PALESTINE POLYTECHNIC UNIVERSITY
College of IT and Computer System Engineering
Department of Computer System Engineering

Graduation Project

Vehicle's Smart Parking

Project Team

Mohammad Obido

Muna Abu_areesh

Rana Ali Al_saree

Supervisors

Dr. Mousa Alrefayah

Eng.Yousef Salah

Hebron-Palestine

Dec-2014

Graduation Project

Vehicle's Smart Parking

Project Team

Mohammad Obido

Muna Abu_areesh

Rana Ali Al_saree

Supervisors

Dr. Mousa Alrefayah

Eng.Yousef Salah

According to the system of the College of IT and Computer System Engineering, and to the recommendation of the Project Supervisors, this project is presented to Computer System Engineering Department as a part of requirements of B.Sc. degree in Computer System Engineering.

Hebron-Palestine

Dec-2014

Signatures:

Project Supervisor Signature

……...

……..

Testing Group Signature

……...…..…… …….…….…… ……..……

Department Headmaster Signature

…….

Acknowledgment:

We would like to thank all people who helped us and have direct or indirect contributions in our project.

Our deepest gratitude goes to our supervisors, Dr. Mousa Alrefayah and Eng.Yousef Salah for their enlightening guidance, supports, encouragement and continuous patience throughout the entire period of the project.

Special thanks to our parents, who always encourage, support and care throughout our life. We are also grateful to all our friends in the computer engineering field and in IT and Computer Engineering College in Palestine Polytechnic University.

Abstract:

With the rapid proliferation of vehicle availability and usage on the road in recent years, finding a vacant car parking space is becoming more and more difficult resulting in a number of practical conflicts. Lots of researches and developments are being done all over the world to implement better and smarter parking management mechanisms. In order to alleviate the aforementioned problems, "Vehicle's Smart Parking" system has been developed. With the implementation of the "Vehicle's Smart Parking system" drivers can easily locate and secure a vacant parking space at the parking lot deemed convenient to them in easy and technology-based way. Smart parking is the first step in the right direction to solve many problems in the city leading to urban environment.

We implemented a "Vehicle's Smart Parking" prototype system for parking management to realize the design functionalities and features mentioned. The system will direct and guide drivers to the most convenient available parking space in the parking facility in the shortest amount of time and least effort. The system consists of web and android application on the drivers mobile, which allows the driver to know where the nearest available parking spaces is. The system can display the location of the driver's car on his smart phone through the android application when one comes back to get ones car. In addition, the system allows the driver to reserve a slot in the parking from anyplace where the internet is available.

List of Contents

Chapter Five System Design

Chapter Six System Implementation

List of tables

List of figures

Chapter One

Introduction

1.1 Overview

1.2 Project Description

1.3 Motivation

1.4 Project Scope

1.4.1 System Requirements

1.4.2 System Boundaries

1.4.3 System Input

1.4.4 System Output

1.4.5 Components

1.4.6 Functions

1.4.7 Project Milestones

1.4.8 Project Assumptions

1.4.9 Project Constraints and Limitations

1.4.10 Initial Project Organization

1.5 Description of Later Chapters

1.1 Overview

This chapter introduces the three main topics of our project. We will begin by a brief description about our project's idea. Then we will explain the problems and motivations which lead to the idea of this system. Finally, we will explain the scope of the project.

1.2 Project Description

With the high percentage of vehicle ownership in Palestine, parking has become a conflicting and confusing situation for a number of people. Whether at a bus stops or shopping centers, problems with parking are an everyday occurrence. Drivers usually cannot find parking spaces as easily as they would like. They spend a lot of time, effort, and even fuel trying to find a parking spot. This can hurt local business and decrease the quality of life for residents. In order to deal with the problem of the conurbations, we developed a solution for efficient parking space management. The system guides drivers to the nearest available parking space in an easy and scientific way.

Our "Vehicle's Smart Parking" system can direct and guide drivers to the most convenient available parking space in the shortest amount of time. It's a smart system that shows users where the open slots are. The system will be far better than simply cursing around and hoping to just randomly come across a slot. Smart parking has been developed for a better parking management. It is the first step in solving many parking problems in the city leading to urban environments. At last "Vehicle's Smart Parking" system operates by monitoring the availability of car-parking spaces and making that information available to drivers and facility administrators.

It was planned to use existing components to build a prototype for vehicle's smart parking system. This is based on using presence sensor networks because they allow status to be monitored very accurately for each parking slot. The proposed scheme consists of presence sensor networks, embedded web-server, mobile phone application and cars. In this system low-cost presence sensors network modules are deployed into each parking slot equipped with one sensor node. The state of the parking slot is detected by sensor node and reported periodically to embedded web-server. This information is sent to the smart mobile using Wi-Fi networks in real-time. The driver can find vacant parking slots using android application.

1.3 Motivations

1. The aggravation waste of time and fuel for the drivers looking for parking.
2. The waste of time and fuel contributes a result of traffic congestion.
3. Finding a vacant parking space in a metropolitan area is a daily stress and concern for most drivers.
4. It also imposes significant societal waste, both economically and ecologically.
5. The high cost of parking garages.

1.4 Project Scope

1.4.1 System Requirements:

The proposed project should be able to:

1. Determine the nearest available parking space to the destination that is intended to be reached.

2. Determine the nearest empty slot to the driver from all the empty slots that have been identified and displayed it on an android application.

3. Be easy to use by any driver.

4. Be purchased in a low cost.

5. Collect all information from sensors by using a micro web server.

1.4.2 System Boundaries:

1. By using sensor networks, the system will be able to figure out and specify the unoccupied slots.
2. By using micro web server, the system will be able to collect all information from network sensors and specify where the nearest empty slot from driver.
3. Connection with driver phone through android application to tell him where is the nearest free slot.

4. Mac address of the smart phone will be considered as identification for drivers when they enter the parking.

1.4.3 System Input:

- Drivers demand for the location of vacant space for his car.
- Signals from sensors that indicates the location of vacant space.
- Demand for statistics and data from parking managers.
- Drivers demand for reservation location in the parking.

1.4.4 System Output:

- Locating the vacant space in the parking at the android application on smart phones or tablets at real time.
- Generate reports for parking managers about the current situation of the parking or statistics reports.
- Increase the comfortable of using the parking and decrease time, effort and cost.

1.4.5 Components:

- Presence sensor network module: it collects data and checks parking slot state in real-time then sends parking slot information to embedded web-server.
- Embedded web-server: receive information from presence sensors networks then sends them with the position of parking zone to smart phone.
- Mobile device: connect to micro web server and receive parking slot information from it and the display the real time monitoring of parking slots state in the nearest parking zone.
- Android application: we will program an application using android environment which will be installed at the smart phone.

