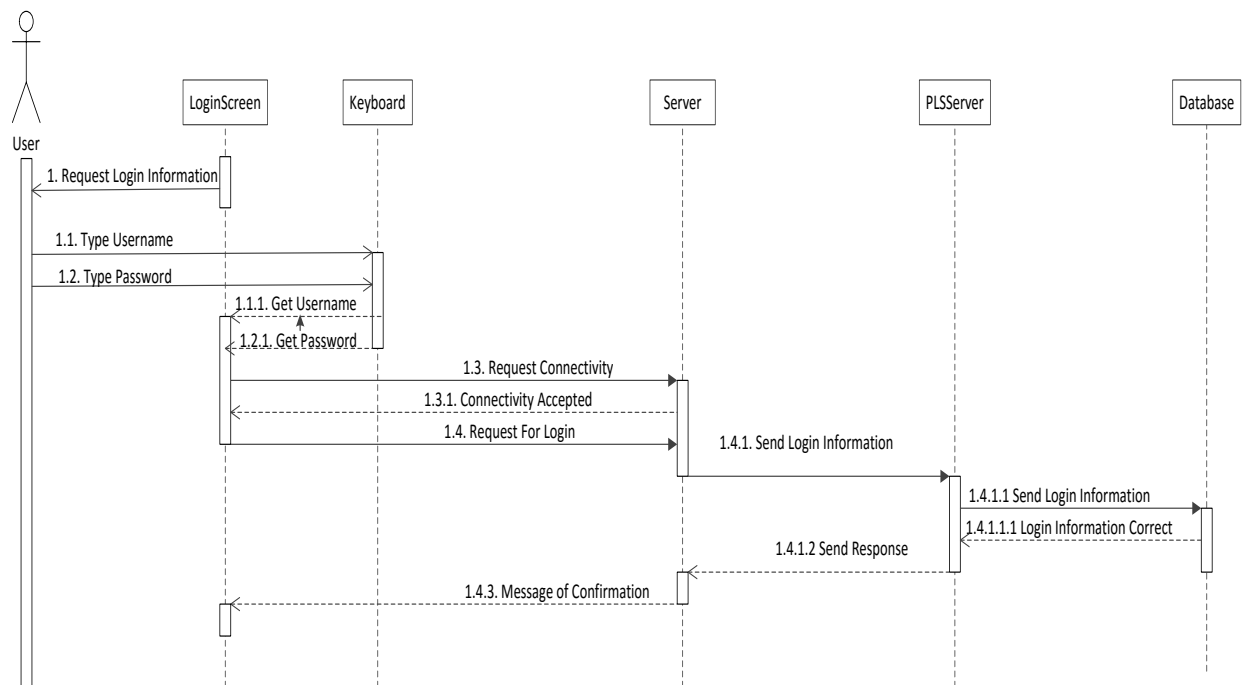


Figure 14 Sequence Diagram for Login

This is the most important part of the System, since before doing anything, it is necessary to login to keep the all application up and running. As was explained before about the four main classes (LoginScreen, Server, ServerReturn and ClientReturn), here it can be seen more clearly that in order to access the database, the client side of the communication, has to request first on the server permission to access the database, in case of connection failure, the access to the database would be impossible and it would not be possible to login.

## **7 Problem Design**

By now, the understanding of the overall functionality of the application must be clear. In this phase, the functionalities of PLS will be broken into smaller pieces and explained state by state. In other words, will show exactly what and how it will happen on the system as soon as the user tries to do a specific scenario.

### **7.1 PLS Scenarios**

The PLS is divided in two parts: Software and Hardware. For the purpose of this phase, the hardware part will also be tested in order to have a more complete understanding of how the system works.

Since there are three users (Administrator, Security and Driver) in this application, we will divide the scenarios according to the users since each user will have different outputs.

Note: Every scenario is taking into consideration that the user has already logged in.

### **7.2 Administrator Scenarios**

For the Administrator, we will focus only on Adding a new record and deleting a Record, since the process will be almost the same when it comes down to the other buttons.

## 7.2.1 Adding New Record

### User Interface

**THE PARKING LOT SYSTEM**

**View Employees**

STAFF_ID	First_Name	LAST_NA..	GENDER	DATE_OF...	ID_NUMB...	PHONE...	ADDRESS	ROAD
1	Manuel	Lopes	Male	9 May 1994	1	014216419	204 Marat...	Marais R
2	K	K	null	null	null	null	null	null
3	Henrique	Niuca	Male	1 January ...	BN1213			
4	Unquidea	Lopes	Female	17 Novemb...	NUH43			
5	Linea	Monica	Female	7 Novemb...	H983			
6	John	Napier	Male	5 January ...	12	06134567...	12 Grove...	High Lev
7	Ricardo	Costa	Male	6 April 1993	7771291...	07122234...		

**Manage Records**

First Name:  Last Name:

Gender:  Date of Birth:

Road:

City:

Postal Code:  Phone Number:

ID Number:  Staff ID:

Position:  Date Hired:

**Buttons**

**Records Control**

Search By:

Number of Records: 7

**Message**

The record has been added

Figure 15 Administrator adding a new record

### Use Case Scenario

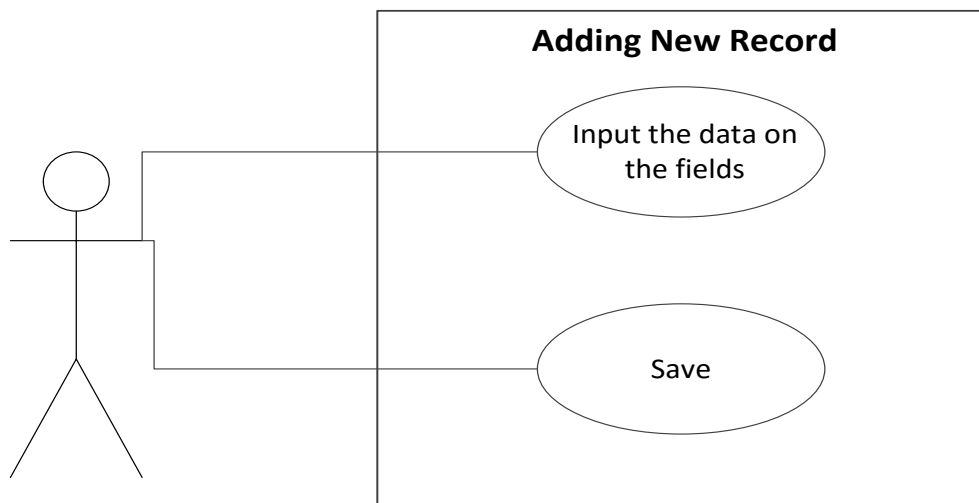


Figure 16 adding a new record Use case Scenario

## Scenario Description

Pre-Condition: The “Add new” button should be clicked.

Steps:

- Input all the information about the new employee
- Click on Save button

Post-Conditions: none

Alternatives: Click on the refresh button to refresh the JTable and see the record added.

## Activity Diagram

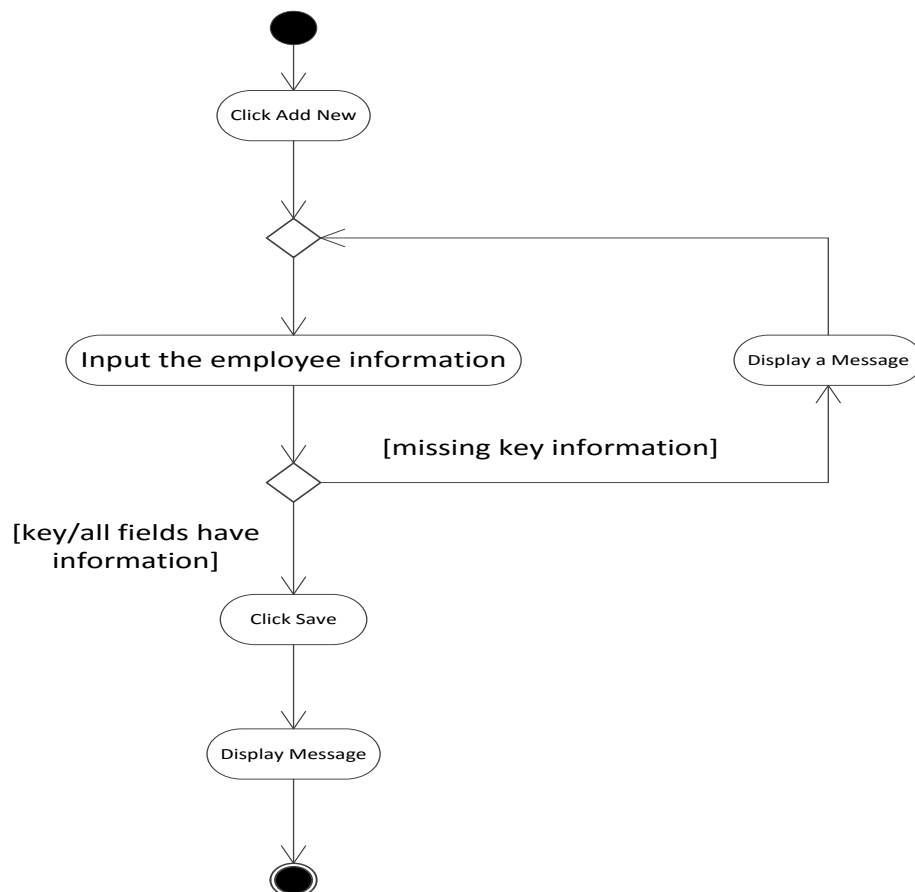


Figure 17 Use case for the saving button

### 7.2.2 Deleting a record

#### User Interface

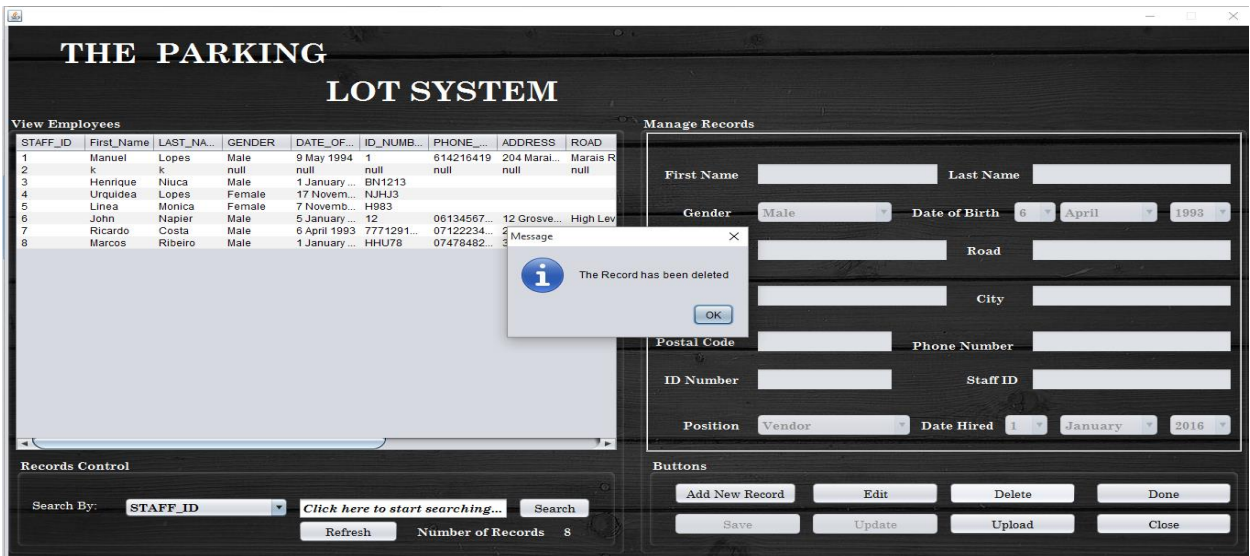


Figure 18 User interface for Deleting a record

#### Use Case Diagram

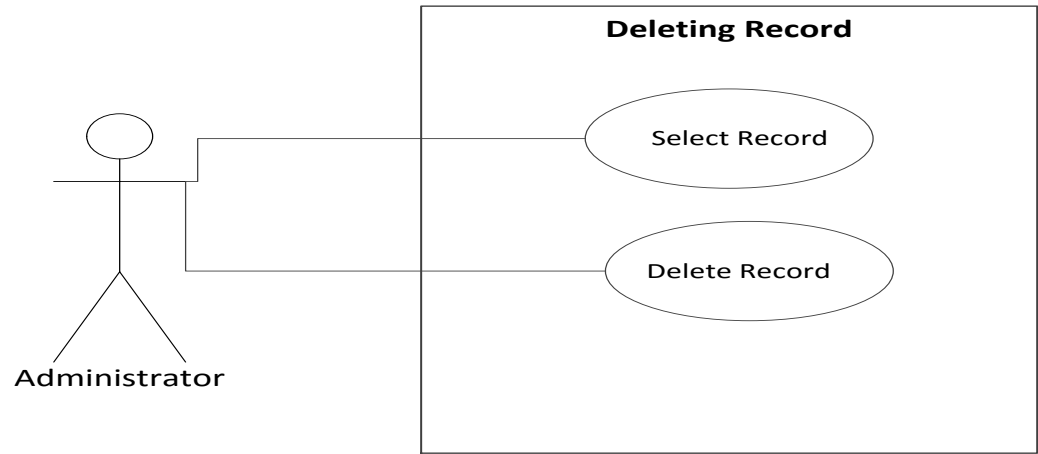


Figure 19 Use case for Deleting a record

## Scenario Description

Pre-Condition: none

Steps:

- Search for the desired record to be deleted
- Select the record if 2 have or more information seen
- Delete selected record

Post-Conditions: none

Alternatives: Click on the refresh button to refresh the JTable and see the record Deleted

Activity diagram

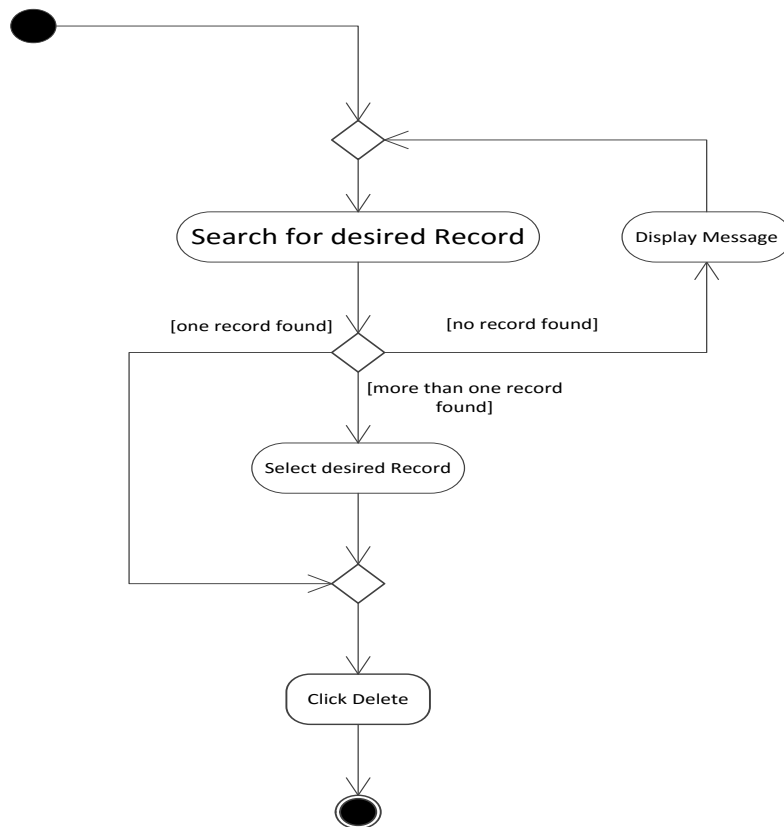


Figure 20 Activity Diagram for deleting a record

As it can be seen on the figures above, to be able to add, or delete a record, some steps have to be taken into account. The application to add and delete, will have to send information to the server,

just like it was happening when explained using a sequence diagram of how the server and client communicate using Server/Client communication. In this case it is the same, the client (AdminForm) will send the information to be added on the database via Server/Client communication and after the database insert the information, and it will send a response to the client saying that the information was added successfully. The same thing happens with the deletion of a record.

## 7.3 Security Scenario

In this case, we will focus on the report and the monitoring the Parking lot.

### 7.3.1 Monitoring the PLS

User Interface

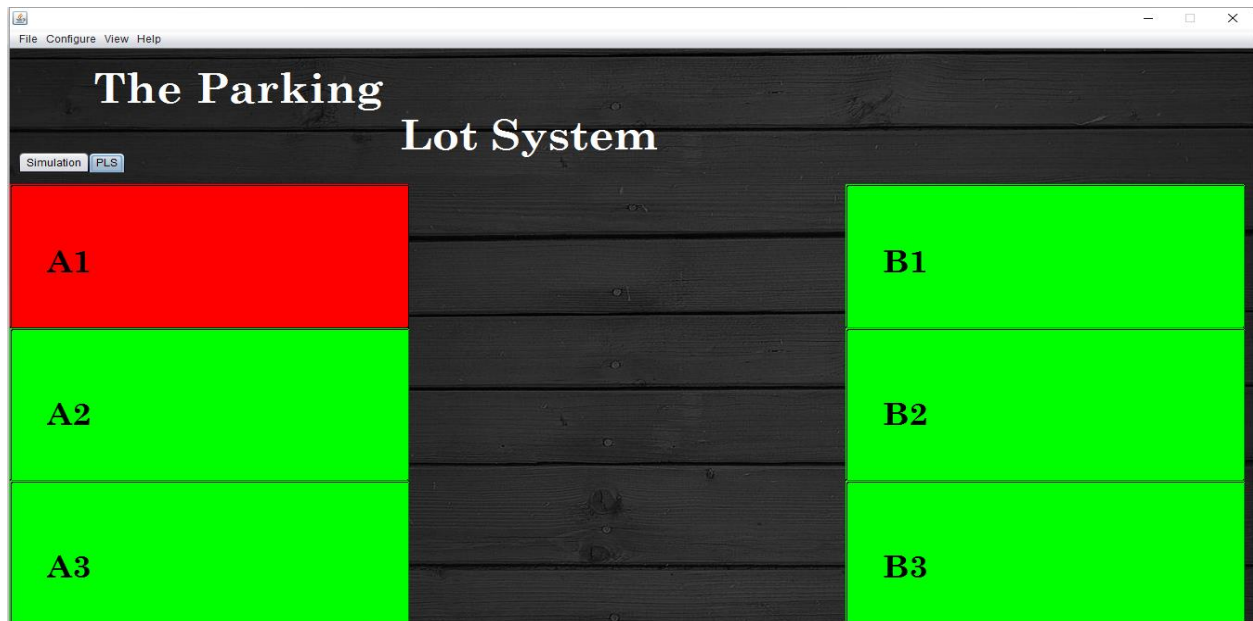


Figure 21 User Interface for Monitoring PLS



## Use case Diagram

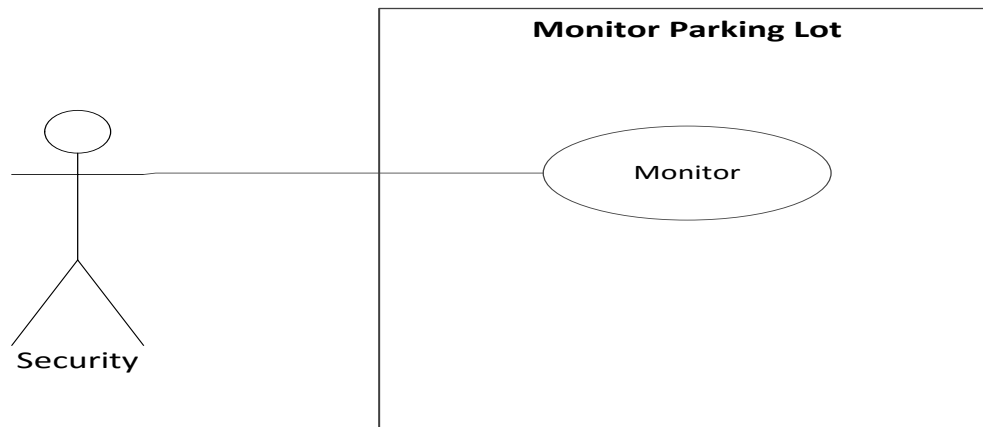


Figure 22 Use case Diagram for Monitoring the parking system

## Scenario Description

Pre-Condition: none

Steps: none

Post-Conditions: none

## Activity Diagram

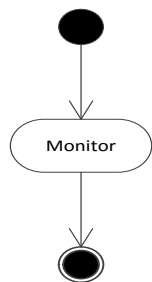


Figure 23: Activity Diagram when monitoring the PLS

The Security is the most limited in terms of using the application, since he will only monitor and get the report after the day of work. In this phase, we also tested the hardware part, as it could be seen on the Figure 21; there is one red light and five green lights. The Green lights represent the parking space is not being used, and the red the parking space is being used. In the figure above, the space at A1 is red, therefore, there's a car parked on that space.

## 7.4 Driver Scenario

The Driver is the third and most important part of this application because is the driver that will buy, pay the ticket and even park his car. This application was developed especially to help the driver to know where exactly he can park he's car without wasting his or her time looking for the parking place.