1.4.6 Functions:

By creating this system we have to be able to program embedded micro web server. Learn how to connect sensors network with micro web server. Also we will use android environment to program mobile application, and to connect the mobile to the server to get the data. We will use an algorithm to find the shortest path to the nearest vacant space as soon as possible. We have to develop a web application for reservation issue.

1.4.7 Project Milestones:

There are many achievements that are expected to be delivered at the end of the project which are:

- Project management plan.
- Project analysis.
- Project design.
- Parking prototype equipped with sensors network and micro web server.
- Android and Web application.

1.4.8 Project Assumptions:

In our project we assume that we have two floors buildings with four parking spots in each floor. Each floor has one entry gate and one exit gate. This project will be applied on a small scale model with fact cars, i.e. module car in wooden parking model.

1.4.9 Project Constraints and Limitations:

We faced some restrictions in applying this project. One of the limitations of smart parking is that we need a sensor at each parking slot, presence sensors network, and smart phone or tablet that supports android operating system. Another limitation of smart parking is to have an installed version of our application with each driver in the parking lot.

1.4.10 Initial Project Organization:

- Project Team:

  - Group of 3 students.

- Stakeholders:

  - Parking managers.
  - Drivers in the parking.

1.5 Description of the Chapters

This section is a brief summary of what each chapter will explain.

Chapter One: Introduction.
In this chapter we will describes the projects idea, objective, and the important of it.

Chapter Two: Literature review and theoretical background.
In this chapter we will discuss the theoretical background and literature review that are related to the main idea of the project.

Chapter Three: Project management plane.
In this chapter we will discuss the project management plane. It talk about task set, project risks, project resource, the cost, and time estimation.

Chapter Four: Software requirement speciation.
In this chapter we will discuss the requirement description of the project. It explains the requirement as scenarios using Use-case diagram, class responsibilities, collaborator, class hierarchies and relationships.

Chapter Five: System design
In this chapter we will discuss the object relational module for software component, state behavioral, and class and object design, the software and hardware interface design.

1.6 Summary

In this chapter we discussed the motivations and boundaries that drove us to crate this system. We also talked about the system how it's built and how it functions. We will talk about the advantages and benefits of this system.

Chapter Two

Literature Review and Theoretical Background

2.1 Overview

2.2 Theoretical Background

2.3 Literature Review

2.4 Summary

2.1 Overview:

In this chapter will describe the theoretical background of vehicle's smart parking and the concept of parking. In addition, we will consider some related works in the literature and compare them with our project.

2.2 Theoretical Background:

This section explains the concept of parking and the techniques used to build vehicle's smart parking system.

Parking is the act of stopping vehicle and leaving it unoccupied, on one or both sides of a road, is often permitted, though sometimes with some restrictions. Parking facilities are constructed in combination with some buildings, to facilitate the coming and going of the buildings' users.[1]

2.2.1 Sensors:

A sensor is a device that detects and responds to some type of input from the physical environment. The input could be light, heat, motion, moisture, pressure, or any one of a great number of other environmental phenomena. The output is generally a signal that is converted to human-readable display at the sensor location or transmitted electronically over a network for reading or further processing.

There are two types of sensors:

- Sensors that are built in to your computer.
- Sensors that are connected to your computer by a wired or wireless connection.

A good sensor obeys the following rules:

- Is sensitive to the measured property only.
- Is insensitive to any other property likely to be encountered in its application.
- Does not influence the measured property. [2]

2.2.2 Microcontrollers:

A microcontroller is a small computer on a single integrated circuit containing a processor core, memory, and programmable I/O peripherals, designed to govern the operation of embedded systems in motor vehicles, robots, office machines, complex medical devices, mobile radio transceivers, vending machines, home appliances, and various other devices, They are programmable, cheap, small, can handle abuse, require almost zero power, and there are so many varieties to suit every need. This is what makes them so useful for robotics; they are like tiny affordable computers.

Some microcontrollers may use four-bit words and operate at clock rate frequencies as low as 4 kHz, for low power consumption .They will generally have the ability to retain functionality while waiting for an event such as a button press or other interrupt.

Other microcontrollers may serve performance-critical roles, where they may need to act more like a digital signal processor (DSP), with higher clock speeds and power consumption.

One important feature on a microcontroller is the I/O ports. Input ports are used for taking in sensor data, while output is used for sending commands to external hardware such as servos. [3]

2.2.3 Android Operating System:

Android is an open source operating system a large community of companies and developers maintain it and contribute toward developing newer versions of it.

The Android Operating System is a Linux-based OS developed by the Open Handset Alliance (OHA), dislodging the later from the number one slot among smart phone OSs, the user interface of android is based on direct manipulation, using touch inputs that loosely correspond to real-world actions, like swiping, tapping, pinching and reverse pinching to manipulate on-screen objects.

Android's source code is released by Google under the Apache Lances this permissive licensing allows the software to be freely modified and distributed by device manufacturers, wireless carriers and enthusiast developers. [4]

Android software development is the process by which new applications are created for the Android operating system, applications are usually developed in the Java programming language using the Android Software Development Kit, but other development tools are available.

The Android Software Development Kit (SDK) includes a comprehensive set of development tools. These include a debugger, libraries, a handset emulator based on QEMU, documentation, sample code, and tutorials. Additionally, developers may use any text editor to edit Java and XML files, then use command line tools (Java Development Kit and Apache Ant are required) to create, build and debug Android applications as well as control attached Android devices .[5]

2.2.4 Wireless USB:

Wireless USB is a short-range, high-bandwidth wireless radio communication protocol created by the Wireless USB Promoter Group. Wireless USB is sometimes abbreviated as "WUSB", although the USB Implementers Forum discouraged this practice and instead prefers to call the technology Certified Wireless USB to distinguish it from the competing UWB standard.

Wireless USB was based on the Wi-Fi Media Alliance's ultra-wideband (UWB) common radio platform, which is capable of sending 480 Mbit/s at distances up to 3 meters (9.8 ft) and 110 Mbit/s at up to 10 meters (33 ft). It was designed to operate in the 3.1 to 10.6 GHz frequency range, although local regulatory policies may restrict the legal operating range for any given country.

Wireless USB is used in game controllers, printers, scanners, digital cameras, portable media players, hard disk drives and flash drives. It is also suitable for transferring parallel video streams, using USB over ultra-wideband protocols.[6]

.2.3 Literature Review:

The following papers and projects are related to our project:

2.3.1 Smart Car Parking System:

Smart car parking system project is a microcontroller based a smart parking system. Usually, in large parking areas we need to search through the whole place for space and then park the car, this project aim to overcome this trouble.

So, the project talk about general parking for all the people, consist of two floors and in each one the park has a place from which the car to enter the park and another to leave it, the output of the system is displayed on LCDs the free places for the car that enter the park, the team use push button switch sensor and connect this sensor to microcontroller to get the number of car in the park .