### 7.4.1 Getting the Ticket

User Interface

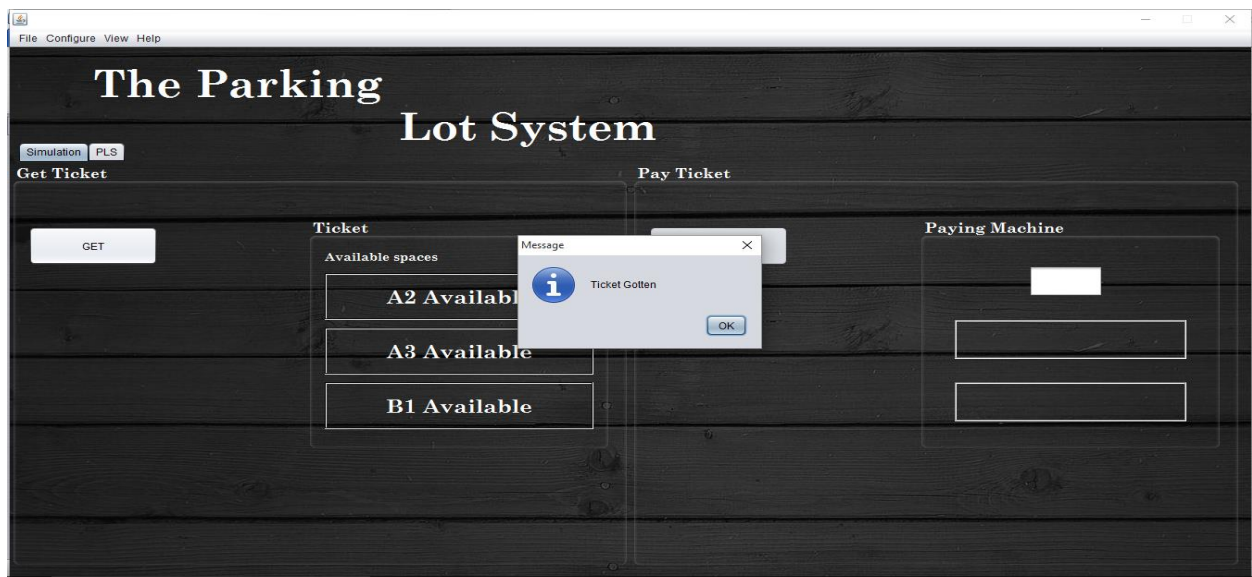


Figure 24 UI of the simulation when getting a ticket

Use Case Diagram

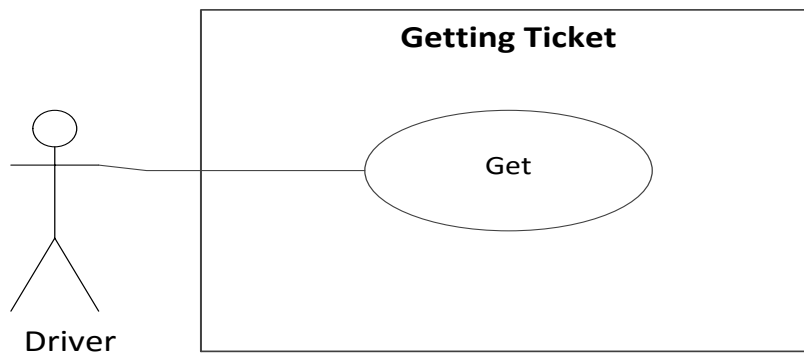


Figure 25 Use case Diagram for when getting the ticket

Scenario Description

Pre-Condition: none

Steps: none

Post-Conditions: none

Use Case Diagram

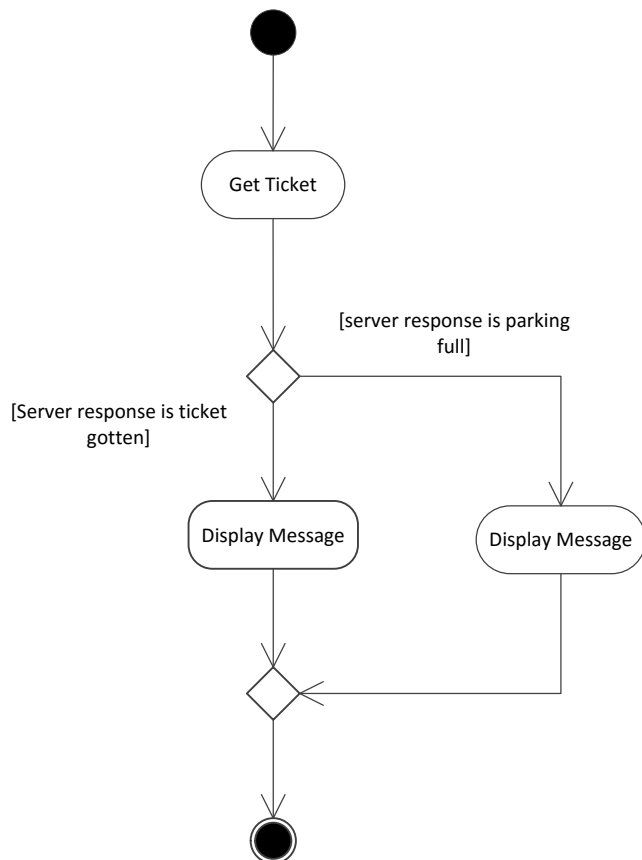


Figure 26 Activity Diagram for when getting a ticket

### 7.4.2 Pay Ticket

#### User Interface

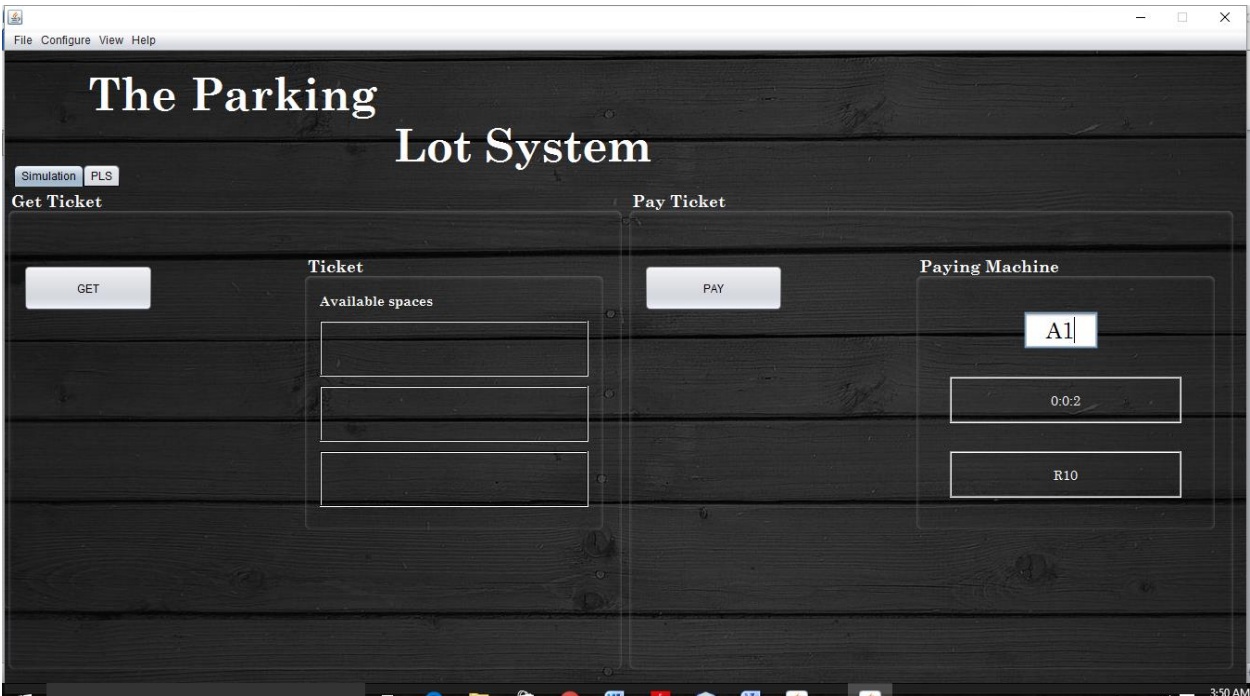


Figure 27 UI for when paying the ticket

#### Use Case Diagram

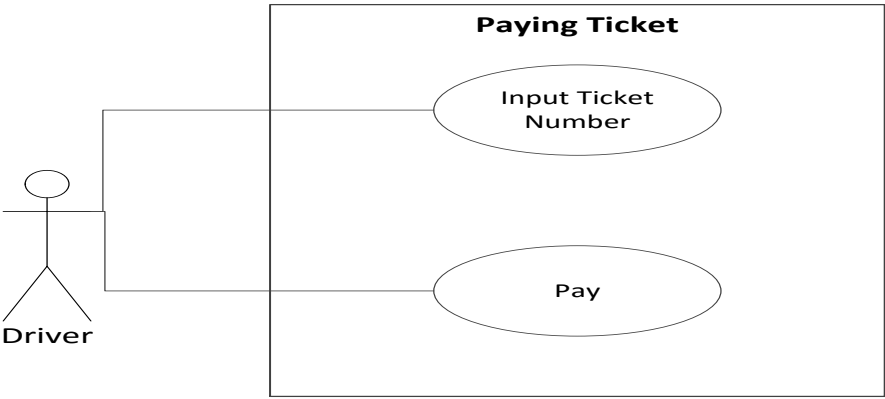


Figure 28 Use case diagram for paying ticket

## Scenario Description

Pre-Condition: Ticket must be gotten

Steps:

- Insert the ticket number inside the textField
- Press Enter to get the time and amount to be payed
- Click on the pay Button

Post-Conditions: Take car from the parking lot

## Activity Diagram

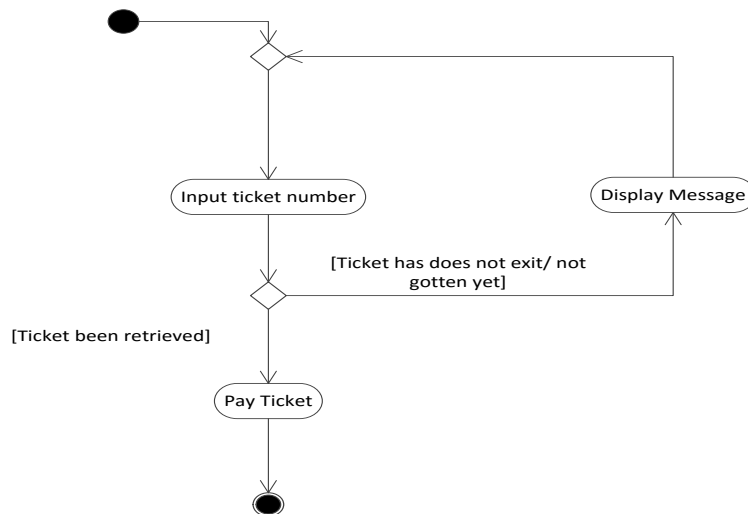


Figure 29 Activity Diagram for when paying the ticket

As it can be seen in the above figures, as soon as the driver takes the ticket, in the ticket panel which is the simulation of a ticket, there are three spaces, in those empty spaces, will be the available spaces helping like this the driver to find the desired parking space. In case it is full, a message will be displayed. To pay the ticket, the driver will have to input the ticket number on the ticket machine, and the time and amount to be payed will be shown. However, if a ticket that hasn't been taken is entered, a message will also be displayed warning the driver that that ticket doesn't exist.

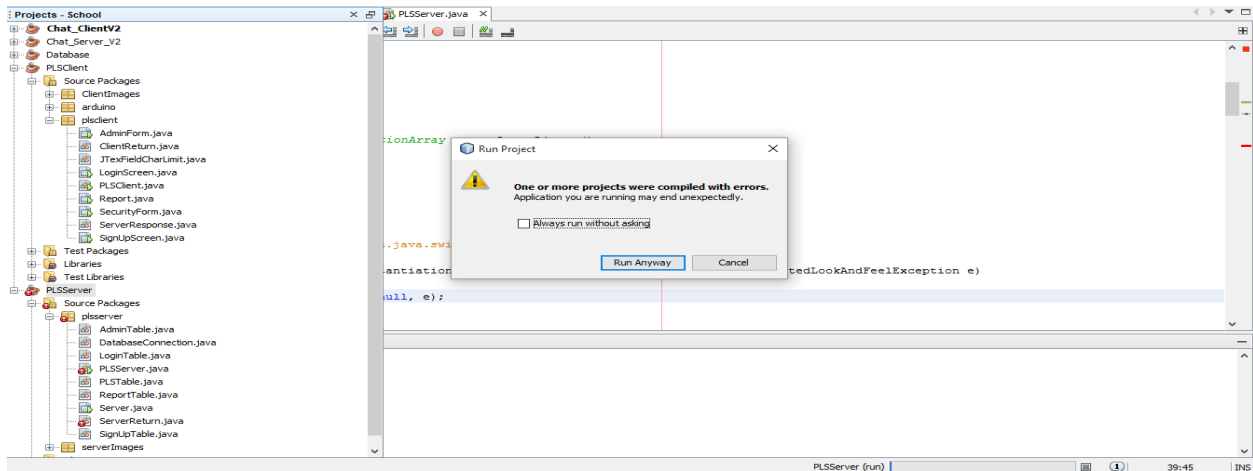
Note: The input of the ticket number is only to simulate a real ticket going in the ticket machine.

## 8 Test Reports

### 8.1 White-Box Testing

Test Performed: Trying to run the Server

Test Result: Error while trying to run the Server



Solution: Was trying to put a Socket into and ArrayList<Strings>, therefore, created and ArrayList<Sockets>

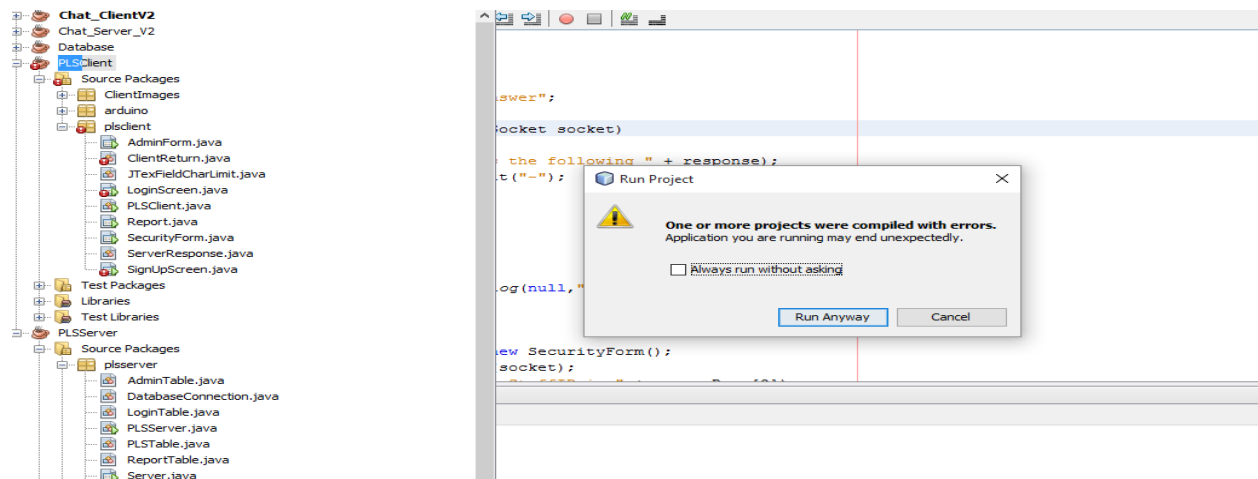
Test Performed: Trying to run the server again to check more errors

Test Results: None, the Server is running normally.



Test Performed: Trying to run the Client.

Test Result: was not possible to run the client



Solution: had to change the method serverResponse from public void to public static void.