There will be a screen (LCDs) on the place from which the car enter show if the park is full or have places in it for another car , also show the number of empty places and which places in each floor is empty, so the driver can go to any empty places and then put his car on it, on each slot the team put sensor (switch sensor) that show if the place is empty or available , and display it on the screen.

The project has been implemented by using an 8051 microcontroller, micro-switch sensor, motor, (LCDs) screen, and wire to connect all component together, gate has a motor to open and close it , LCDs to tell the driver which floor number of available space and what the exact location to park the car, so used a microcontroller, many LCDs ,sensor, motor for each gate , and other hardware devices.

In this project the team used three display screen to show the output of the system, the main one to display free spaces and the number of this spaces in the park, and the other two screen to tell the driver the location of these space in the park, and use two gate, 8051 microcontroller, and wired to connect all the component to each other, which make the cost of this project more expensive .

The team enable to control the hardware application connected with pc through a parallel port, enable to send data from switches connected with pc through a parallel port, then analyze the data and send the related voltage signal to the parallel port to run the application related to that signal and then set the specified data on LCDs, and enable to send signal to motor and controlling its movement.

The team failed in programming the 8051 microcontroller and use the parallel port in PC instead of 8051microcontroller which need to use new software technology, they failed in some connection of the component in each of other so have an internal damage in some devices, additionally the cost of the project was so high around 51487$ because the team failed in picking the hardware component. [7]

Differences between this project and our project:

- ✓ We will use android application to display the free slot in parking lot instead of using many LCDs and visual basic interfaces.
- ✓ We will use wireless communication to send data to the android application instead of using wire to connect the screen with microcontroller.
- ✓ Using micro web server to collect all information from sensor instead of using 8051 or PC parallel ports.
- ✓ In our project we compute the cost of time parking.
- ✓ We designed our project in order to help any person parked in a parking slot to find his car.
- ✓ Our project will also help reserve a slot in a parking slot before reaching it.

The main aim of our project is to determine where the available parking space in the parking slot and determine the nearest one from the driver, by using modern technology and suitable hardware component to reduce the cost of the project as much as possible.

2.3.2 Wireless Operated Parking System:

In this project the main idea is to reduce an automated parking system that can control the whole process and collect the fees of the parking service. The idea of this project to enable the driver to communicate with administration center which will control the gates and get the fees from the driver when he is leaving the parking, all of that is done automatically, which will decrease the needed time for parking and increase the accuracy of the payment process, and reduce the role of a human administrator at the park.

This system consist of sensors and counters that monitor the state of the spaces at the parking and feed back the administration center with data about the parking spaces , and this administration center communicate with the driver and opens the gate if there is a space and if the driver is permitted to enter the garage . In other word the driver communicate with the administration center and ask it to enter the garage, the administration center check the identity of the driver and the park status , and the send a signal to gate to open.

The project consist the software that controls the administration center , the gates and sensors , the mobile station , the software that controls the communication process between driver's mobile station and the administration center via communication technique which is Bluetooth and electrical circuit that connect the whole project together. [8]

Differences between this project and our project:

This project guides the driver just to suit floor by sensors at the gate of each floor and counter to count the cars at each floor, but our project detects the accurate position of nearest vacant space, since we have a sensor at each parking slot.

In our project we will use android application as interface with driver that shows where the vacant space, but in this project it counts the cars in each floor and close the gate if the floor is full or open it if there is a space.

2.3.3 Smart Parking:

The system itself consists of a user-interface at the entrance of the parking garage, which allows the driver to view all available parking spaces. Although the system will recommend the slot it deems most favorable, the driver is free to view each floor and select any available slot. Once a slot has been selected, a red LEDs built into the slots will guide the driver to their chosen destination. In addition, a ticket with directions will be printed that can be used upon return to the parking garage to guide the driver to their vehicle.

The visual basic interface on PC must send and receive the appropriate information from the microcontroller. The microcontroller, in turn, must be able to multi-task and have numerous threads running at the same time in order to track multiple cars throughout the garage. The sensors in the road and parking slots must be accurate enough to locate each car in the garage. In order to reliably park each car, these key performance specifications must be met. [9]

Differences between this project and our project:

- ✓ This project used visual basic interface, but in our project we will use android application.
- ✓ This project used vehicles loop detector which connected directly to the micro controller, but in our project we will use presence sensors and WiFi technology.
- ✓ One of disadvantages of this project that the driver deals with central computer which is not suitable as android application on smart phone as in our project.

2.4 Summary

This chapter presented an overview of vehicles smart parking and its related issues. In addition, we discussed some of previous smart parking projects and studies and analyzed their results.

Chapter Three

Project Management Plan

3.1 Chapter Overview

3.2 Project Management Plan

3.2.1 Task Set

3.2.2 Risks

3.2.3 Project Resource and Total Estimated Cost

3.2.4 Time Estimation

3.3 Project Methodology

3.3.1 Options

3.3.2 Analysis of Each Option

3.4 Project Components

3.4.1 Hardware Component

3.4.2 Software Component

3.4.3 Other Component

3.5 Summary

3.1 Overview

This chapter explains the project plan, which includes all the sets of project tasks in addition to the project timeline. It also talks about the methodology of the work during project manipulation. Project methodology consists of two parts, the first part introduces in general the steps of work flow in the project, and the second one states and analyzes the available options that can be used to manipulate the project. In addition, this chapter introduces some of project risks and information sheet for each risk. Finally, it includes an analysis to all the project components with their estimated costs.

3.2 Project Management Plan

This section describes task set, risks, project resources, total estimated cost and time estimation.

3.2.1 Task set

The project activities depend on each other, so in this section we will describe the tasks that will be completed in this semester and next semester.

1. Communication:

- The group communicates with each other to select idea for the project, searching in Internet and library for the idea of project, ask advisor for help to select the idea (T1).
- Identify all the stockholders and the end user of our project, which are the users who can access the system and who will benefit from it.
- Meet the users in order to gather all the needed requirements, also to determine the overall goals of the access control system, and to know exactly how will benefit from this project.

- Determine the scope of our project in order to know in which environment to be in, also to determine all the inputs and outputs of the access control system and all the functions that will perform.
- Final step of communication is modifying the statement of the scope as required, to update the statement of the scope if any change happened.

2. Planning:

Determine the schedule:

- Determine all the resources and the components needed for our project, in order to determine the budget of the project (T3).
- Determine the process model that will be used in working, and we choose the waterfall model because all the work will be done sequentially.
- The deadline will be determined, depending on the budget and functions to be performed (T3).

3. Modeling:

- Analysis: analyze all the requirements in order to design the project correctly, and to make this operation as easy as possible (T3).
- Design: we will design the project in order to achieve the requirements (T4).

4. Construction:

- Coding: java language will be used to implement our project and micro controller programming in order to achieve all the requirements (T6).
- Testing: this step will be done to test the correctness of our code (T7).