Test Performed: Trying to run the Client again to check more errors

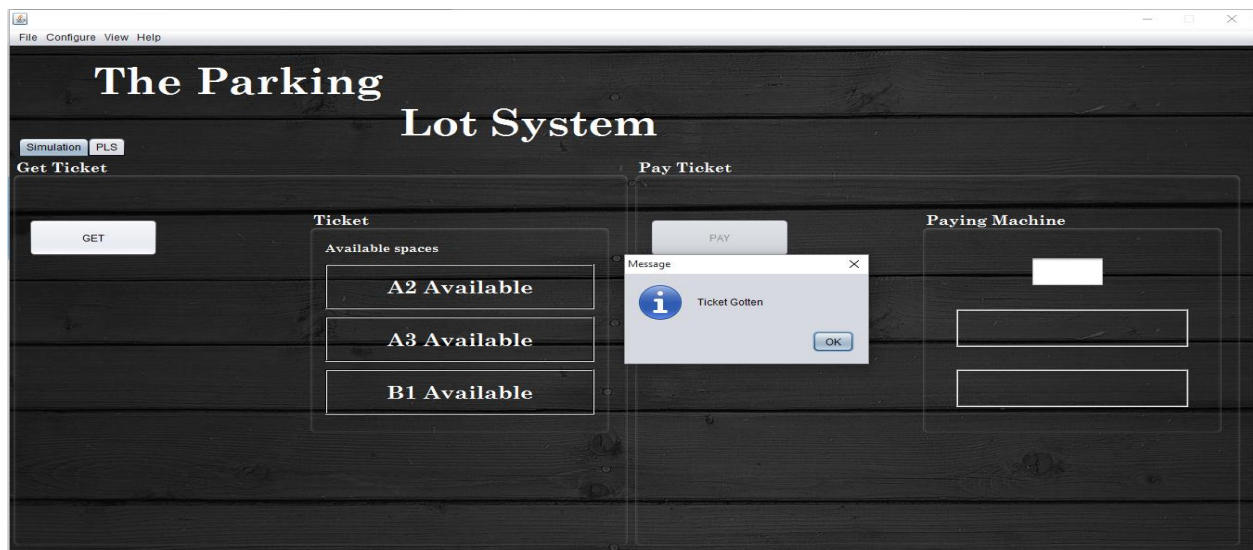
Test Results: None, the Server is running normally.



## 8.2 Black-Box Testing

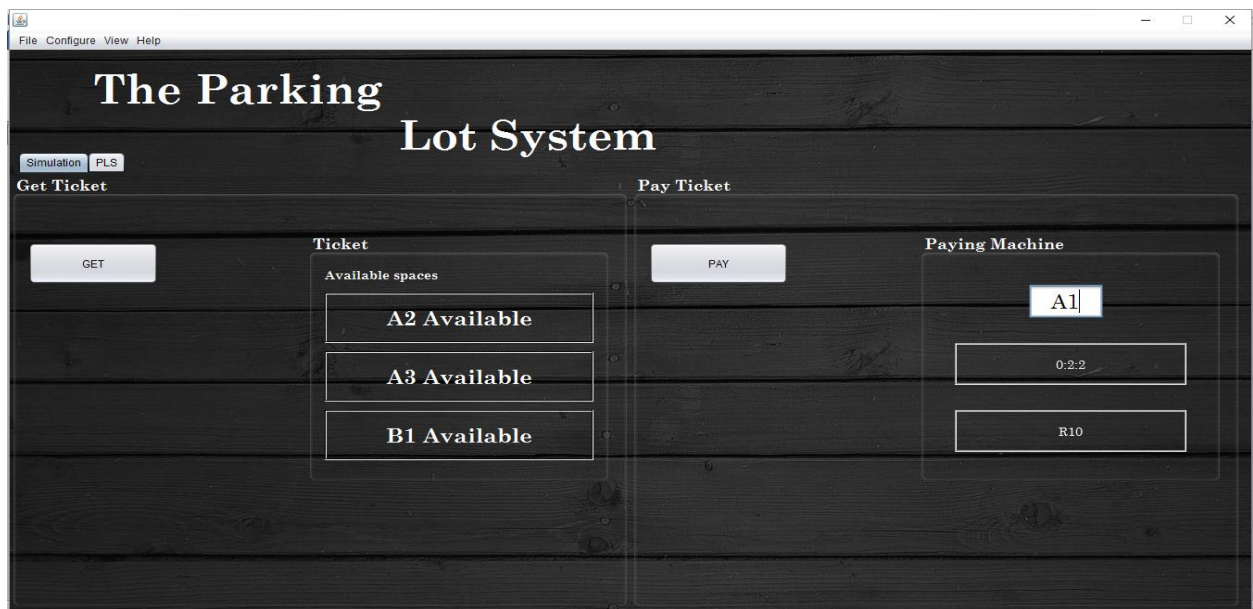
Test Performed: Trying to get the ticket and get the available spaces to park

Test Result: it worked as expected



Test Performed: Trying to get the time spent while being parked from the ticket

Test Result: it worked as expected

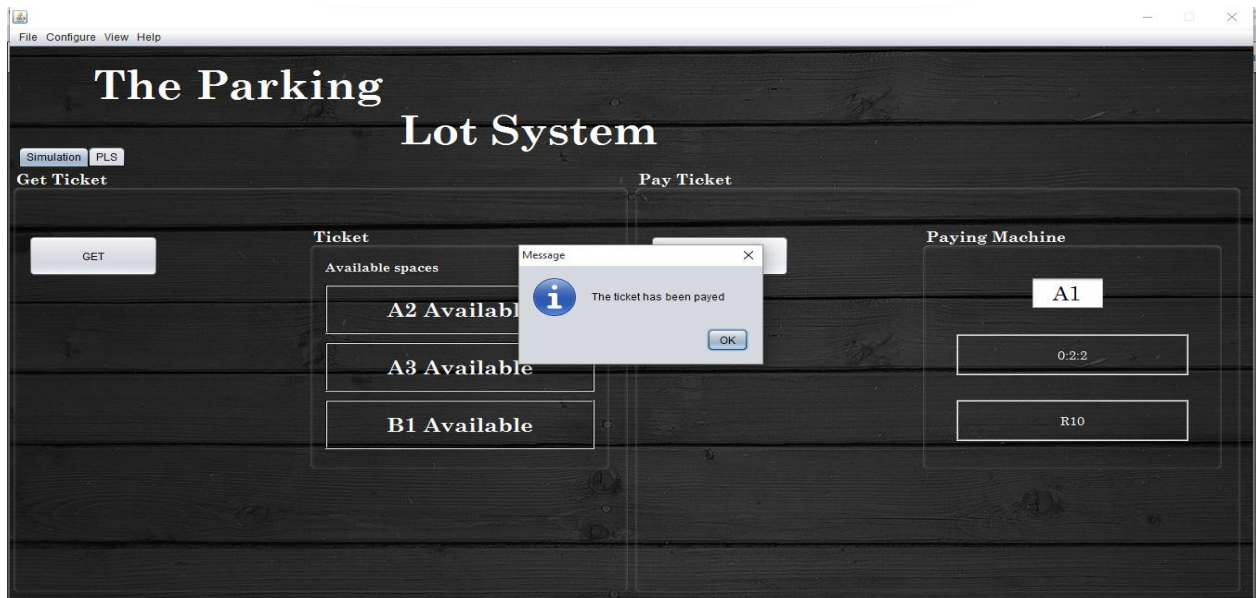




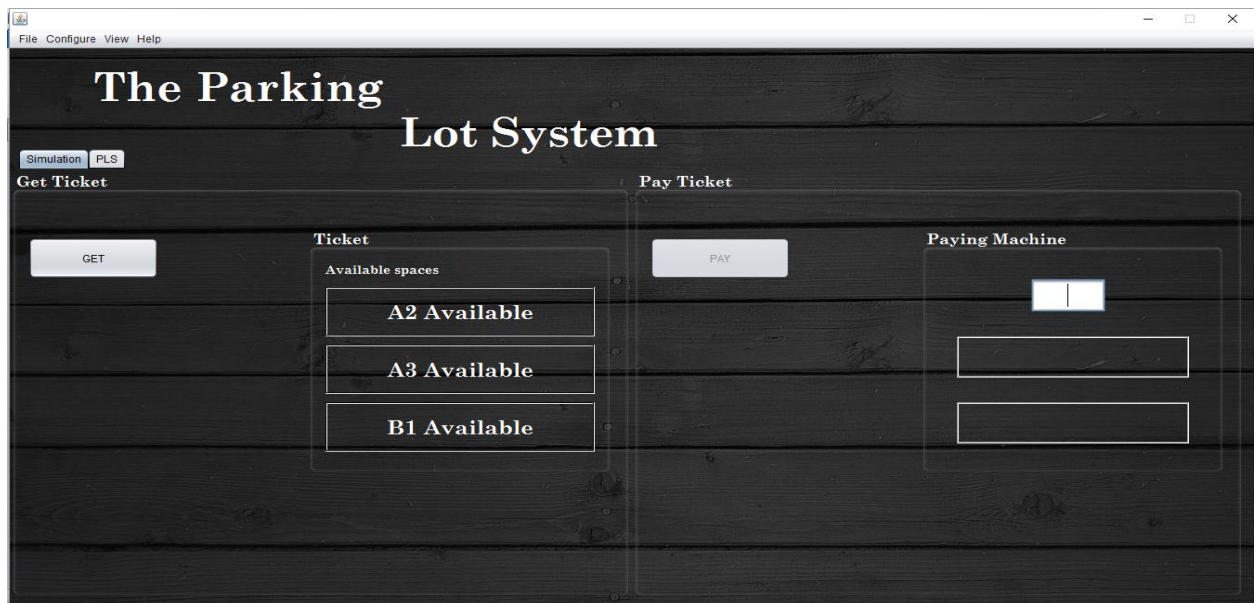
Test Performed: Trying to pay the ticket and expecting to clean the fields afterwards

Test Result: it worked as expected

Screenshot 1:



Screenshot 2:



Test Performed: Expecting to connect to the port in order to get serial communication with the Arduino Uno.

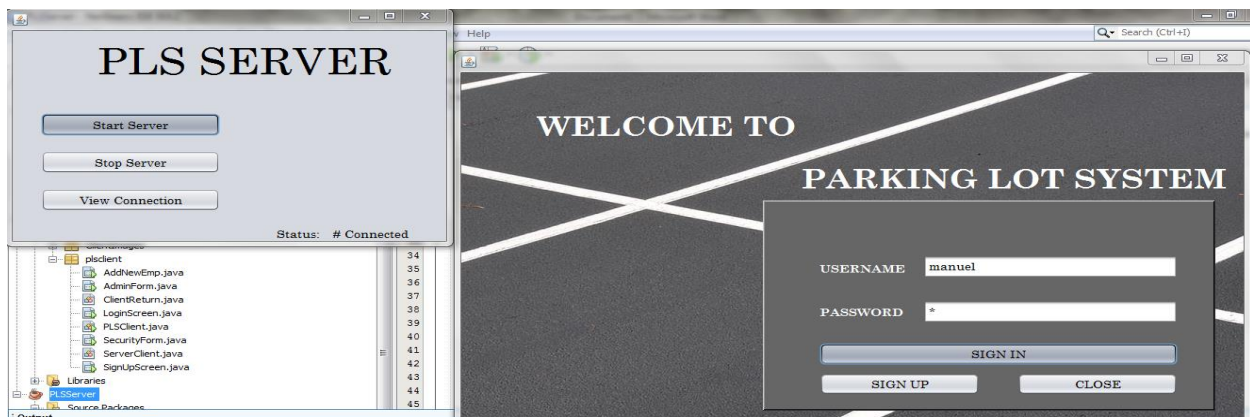
Test Result: it worked as expected



### 8.3 Module Testing

Test performed: Trying to connect the client with the server

Test Results: The Server and the client are both stuck. And the remaining information is only sent to the server after the disconnection of the client.



Solution: Change the PrintWriter command from write to println.

Test performed: Trying to put an incorrect password

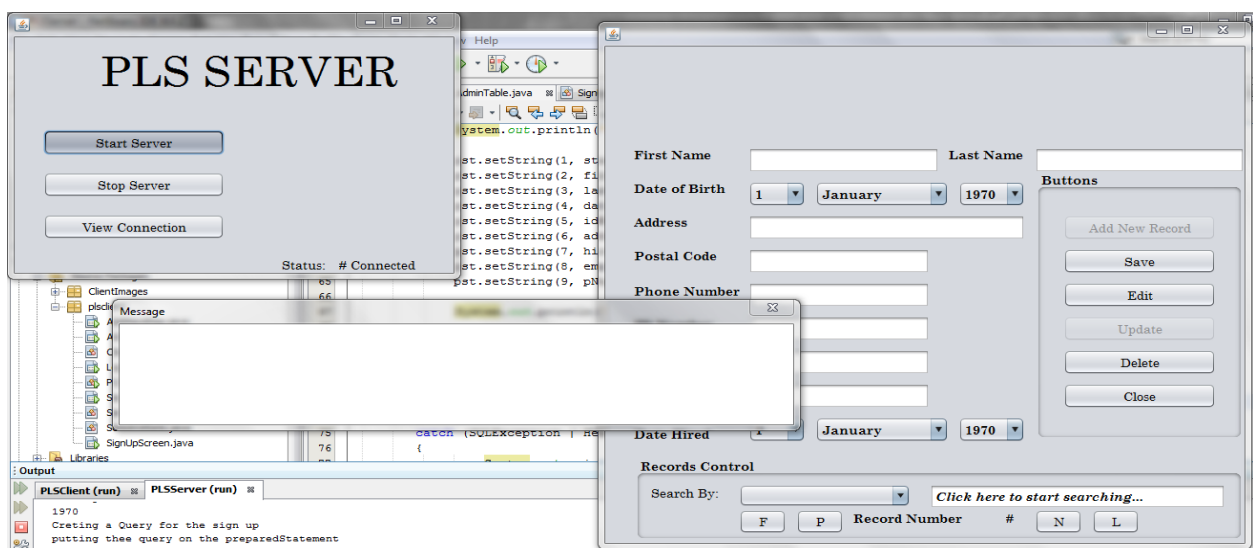
Test Result: Giving an out of bounds error.



Solution: change on the response of the server when incorrect so send 2 parameters instead of one.

Test Performed: Trying to add a new employee

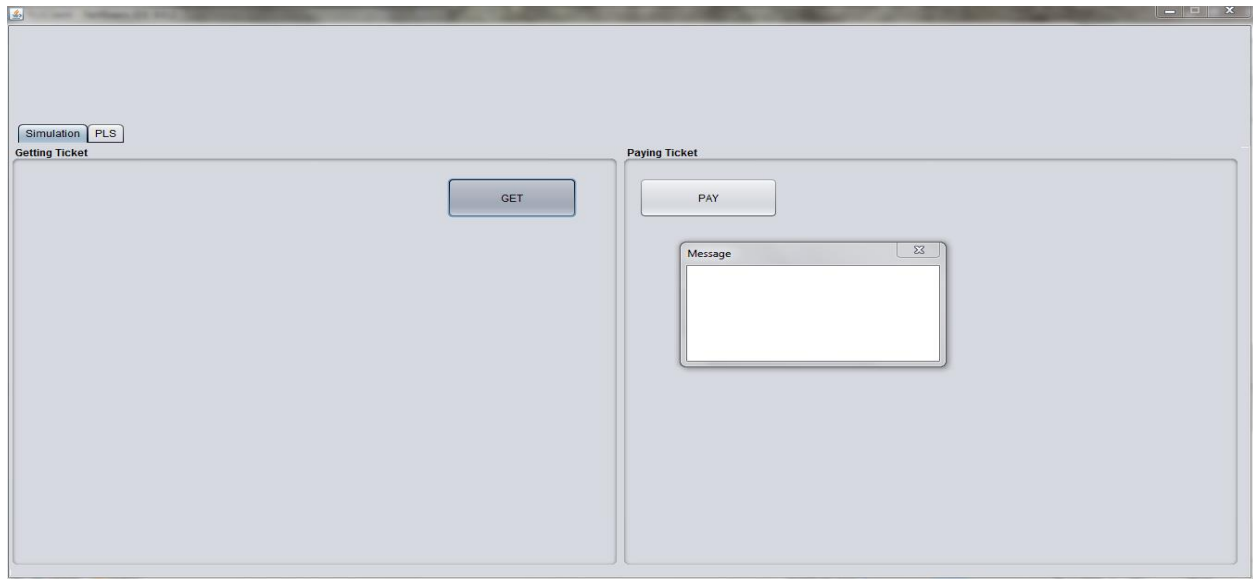
Test Result: it was connecting to the database but it was giving a query error where a semi colon was missing from the query.



Solution: take one parenthesis from the query.

Test Performed: Tried to get the availability of the ticket

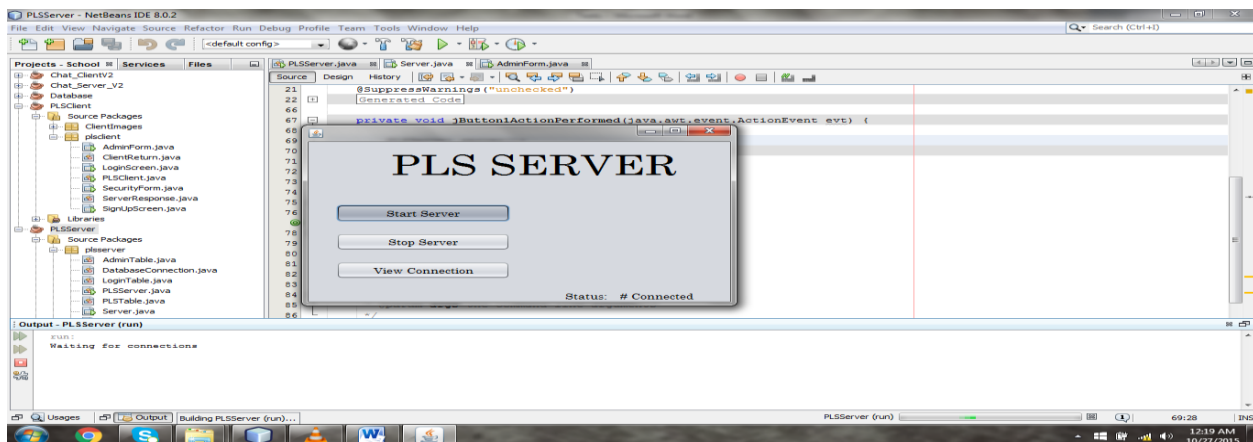
Test Result: The server could get the availability based on the information of the client but it couldn't send the response back therefore the application got stuck.



Solution: Change the classes where the accessors and mutators where, and put it on the main class.

Test Performed: Trying to start the server and wait of the button start server will be disabled.

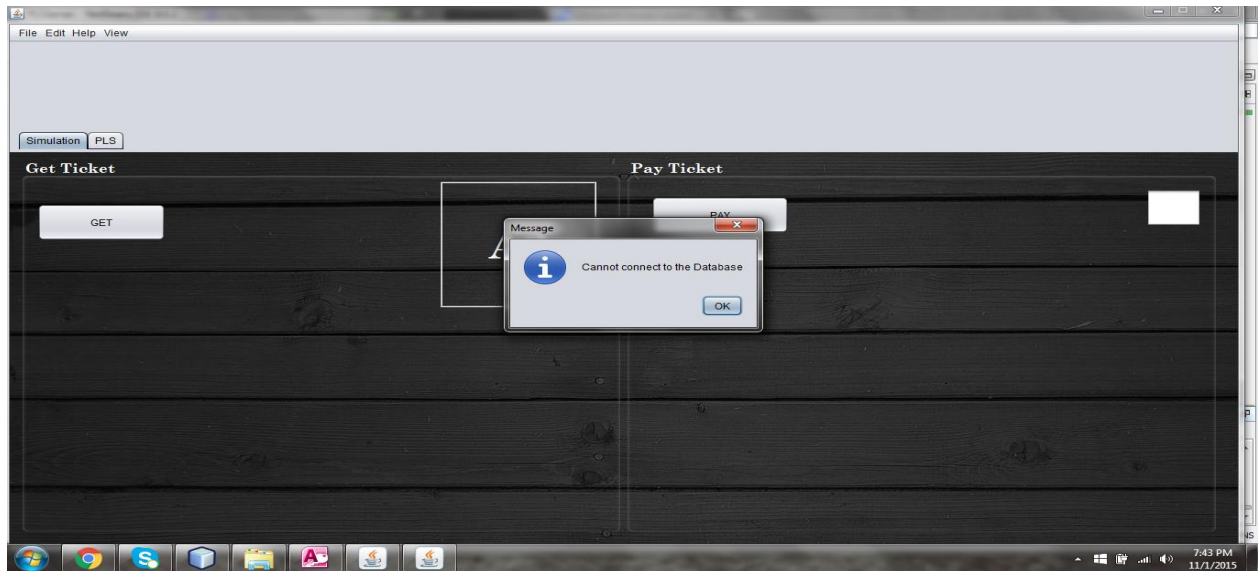
Test Result: The Button to start the server is stuck and no other button can be pressed .



Solution: Implement a Thread on for the button start server and then disable the button.

Test Performed: Trying to get the ticket and update the Report table with some information.

Test result: Ticket was retrieved but an out of bound index error was shown when trying to update the Report table.



Solution: the preparedStatement was missing on the insert Statement.

## **9 Evaluation Report**

### **9.1 Client Perspective**

The Parking Lot System meets the user requirement. The application is straight forward, once it is explained what and how to use, anyone can use it.

### **9.2 Programmer Perspective**

The Parking Lot System was designed in a way to help the driver to pinpoint a free parking space at any moment, by that I mean to pinpoint where to park his car, even if the parking is almost full or empty. Following this pattern, we can say that the system meets the user specification.

### **9.3 Features**

Some features were added in comparison to the normal parking system:

- In the ticket, come a maximum of three and a minimum of 1 available space, whenever the ticket is withdrawn from the ticket machine.
- It uses LDR sensors, in order for the application to be able to pinpoint the exact location of the space occupied and update the database with the information.
- It reduces the time for the driver to look for the parking space

### **9.4 Problem Areas**

No human work is perfect; therefore, there are some errors that can be seen. One of them is the fact that after adding a record on the administrator form, it does not update automatically, in order to update it, the coder created a button to refresh the table.

Since this is a Client/Server communication application, sometimes the server will not be able to respond.

This application will only be able to work whenever the application on the same computer, since the ip “127.0.0.1” is being used.

## **9.5 Future Improvements**

The following is what is programmed to be the future improvements:

- The Security will be able to print the report
- The Client/Server communication will be better incorporated
- The application will be able to connect to whatever IP that is in the network
- In the Administrator form, after the record has been added, modified or deleted, it will refresh automatically, without the aid of another button
- The search will be dynamic, that is, while typing the search will be made.

## Bibliography

- [1] "Wikipedia," 19 August 2015. [Online]. Available: [https://en.wikipedia.org/wiki/Computer\\_hardware](https://en.wikipedia.org/wiki/Computer_hardware). [Accessed 20 August 2015].
- [2] "Wikipedia," 16 August 2015. [Online]. Available: <https://en.wikipedia.org/wiki/Software>. [Accessed 20 August 2015].
- [3] "Wikipedia," 21 March 2015. [Online]. Available: [https://en.wikipedia.org/wiki/Electronic\\_grade\\_book](https://en.wikipedia.org/wiki/Electronic_grade_book). [Accessed 20 August 2015].
- [4] M. Rouse, "TechTarget," May 2009. [Online]. Available: <http://searchenterpriselinux.techtarget.com/definition/embedded-system>. [Accessed 19 August 2015].
- [5] "Wikipedia," 12 July 2015. [Online]. Available: [https://en.wikipedia.org/wiki/Blackboard\\_Learn](https://en.wikipedia.org/wiki/Blackboard_Learn). [Accessed 19 August 2015].
- [6] "Wikipedia," 19 August 2015. [Online]. Available: <https://en.wikipedia.org/wiki/Sensor>. [Accessed 20 August 2015].
- [7] "Wikiperia," 18 August 2015. [Online]. Available: <https://en.wikipedia.org/wiki/Transducer>. [Accessed 20 August 2015].
- [8] "Wikipedia," 17 August 2015. [Online]. Available: <https://en.wikipedia.org/wiki/Frequency>. [Accessed 20 August 2015].
- [9] J. Lucas, "liveScience," 06 April 2015. [Online]. Available: <http://www.livescience.com/50399-radio-waves.html>. [Accessed 20 August 2015].
- [10] "Wikipedia," 16 August 2015. [Online]. Available:



- [https://en.wikipedia.org/wiki/Hacker\\_\(computer\\_security\)](https://en.wikipedia.org/wiki/Hacker_(computer_security)). [Accessed 20 August 2015].
- [11] S. M. a. P. R. Carlos Coronel, "Database Systems," in Database Systems: Design, Implementation and Management, Joe Sabatino, 2011, p. 5.
- [12] M. Rouse, "TechTarget," October 2008. [Online]. Available: <http://searchnetworking.techtarget.com/definition/client-server>. [Accessed 20 August 2015].
- [13] "Wikipedia," 17 August 2015. [Online]. Available: <https://en.wikipedia.org/wiki/Arduino>. [Accessed 20 August 2015].
- [14] K. Storch, "IT 103: Information and Technology," [Online]. Available: <http://kstorch.blogspot.com/2009/02/blackboard-advantages-and-disadvantages.html>. [Accessed 18 August 2015].
- [15] I. C. R. Society, "Imperial College Robotic Society," 30 October 2011. [Online]. Available: [http://icrobotics.co.uk/wiki/index.php/IR\\_Sensors](http://icrobotics.co.uk/wiki/index.php/IR_Sensors). [Accessed 18 August 2015].
- [16] K. Evans, "Opposing Views," [Online]. Available: <http://science.opposingviews.com/disadvantages-wireless-alarm-system-1472.html>. [Accessed 18 August 2015].
- [17] P. R. a. C. Coronel, "Database Systems: DESIGN, IMPLEMENTATION AND MANAGEMENT," in Database Systems, Course Technology, 2009, p. 69.