5. Deployment:

- Operation: the system will be operated in order to be evaluated from the users.

- Maintenance: we will ask from users a feedback, and we will behave based on this feedback.

The task durations and dependencies are shown in Table [3.1] A and B.

Table 3.1: Tasks Plan

Task No.	Task Name	Time (weeks)	Parallelism	Dependency
T1	Selecting the idea of the project	3		
T1.1	Searching in internet and library for the idea of project	3	T1.2	
T1.2	Ask advisor for help to select idea	3	T1.2	
T2	Collecting information, literature review and related theory	4		T1
T2.1	Study Literature review and theory	4	T2.2	
T2.2	Summery and document Literature review and theory	4	T2.1	
T3	Project analysis	4		T2
T3.1	Time scheduling	2	T3.2 , T3.3	
T3.2	Determine the HW and SW needed	2	T3.1 , T3.3	
T3.3	Determine the cost of the project	2	T3.1 , T3.2	
T3.4	Risk management	1	T3.5	
T3.5	Project methodology	1	T3.4	
T3.6	HW components analysis	1	T3.7	
T3.7	SW components analysis	1	T3.6	
T4	Project design	5		T3
T4.1	Design block diagram	2		
T4.2	Design the software schematic diagram	1.5		
T4.3	Design the hardware schematic diagram	1.5		
T5	Hardware design and implementation	6		
T5.1	System speciation	2		

	Design			
T5.2	Collect the HW Components	1		
T5.3	Implement the sensors and microcontroller	2		T5.1 ,T5.2
T5.4	Connect the micro-controllers with peripherals	1		T5.3
T6	Software implementation and execution	7		T3,T4,T5
T6.1	Collect the samples for training and testing phases	3	T6.2	
T6.2	Perform the train phase using PC ,testing accuracy	3	T6.1	
T6.3	Write the software Program	4		
T7	Test the system	3		
T7.1	Correct error if exist in HW or SW	1.5		
T7.2	Re- execution to sure it run correct	1.5		
T8	Writing documentation	29	T1 ,T2,T3,T4 ,T5,T6,T7,T8	T1,T2,T3,T4, T5,T6,T7,T8

3.2.2 Time Estimation

Figure (3.1) shows time estimated for our project.

ID	Task Name	Start	Finish
1	selecting the project idea	2014/1/30	2014/2/20
2	collecting information about project idea.	2014/2/20	2014/3/10
3	management planing	2014/3/10	2014/3/25
5	project analysis	2014/3/25	2014/4/16
6	project design	2014/4/17	2014/5/29
6	learning and improve out experience	2014/5/29	2014/9/1
7	hardware design	2014/9/1	2014/10/15
8	software design and coding	2014/10/15	2014/12/4
9	testing and modification	2014/12/4	2014/12/18
10	Documentation	2014/1/30	2014/12/31

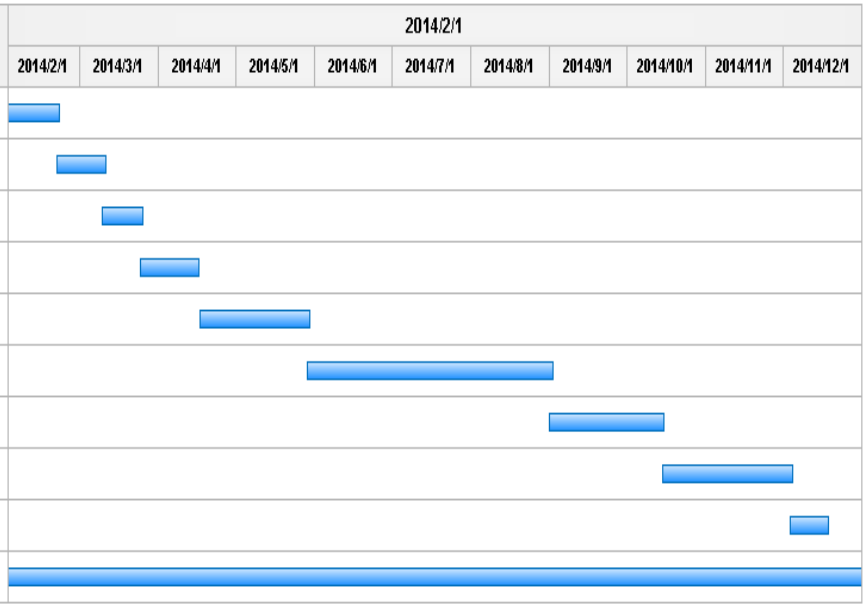

Figure 3.1: Time estimated – Gant chart

3.2.3Risks

Risk is one of the most important factors that must be considered in details and take all the edges of the risk that may occur, in order to avoid them and to ensure a successful project work. In this section we will briefly describe those risks and show their management strategies. Table [3.2] shows risks identification, probability analysis and strategies.

Risk ID	Possible Risks	Probability	Effects	Strategy
R1	Malfunction of hardware parts (Microcontroller, Sensors, smart phone).	Low	Serious	Continue the project development and simulate the network operation.
R2	Undesirable and unexpected software error.	Moderate	Serious	Developing the team experience in programming languages.
R3	May not be receiving the hardware device on time.	Low	Catastrophic	Identify the needed hardware as fast as possible and order them.
R4	Lack of full knowledge of how to deal with the tools.	Very low	Tolerable	Search the web and other references to know how to work with them.
R5	Different kinds of customers and number of users.	Moderate	Tolerable	Intensive interviews with costumers and surveys about the user interest in the system and try to estimate their number.
R6	There may be a significant change in requirements more than we expect.	Low	Serious	Deal with this requirement in a flexible way in order to solve this problem.
R7	Budget is not sufficient.	Moderate	Catastrophic	Determine the project's cost before starting the implementation and Borrow the needed money if this risk occur.
R8	Sensors interrupt by mistake.	high	Serious	Put signs to warn people to avoid standing in front of the sensors or put any things. there
R9	Fall in transmission of data between system components using wireless technology.	Moderate	Serious	Reduce the region that our project covers.
R10	The required time developing the software and the hardware is underestimated**.**	Low	Tolerable	Understand almost the exact time needed in developing the software and hardware.
R11	Code of the software is damaged or deleted suddenly.	Low	Catastrophic	Always save backup copies of the software (on ash, CDs, hard disks).
R12	Change the delivery date of the project.	Very low	Serious	Try to done the job to comply with the specific period.
R13	Shortened in work of any group members for any reason.	Low	Tolerable	Reorganize team so that there is more overlap in the work.