## Appendix A: User Manual

### 10 Getting Started

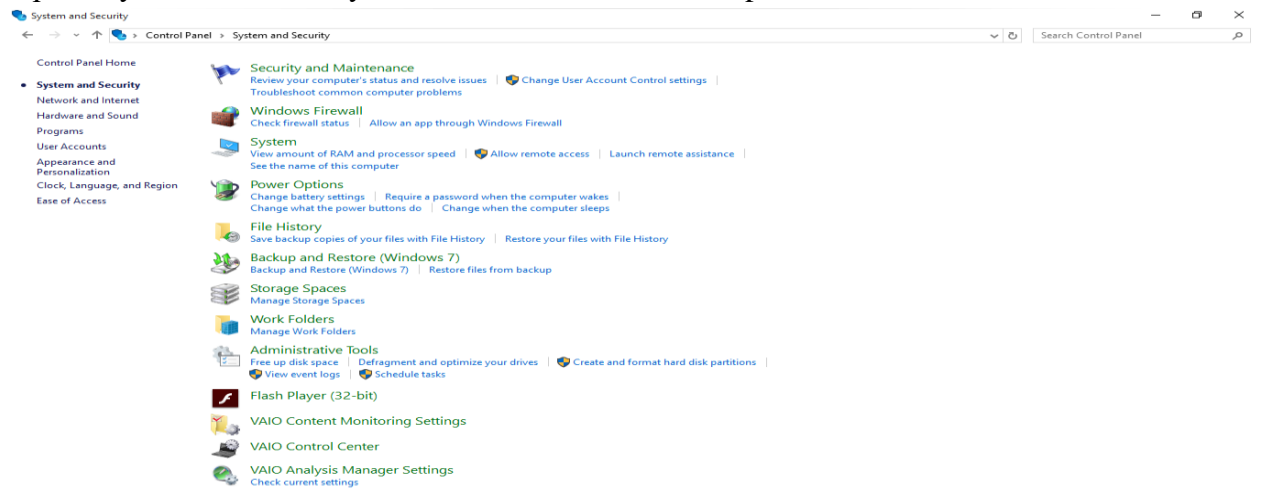
Welcome to the user manual for the PLS (Parking Lot System). Before getting started, it is to be reminded that this application runs on any windows operating System that starts from windows 7 above and it requires java software to run it, therefor, check if you have it or not. If yes than you can keep reading from here, if not, check at the end of the user manual for more information about java and how to install it.

There are many ways of running the application however we will focus on the following:

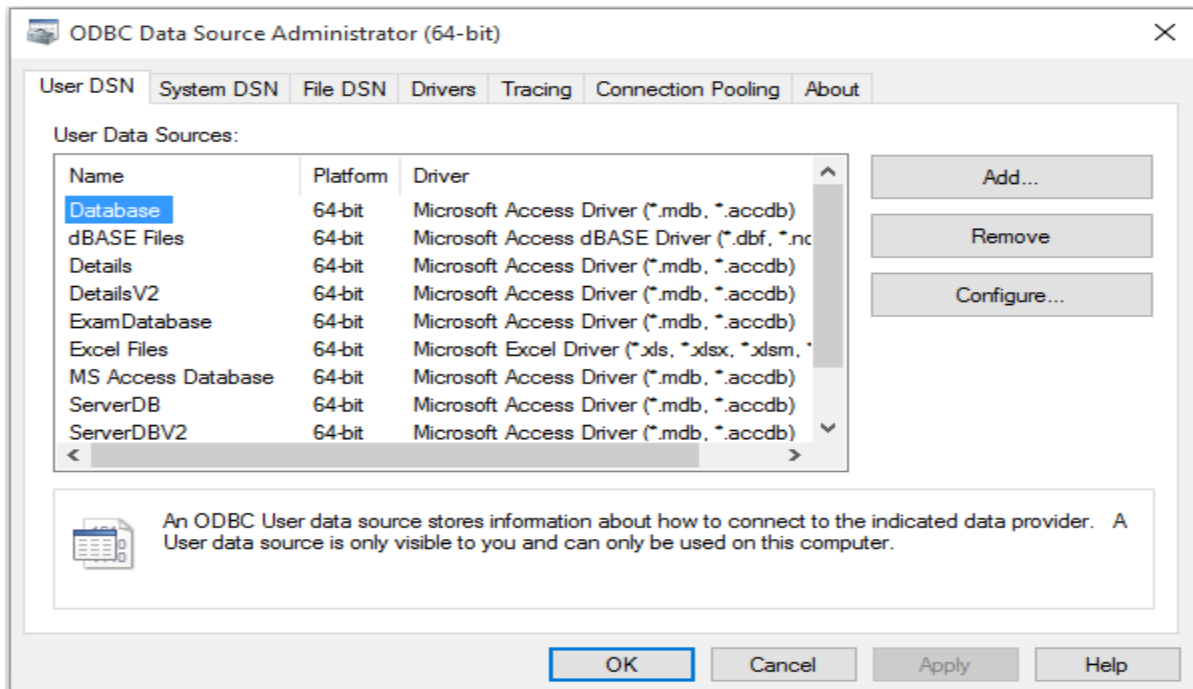
- By use of a NetBeans application

Before trying to run the application, it is required to connect the driver of the database with the computer. To do that, just follow the following order:

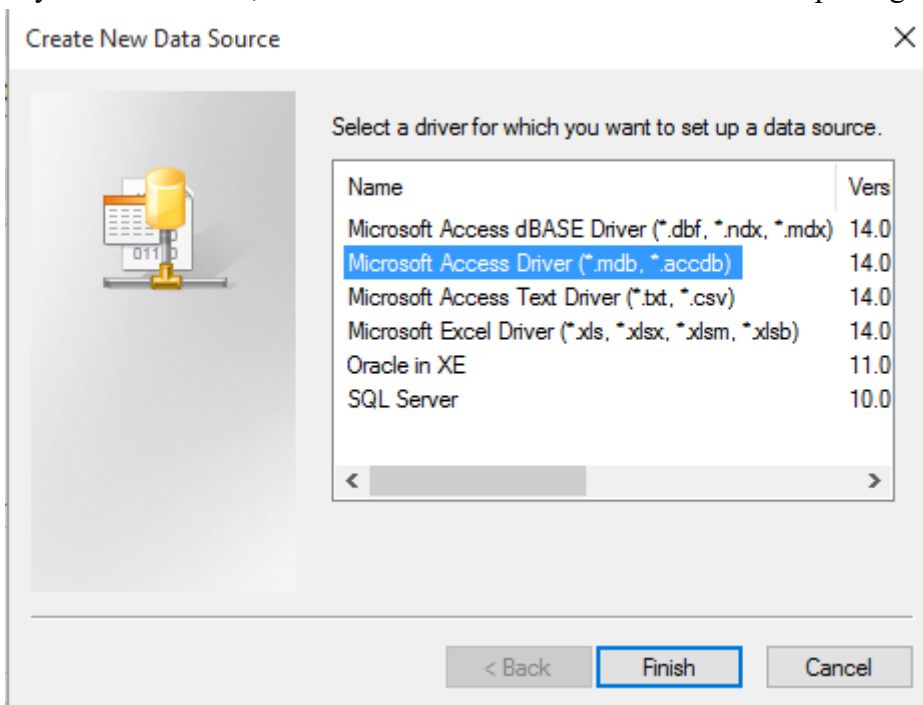
- Go to the start menu and open “Control Panel” (if you don’t know where the Control panel is, then just type it on the search bar and you will find it
- Open “System and Security”, and a screen like this will open:



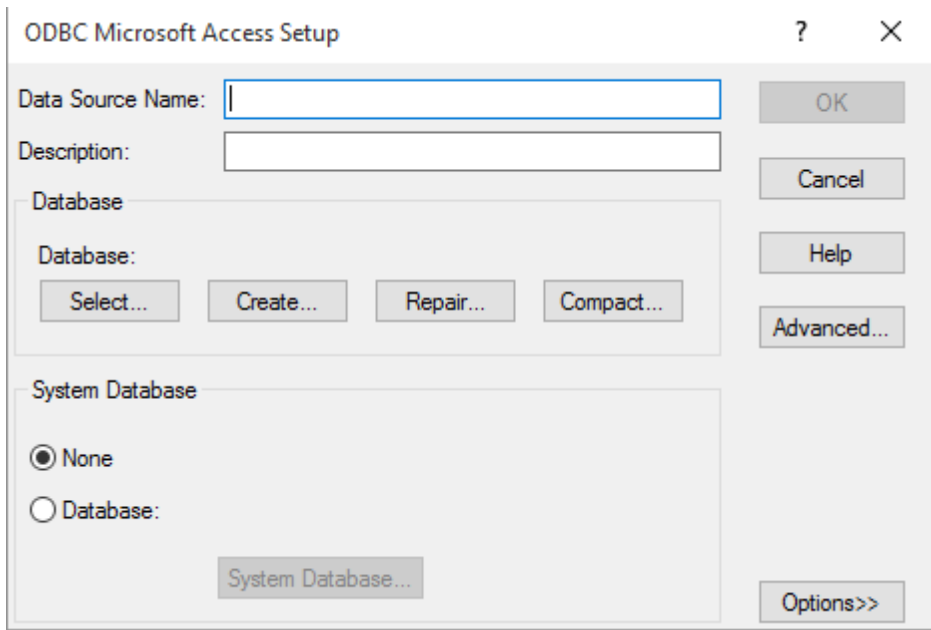
- From there, click on “Administrative Tools”. In Administrative Tools, open “ODBC Data Sources” as Administrator.
- If you have two versions (32 and 64 bits), Open the latest and a Screen like this will appear



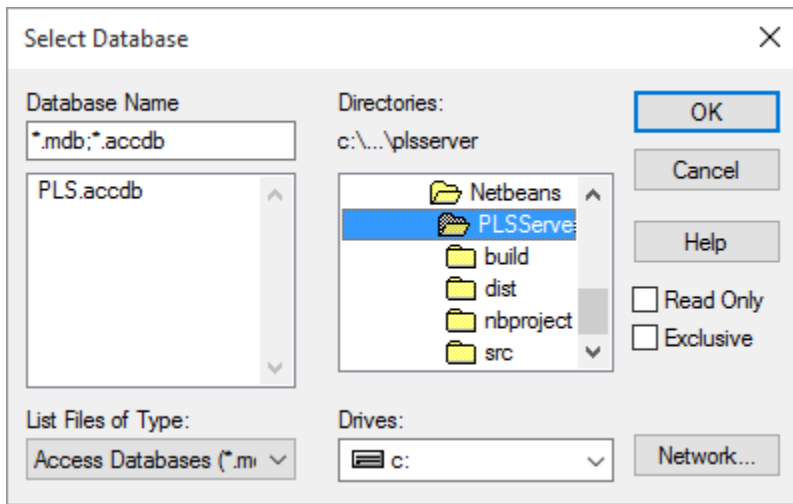
- On the “Tabs”, click on System DSN, and then click on Add. A screen like the one bellow will appear and select “Microsoft Acces Driver(\*.mdb, \*.accdb)”, then klik on finish.
- If you don’t have it, then it is recommended to install an office package.