Table 3.2: Risks Identification, Probability, Analysis and Strategies

Here risk information sheet of the risks that affect seriously on the project:

Table 3.3: Unexpected software error information sheet

Risk information sheet			
Risk ID: 02	**Date :** 17/3/2014	**Prob :** 50%	**Impact :** Medium
Description: (undesirable and unexpected software error) May we have unexpected and undesirable error in software programs of our project.			
Refinement / context : **Subcondition1:** Not enough knowledge in programming languages causes software errors**.** **Subcondition2:** Use old version of programs in software programming language.			
Mitigation strategy: 1- Developing the team experience in programming languages. 2- use the new and latest version of programs we need it in our project			
Contingency plan and trigger: 1- Ask help from some persons who have an advanced experience in program language we need it in our project. **2-** Buy new version of programs we need it in our project			
Current status: mitigation steps initiated 22-11-2014			
Originator : Muna		**Assigned :**Mohammad	

Table 3.4: Lack of full knowledge information sheet

Risk information sheet			
Risk ID: 04	**Date :** 17/3/2014	**Prob :** 40%	**Impact :** Medium
Description :(Lack of full knowledge of how to deal with the tools) The team has a lack of experience in dealing with system components that may affect performance of the final project.			
Refinement / context : **Subcondition1:** This system used by different types of people that need user-friendly System design which need experienced application designer.			

Subcondition2: The team does not have enough knowledge about languages will be used in construction of this project like micro controller programming and their packages and android applications. **Subcondition3:** Leak knowledge in presence sensors and micro web server with wireless communications .
Mitigation / monitoring : 1-The team adequate training on presence sensor network design. 2- Team should training on programming language "android application".
Management / contingency plan / trigger : Ask help from teachers and people who are experienced in android applications and presence sensor net work.
Current status: mitigation steps initiated 22-11-2014

Originator : Muna	**Assigned :** Rana

Table 3.5: Different kinds of users knowledge information sheet

Risk information sheet			
Risk ID: 05	**Date :** 17/3/2014	**Prob :** 80%	**Impact :** high

Description: (Different kinds of customers and number of users)

Since this is a system that provides services to different types of users, it may be affected by different modes and requirements of those users.

Refinement / context :

Subcondition1: System's users are simple people and others with different modes and requirements. The number of those users who are interested in this system cannot be determined easily which affect the estimations.

Subcondition2: There is an uncertainty about users' satisfaction of the system and how the user will interact with this system and interactive with android application.

Subcondition3: The driver may not have a smart phone that supports android applications.

Mitigation / monitoring :

1-intensive interviews with costumers and surveys about the user interest in the

system and try to estimate their number. 2-try to make prototypes that enables the users to imagine the final product. 3- State an attractive description of the project that clarifies the importance of this project and its benefits.
Management / contingency plan / trigger : 1-make a balanced estimation of the average number of system users. 2-make friendly-system that will be acceptable in balance. 3-try to put some explanation of our project and the android application on the web.
Current status: mitigation steps initiated 22-11-2014

Originator : Muna	**Assigned :** mohammad

Table 3.6: Budget information sheet

Risk information sheet			
Risk ID:07	**Date :** 17/3/2014	**Prob :** 60%	**Impact :** high
Description: (Budget is not sufficient) lacking funding **,** we face it when apply the project			
Refinement / context : **Sub condition 1:** fund of software and hardware that transfer data like micro web server. **Sub condition 2:** some component we will import it from outside so that's expensive **Sub condition 3:** lack of Associations for supporting small projects.			
Mitigation / monitoring : We will try to support our project by present budget to many Associations like our University			
Management / contingency plan / trigger : - Decreasing cost by reducing number of components. - Search for free alternatives.			

- Search for finance.	
Current status: mitigation steps initiated 22/11/2014	
Originator: Mohammad	**Assigned:** Mohammad

Table 3.7: interrupt the sensor by mistake information sheet

Risk information sheet			
Risk ID:08	**Date :** 17/3/2014	**Prob :** 60%	**Impact :** Medium
Description: interrupt the sensor by mistake Sensors may be interrupted although there is no car there.			
Refinement / context : **Subcondition1:** sensors may be cut by any person. **Subcondition2:** sensors may be cut by any thing is put in front of it.			
Mitigation strategy: -put signs to warn people to avoid standing in front of the sensors or put any things there.			
Contingency plan and trigger: -We may use sensors that detect the presence of the iron only; this may reduce the effect of this risk.			
Current status: mitigation steps initiated 22-11-2014			
Originator : Rana		**Assigned :**Muna	

Table 3.8: data communication information sheet

Risk information sheet			
Risk ID: 09	**Date :** 17/3/2014	**Prob :** 70%	**Impact :** Medium
Description:(data communication) Fall in transmission of data between system components using wireless technology.			
Refinement / context : **Subcondition1:** In our project data communication is widely range over all the parking.			

Subcondition2: Fall in transmission data between micro-web server and android mobile	
Mitigation / monitoring : - Test data communication and then decide the suitable positions of transmitters and receivers for a good communication.	
Management / contingency plan / trigger : -search for alternative technology for data transmission.	
Current status: mitigation steps initiated 22-11-2014	
Originator : Mohammad	**Assigned :** Rana

Table 3.9: Change the delivery date information sheet

Risk information sheet			
Risk ID:12	**Date :** 17/3/2014	**Prob :** 45%	**Impact :** Medium
Description: (Change the delivery date of the project) The degree of uncertainty that the project schedule will be maintained and the product will be delivered on time.			
Refinement / context : **Subcondition1:** Not enough knowledge in estimation may cause an unrealistic end date **Subcondition2:** Ambiguity in the system requirements. **Subcondition3:** A need to an unavailable software and hardware **Subcondition4**: Lack of effective project team integration and project assembling problems			
Mitigation strategy: 3- Define the scope accurately. 4- Create a realistic; achievable schedule some level of risk Analysis is required. 5- Analyze risks and adjust the schedule.			
Contingency plan and trigger: 3- Abandon (give up) some additional features that does not affect the whole system. 4- Increase the team productivity. **5-** Evaluate the basic functional requirements of system.			

Current status: mitigation steps initiated 22-11-2014	
Originator : Rana	**Assigned :**Mohammad

Table 3.10: Loss of team member information sheet

Risk information sheet			
Risk ID: 13	**Date :** 17/3/2014	**Prob :** 70%	**Impact :** Medium
Description:(Loss of team member) The number of team members is three students only, so any loss of members will be so critical.			
Refinement / context : We have two types of loss of team member : 1- Temporary which may come from illness or any other reasons? **2-** Permanent, which may come from death?			
Mitigation / monitoring : - Really this type of risk doesn't have any mitigation strategies.			
Management / contingency plan / trigger : - Permanent absence: Refine the project scope. - Temporally absence: Try to work overtime and intensively to perform the work.			
Current status: mitigation steps initiated 22-11-2014			
Originator : Mohammad		**Assigned :** Muna	

3.2.4 Project Resources and Estimated Costs

This section contains resources and estimated costs of the project, previews the system cost estimation in term of system hardware, system software, and human resources.

1. Hardware Resources

The project estimated costs were calculated and found to be around 485 JD. Table [3.11] provides a detailed list of the components needed for the project.

Table 3.11: Hardware Resources Cost

Project Component Table	**Number Of Component**	**Price/Unit**
photo electric sensor	8	15 JD
Micro-web server	1	60 JD
USB dongle for wifi	1	10 JD
Wooden model	1	200 JD
Smart phone	1	200JD
	Total	485 JD

2. Software Resources

The software resources are summarized in Table[3.12].

Table 3.12: Software Resources Cost

Software	**Cost**
Android application Tools ”java”	Free
Micro web server OS ”Linux”	Free
Web application ”php,HTML,MySQl”	Free

3. Human Resources

The human resources are summarized in Table[3.13].

Table 3.13: Human Resources Cost

Project team	Salary
Mohammad Obido	400 JD
Muna Abu_areesh	500 JD
Rana Ali Al_saree	600 JD

3.3 Project Methodology

3.3.1 Options

There are many alternative options for choosing each component and this section describes each available option.

1. **Detection the vacant space options:**

Option One: Image Processing

The implementation of Smart Parking System using image processing technique is that captured image by video camera sensor technologies will be processed through the micro controller using algorithms in image processing to detect the vacant space in the parking. Micro controller store and update the occupancy status of available parking space. [10]

Option Two: Presence Sensors

Smart Parking Services based on presence sensors that allows vehicle drivers to effectively find the free parking places. The proposed scheme consists of presence sensors network, embedded web-server and mobile phone application. In the system, low-cost sensors networks modules are deployed into each parking slot. The state of the parking slot is detected by sensor node and is reported periodically to embedded micro

web-server in real-time, and also the vehicle driver can find vacant parking lots using standard mobile devices.

2. Collection information from sensors option :

Option One: Microcontroller

A microcontroller is a small computer on a single integrated circuit consisting of a relatively simple CPU combined with support functions such as a crystal oscillator, timers, and watchdog timer, serial and analog I/O etc. Program memory in the form of EEPROM (Electrically Erasable Programmable Read Only Memory) or ROM (Read Only Memory) is also often included on chip, as well as a typically small amount of RAM (Random Access Memory).Microcontrollers are used in automatically controlled products and devices[11].

Option Two: The Server

A server is a system (software and suitable computer hardware) that responds to requests across a computer network to provide, or help to provide, a network service. Servers can be run on a dedicated computer, which is also often referred to as "the server", but many networked computers are capable of hosting servers. In many cases, a computer can provide several services and have several servers running. Servers are computer programs running to serve the requests of other programs. [12]

3. Microcontroller

Option One: Raspberry Pi

The Raspberry Pi is a small, powerful and lightweight ARM based computer which can do many of the things a desktop PC can do. The powerful graphics capabilities and video output make it ideal for multimedia applications such as media centers and narrowcasting solutions. The Raspberry Pi is based on a Broadcom BCM2835 chip. It does not feature a built-in hard disk or solid-state drive, instead relying on an SD card for booting and long-term storage.[13]

Option Two: 8051 Microcontroller

The Intel MCS-51 (commonly referred to as 8051) is a Harvard architecture, CISC instruction set, single chip microcontroller (μC) series which was developed by Intel in 1980 for use in embedded systems.[14]

3.3.2 Analysis of Each Option:

There are many issues that must consider in order to build a functional successful and reliable portable system .That issue involves choice of major component from many options.

1- **Why we choose the presence sensor instead of image processing?**

We choose presence sensors for many reasons such as easy deployment in existing parking slots, and the flexibility to connect the sensors for accurately keeping check of vacant space. And also there are many problems in image processing technique such as:

- We need many cameras to cover all the area of the parking.
- Camera sensitive to illumination so the change in lighting may also affect detection performance
- Image processing suffers from a lack of accuracy and can be affected by environmental.
- Major problem of vision based parking detection systems include shadow effects, occlusion effects, vacillation of lighting conditions and perspective distortion.[15]

2- **Why we choose the Microcontroller instead of Server?**

We choose microcontroller for many reasons:

- They can be very low power, like under 1W with some peak draws, under half an amp, battery and solar power is practical, unlike bigger computers.
- Physical size: a microcontroller with WiFi or Ethernet chip puts extra size for the project.
- Cost: a microcontroller suitable for this could be in the single dollar range, the networking parts almost as cheap.

- Disposable, you can put them in one off projects and if they die, it's not as bad as if an expensive computer does.

3-Why we choose the Raspberry Pi instead 8051?

We choose Raspberry Pi for many reasons:

- Power consumption, the Raspberry Pi draws about five to seven watts of electricity.
- No moving parts, the Raspberry Pi uses an SD card for storage, which is fast and has no moving parts.
- Small form factor, the Raspberry Pi (with a case) can be held in your hand, this means the Pi can be integrated inside of devices.
- Huge community support, the Raspberry Pi has phenomenal community support; support can be obtained quite easily for the hardware or GNU/Linux software that runs on the Pi mainly in user forums, depending on the GNU/Linux distribution used.
- Multiple uses , having the storage on an SD card makes it easy to swap with other SD cards running other GNU/Linux distributions to quickly and easily change the functionality of the Pi.
- Affordable, compared to other similar alternatives, the Pi offers the best specs for the price, at least that I've found. It is one of the few devices in its class that offers 512 MB of RAM and 700 MHZ processor.

3.4 Project Components

3.4.1 Hardware Components

This section shows hardware components of our project:

1. **Presence Sensor:**

The system has sensors network modules which are deployed into each parking slot**,** each sensor checks parking slot state in real-time then sends parking slot information to micro web server. Figure (3.2) illustrates the idea.