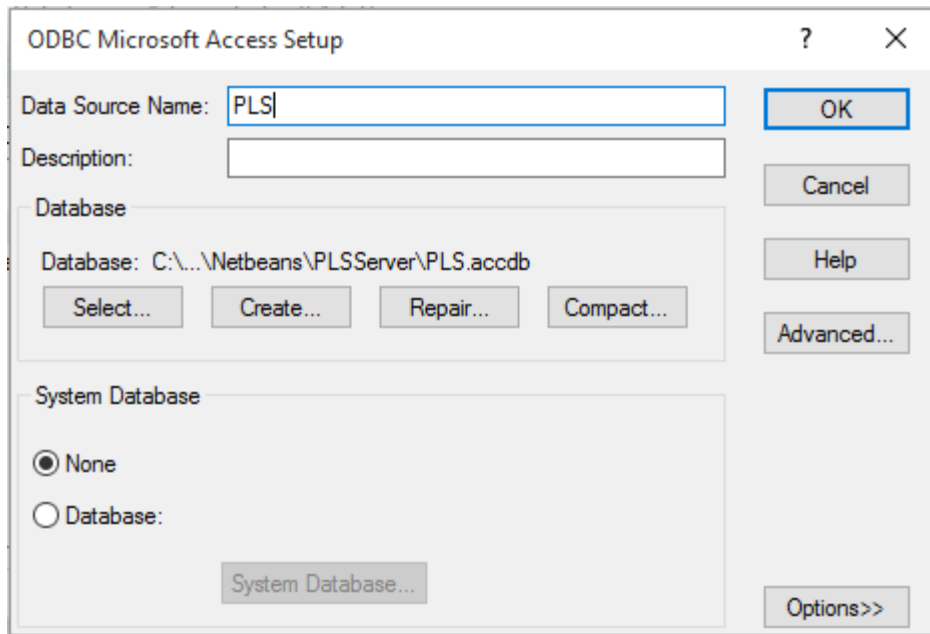
- After clicking on finish, a screen like the one bellow will appear



- After this, click on Select, and a screen to choose the path of the database will appear.



- The database is in the PLSServer folder, go there and select the database that will appear on the left side of the screen (PLS.accdb), and then a screen like the one below will appear.

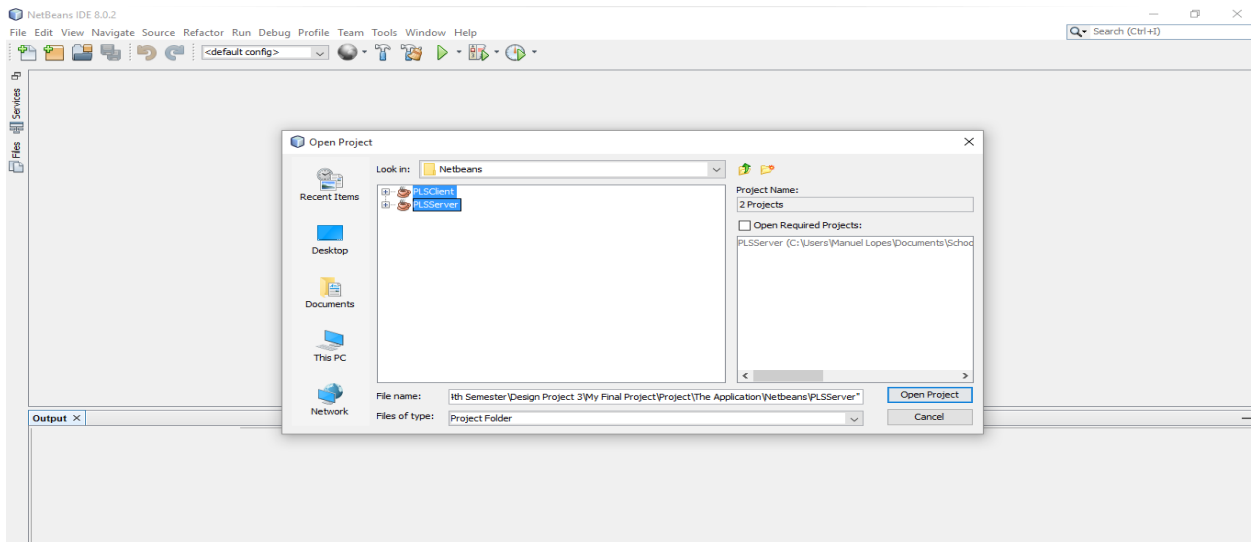


- Name the Data Source name the same as the database name “PLS”, and then click “ok”, and the driver is linked with the computer and can be used anytime whenever called.

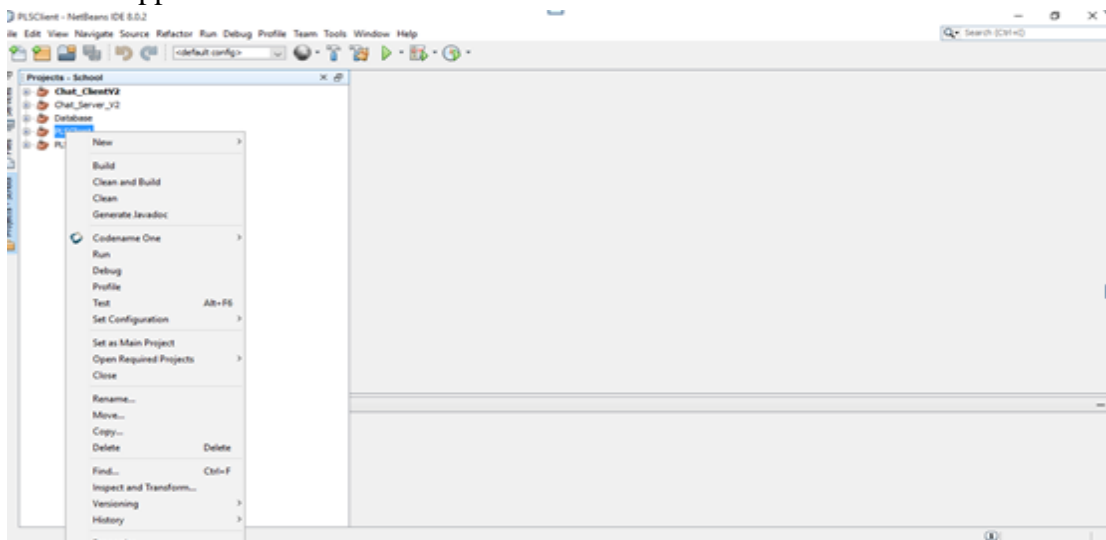
## 10.1 Using NetBeans application

In order to use the NetBeans, it is required that the PC has NetBeans installed, Java and JDKs. If you do not have one of this, then go at the end of the manual to get it explained. If you have everything than:

- Open the NetBeans Application
- Go to file than open project
- Go to the Folder where the application is and then open both (PLSServer and PLSClient)



- Connect Arduino on the PC
- Run the Application.



## 10.2 Using the Application

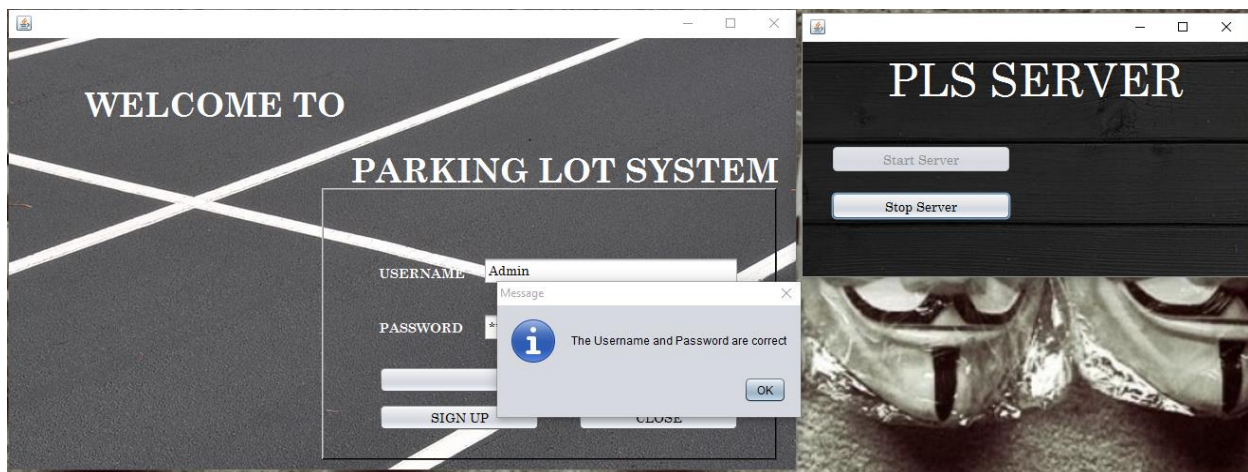
After Running both server and Client, screens like the ones in the figure bellow will appear.

On the left side we have our server and on the right we have the client. Before entering the details on the Login Screen, start the Server first by clicking on the Start Server button.



After Starting the Server, you can input any information on the login Screen. However, you must be a valid user in order to login. Input “Admin” for username and “admin” for password.

Like this, it will access the Administrator Form.



The Figure Bellow shows the Administrator Form.

STAFF_ID	First_Name	LAST_NA...	GENDER	DATE_OF...	ID_NUMB...	PHONE...	ADDRESS	ROAD
1	Admin	admin	Male	9 May 1994	1	614216419	204 Marai...	Marais R
2	k	k	null	null	null	null	null	null
3	Henrique	Niuca	Male	1 January ...	BN1213			
4	Urquidea	Lopes	Female	17 Novem...	NUHJ3			
5	Linea	Monica	Female	7 Novemb...	H983			



In this form, you will be able to add, delete, and update any record.

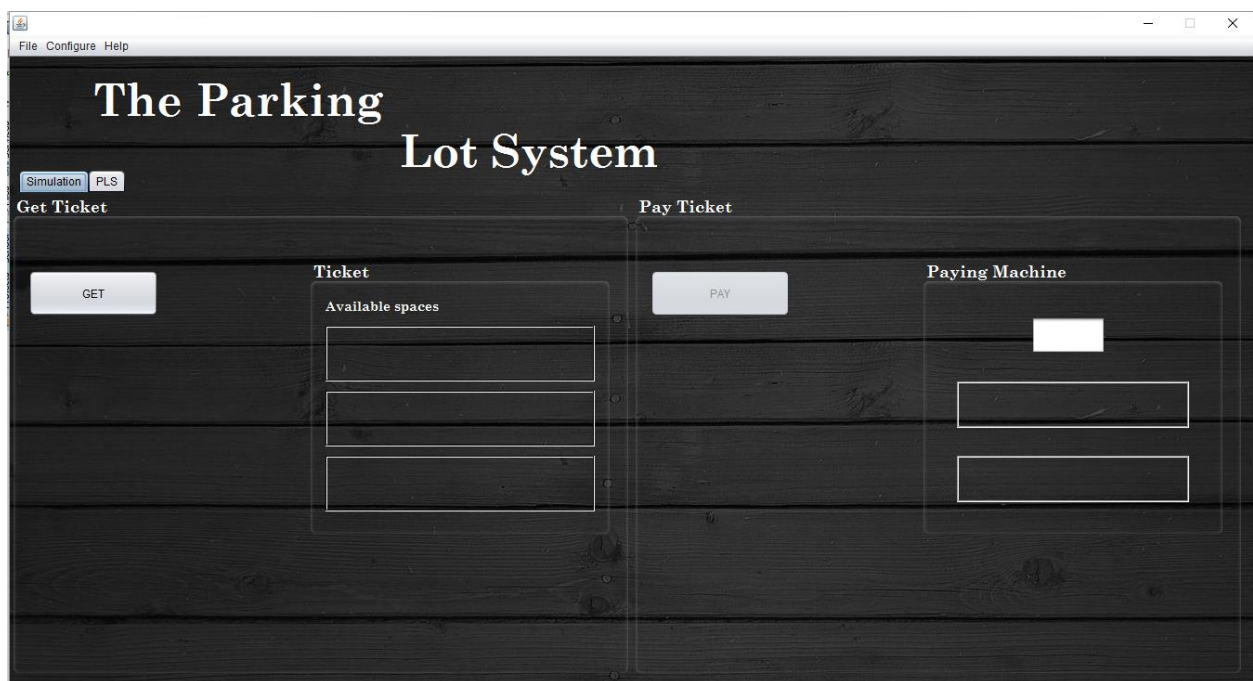
The Left panel shows every record on the AdminTable, the right Panel shows the place where the information will be added.