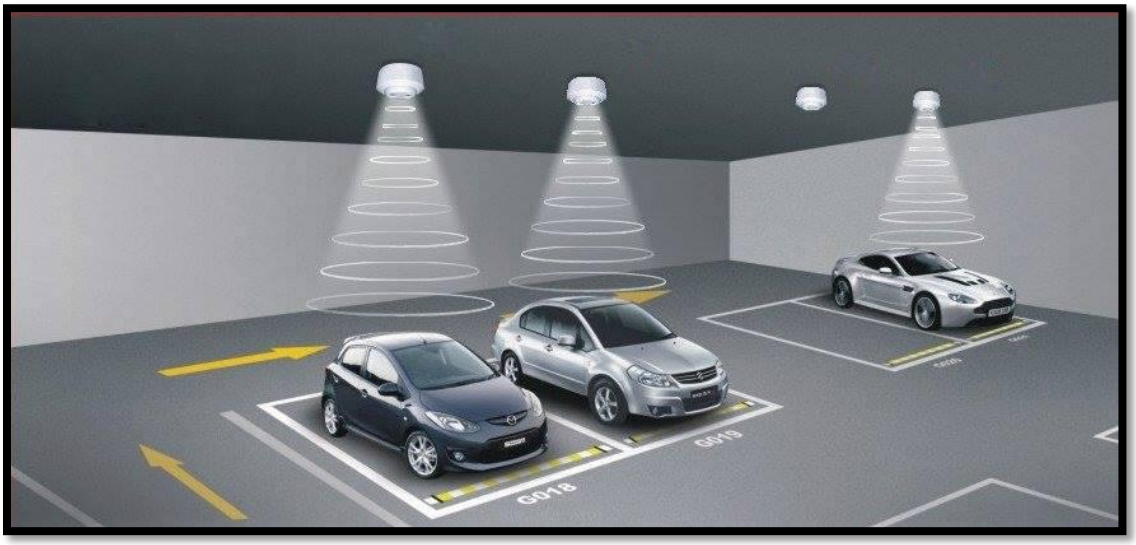

Figure 3.2: presence sensor

The sensor that will be used is photo electric sensor which is a set of transmitter and receiver in one of the photoelectric switch sensor. The detection distance can be adjusted according to the demand. The sensor has a detection range of 3-80cm. Photo electric sensor is small, easy to use, inexpensive, and easy to assemble. Figure (3.3) shows all photo electric sensor components.

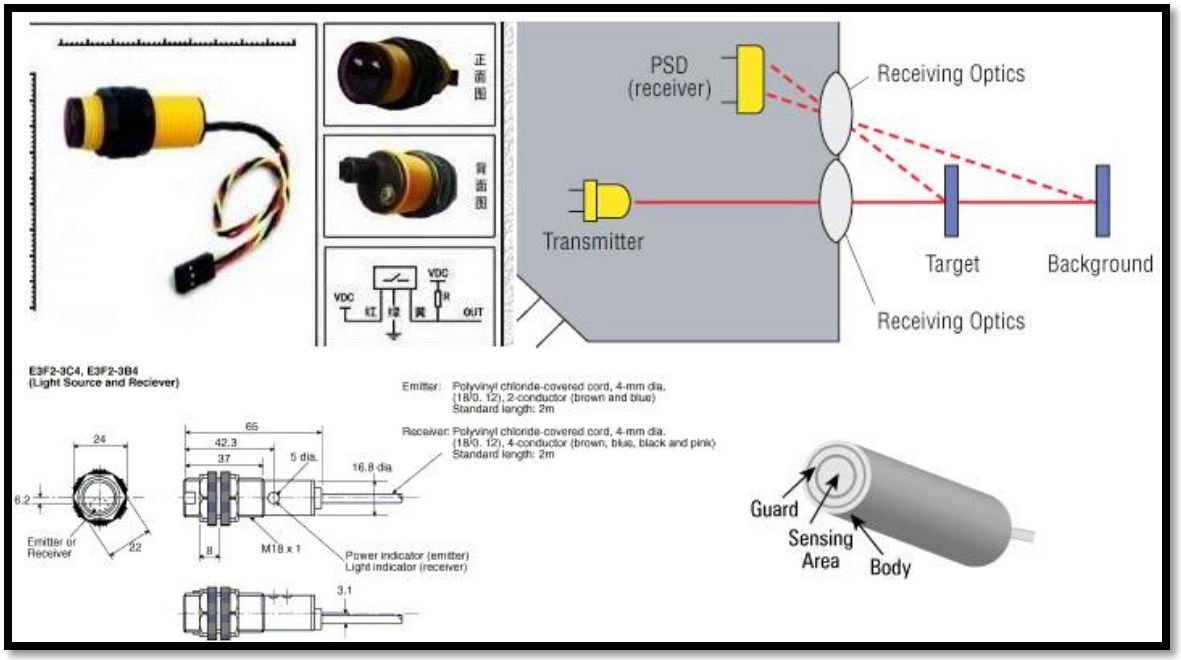

Figure 3.3: photo electric sensor

Sensor specifications:

- Adjustable detection range, 3cm - 80cm
- Small, easy to use/assemble, inexpensive
- Power supply: 5V

Pin out:

- Red - V+
- Yellow - Signal
- Green – GND

Advantages:

- Most accurate.
- Longest sensing range.
- Very reliable.

2. Micro Controller (Raspberry Pi):

The Raspberry Pi is a small, powerful and lightweight ARM based computer which can do many of the things a desktop PC can do. It includes an S 700 MHz

processor and 512 RAM. It uses an SD card for booting and persistent storage. It has also a USB port, an HDMI port and Ethernet port. Figure (3.4) shows all raspberry components.