The Panels bellow is related to management of data. The Panel on left is for searching and refreshing the Table whenever information is added. In other words, after adding a record, click on refresh to update the table.

To delete, or modify, first you need to search the record and then you can modify or delete the record.

After doing everything that you wanted on the AdminForm, you can click on close in order to close the AdminForm.

For The Security, The Username is “Sec” and password is “sec”. After imputing this information in the LoginScreen, the figure bellow will appear.

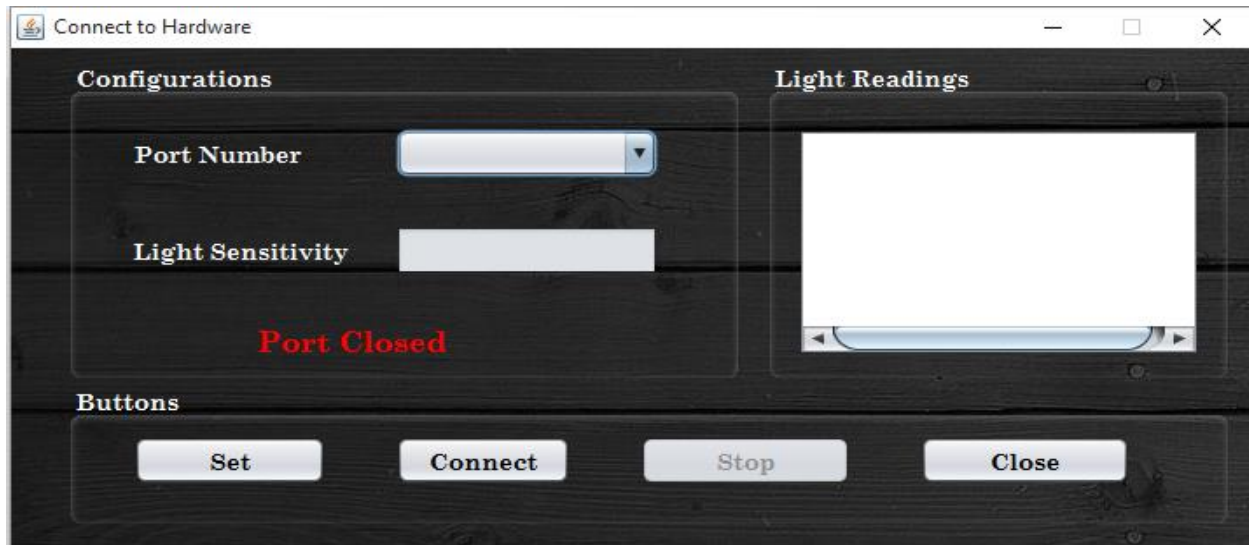


In this Figure, there are 2 actors, the Driver and the Security. As you can see, there are two buttons in this frame, there's the get and there's the pay.

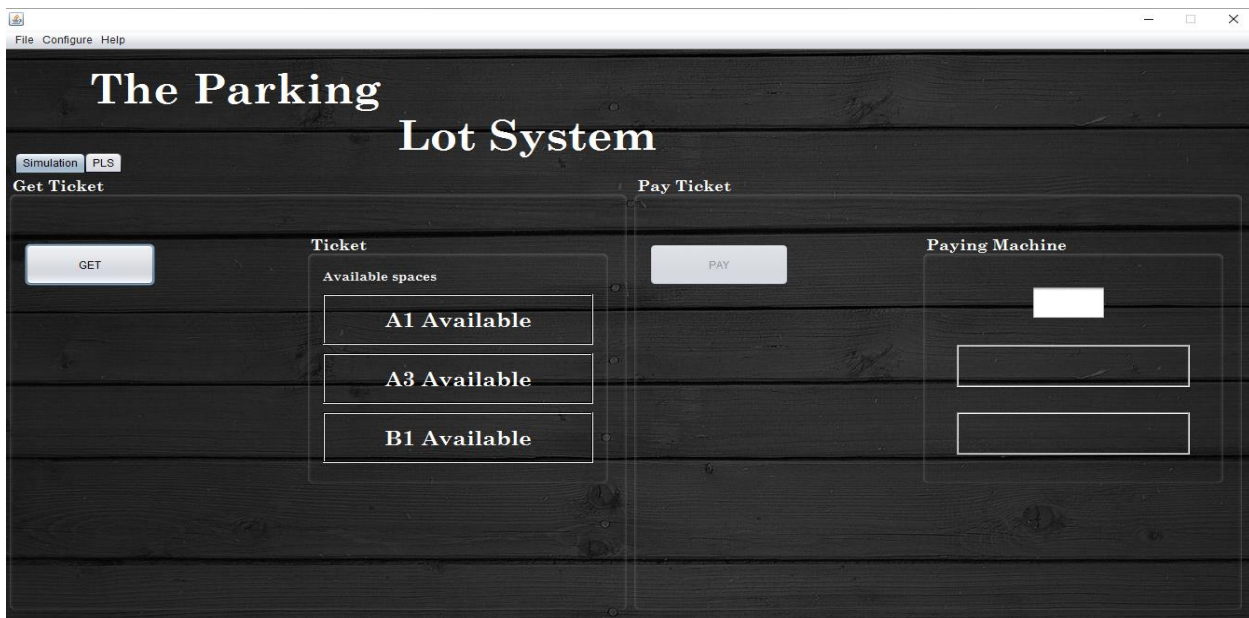
The menu above, there's a menu Configure, that menu will lead to a form where it is required to connect to the Arduino. If clicked, a screen like the one bellow will appear. This screen will have the following information: The port that the port that the Arduino is connected, there is a TextField where you can adjust the light sensitivity, and there's the light reading for you to keep



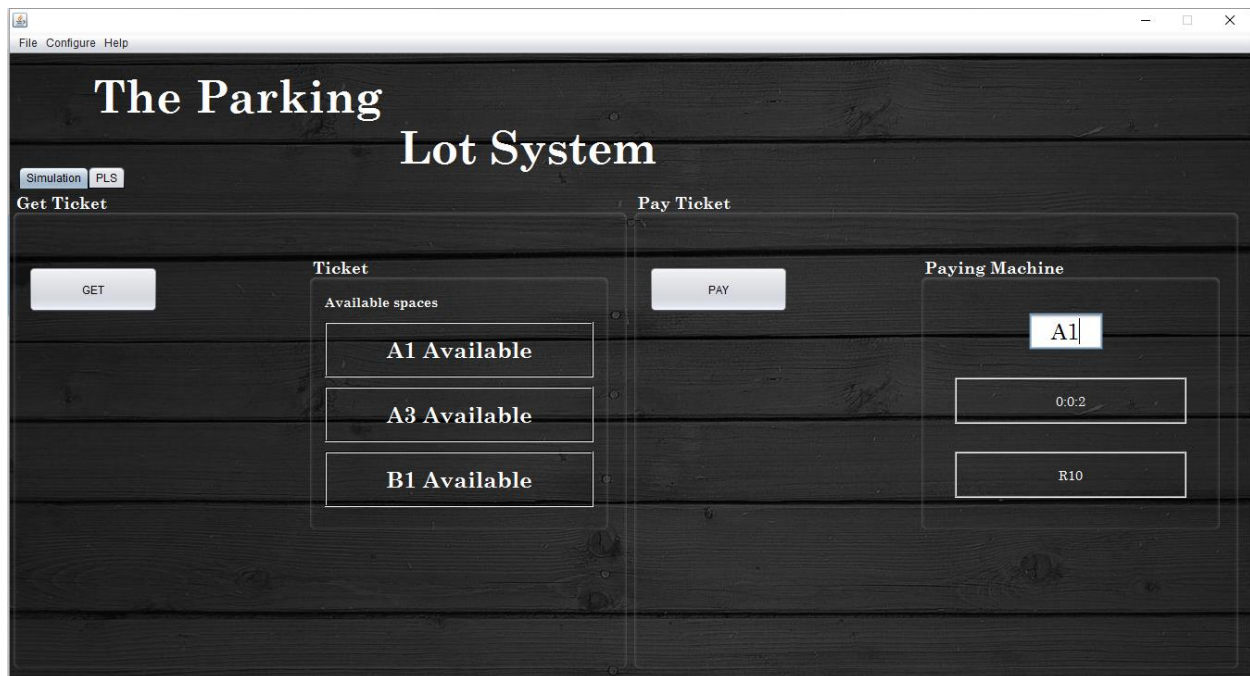
track of the light values. Remember, the value of the light Sensitivity has to be less the the value displayed on the on the readings. Then click on “connect”.



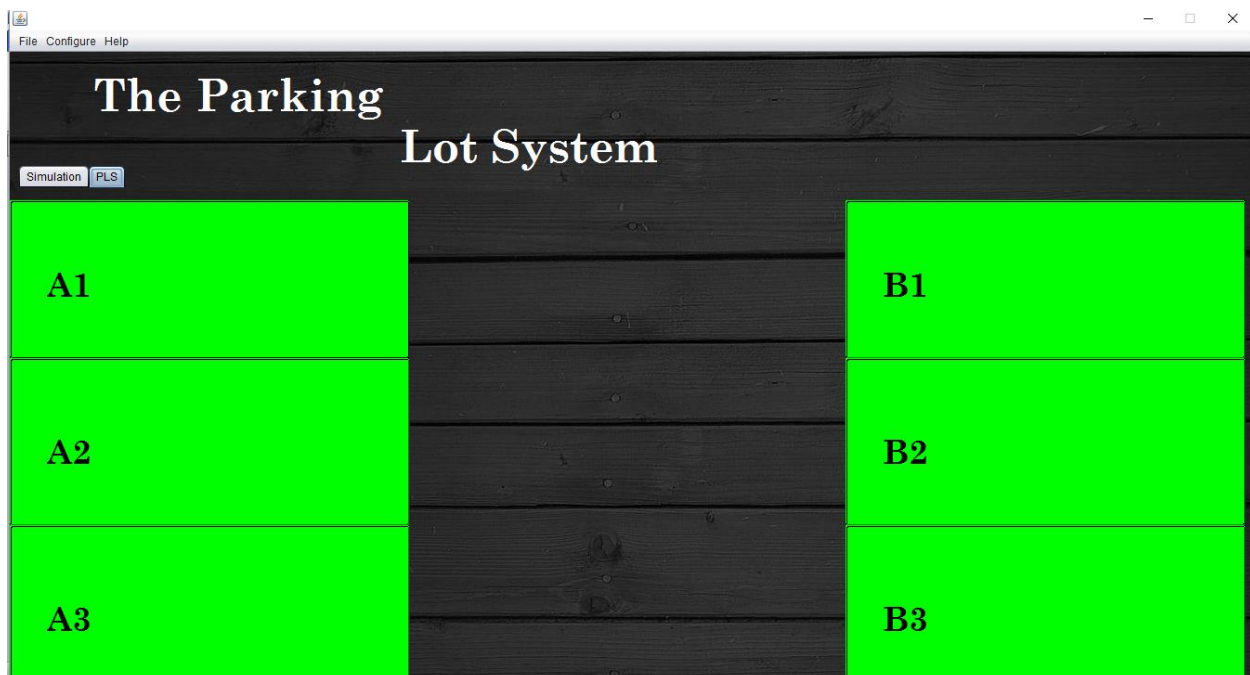
The Get Button will get the tickets and the available spaces will be displayed on the ticket panel right below the “Available Spaces” notice.



On the right Side of the panel is the paying ticket area, in other words, it is there, where the payment of the ticket will be made. After entering the number of the ticket wished to pay, press enter in order to get the time and the amount to pay. After getting the time and the amount you can click on the button pay to pay the amount and then leave the parking lot.



In order to see the parking spaces, you only need to click on the PLS tab and it will display weather is busy or not.



The Figure above shows that the Parking lot is empty.

### **10.3 More Information**

To know how to connect the Arduino on the computer just go to the website:

<https://www.arduino.cc/en/Guide/Windows>

To Download and install java go to the following:

<http://java.com/en/>

To download the JDK go to the following

<http://www.oracle.com/technetwork/java/javase/downloads/jdk7-downloads-1880260.html>

To download NetBeans

## **11 Conclusion**

Resuming and concluding, the PLS or to be more precise, The Parking Lot System, is a system designed for shopping malls, banks and places that people go all the time, and this system, not only helps the drivers with their parking but also the security of the place in which this System is being used. It can be used outside as well but the problem is the light sensors which will keep giving different data.