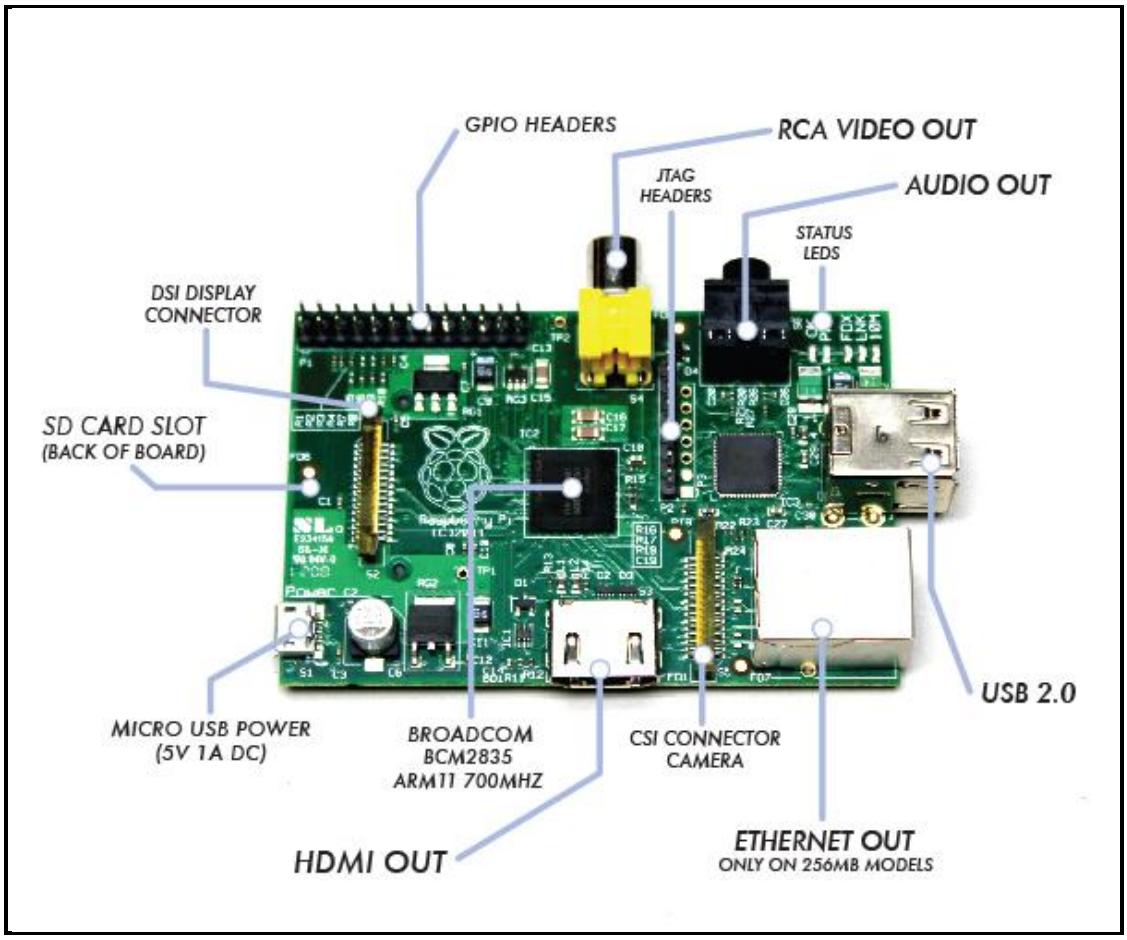

Figure 3.4: raspberry pi

Raspberry pi enables engineers to design their own embedded applications, connect to external devices and facilitate rapid, customized solutions development. [16].

Raspberry pi acts as micro web server in vehicle's smart parking. It collects data from sensors network and sends the result to the android application. Also it saves all information and reports about the parking.

3. USB dongle

It is a cheap USB Wi-Fi adapter and use one of the USB ports to access the wireless network .Setting up raspberry pi OS to access the wireless network.

Take advantage of the raspberry pi USB port to add a low cost, but high-reliability wireless link. These dongles are low power (<100mA nominal draw), so plug directly into the raspberry pi USB ports, requires No External Power. [17] It is used to make a wireless connection between micro web server and android application.

4. Smart Phone

A smart phone is a new-generation high-featured and multi-functional cell phone. Smart phone fuses the functionalities of a handheld computer with the communication capabilities of a cell phone. Modern smart phones include features of a touch screen computer, web browsing and Wi-Fi communication.

In our project we need smart phone that supports android operating system since we have android application which used to guide the driver to the correct position in the parking.

3.4.2 Software Components

1. Android Application

Android is an operating system based on the Linux kernel with a user interface based on direct manipulation, designed primarily for touch screen mobile devices such as smart phones and tablet computers.

Android software development is the process by which new applications are created for the Android operating system. Applications are usually developed in the Java programming language using the Android Software Development Kit (SDK).

The android application in this system is used to display information about parking slot status that received from micro web server to the user in suitable way.

2. Raspberry Pi Software

The Raspberry Pi primarily uses Linux kernel-based operating systems. Raspbian is a Debian-based free operating system optimized for the Raspberry Pi hardware. Raspbian image is written to SD card and then Raspberry boot from this SD card. Raspbian comes preloaded with Python, the official programming language of the Raspberry Pi and IDLE 3, a Python Integrated Development Environment so many more powerful applications can be written using the Python programming language.[18]

3. Web Application

It is an application in which all or some parts of the software are downloaded from the Web each time it is run. It may refer to browser-based apps that run within the user's Web browser, or to "rich client" desktop apps that do not use a browser or to mobile apps that access the Web for additional information.

Our project requires dealing with PHP, HTML and MySQL languages in order to build a web application in this system and store the data on the micro web server in database. This web application allows the user to access the system from anywhere to preserve a free slot in the parking before he reaches the parking. [19]

3.4.3 Other Components

We need other component in our project, to build the parking like wood plate as a model of real parking and some of car models.

3.5 Summary

In this chapter we defined all the project task sets then we generated the project Gantt chart. We briefly described how the work will goes during the project manipulation, and then we stated the available options to use in the project implementation. Some of project

risks was stated and analyzed. Finally we define all the components that are needed for the project evaluation in addition to the costs estimations.

Chapter Four

Software Requirements Specification

4.1 Overview.

4.2 Project Requirement Description.

4.3 CRC “Class Responsibility Collaboration”.

4.4 Class Hierarchies and Relationships.

4.5 Summary.

4.1 Overview:

This chapter represents the requirement description of our project; it explains the requirement scenarios using use-case diagram, class responsibilities collaborator, and class hierarchies and relationships.

4.2 Requirement Description:

This section explains the requirements of the system; represents it as scenarios, and using use-case diagram to show that.

4.2.1 System Actors:

- **Driver:**

Driver can initiate some use-cases in the system. These use cases are:

1. The driver check smart phone to find the nearest vacant space when he enters the parking.
2. When driver come back to the parking he checks smart phone for cost and the place of his car.
3. The driver can reserve location in the parking from anywhere by using web application through the internet.

- **Parking Manager:**

1. Get report of information and statistics of the parking.
2. Set the cost of parking during a specific time period.

- **Sensors:**

1. Detect the vacant space in the parking.

4.2.2: Use-Case Templates:

This section shows detailed description of all the use case in project.

Table [4.1] shows searching for vacant space use-case.

Table 4.1: Searching for vacant space use-case.

Use Case: • Searching for vacant space.
Primary Actor: • Driver.
Goal: • Searching for the nearest vacant space in shortest time and least possible effort.
Precondition: • Installed android application at driver's smart phone. • Installed presence sensors at each parking slot. • Wireless communication between micro web server and smart phone.
Trigger: • Driver wants to park his car.
Scenario: • Driver enters the parking and checks his smart phone. • The system takes MAC address of smart phone and save it in database. • Sensors detect the vacant space and send this data to the micro web server which in its turn sends data to smart phone which convert these data to suitable understandable form.
Exception: • Error in wireless communication between smart phone and micro web server. • Smart phone may be fault.

Table [4.2] shows checking the cost of parking use case.

Table 4.2: Check the cost of parking use-case.

Use Case: • Check the cost of parking.
Primary Actor: • Driver.
Goal: • Check the cost of parking to pay it after exit from the parking.
Precondition: • MAC address of smart phone was taken at the entrance of the parking and stored in database, then time counter start.
Trigger: • Driver wants to pay after exit from the parking.
Scenario: **1**-Driver enters the parking then the system detect the time of entering. **2-** The counter count the time from entering the parking until exit. **3-**When the driver exit, he check his smart phone to see how long he parks his car in the parking and how much the cost.
Exception: • Smart phone may be fault.

Table [4.3] shows detection the location of the car use case.

Table 4.3: Detect the location of car use-case.

Use Case: • Detection the location of car.
Primary Actor: • Driver.
Goal: • Detection the location of the car when the driver comes back to the parking to take his car.
Precondition: • MAC address of smart phone was taken at the entrance of the parking and stored in database with car location.
Trigger: • Detection the location of the car in the parking.
Scenario: • The driver checks his smart phone when he exits from parking. • Display the location of his car on the smart.
Exception: • Smart phone may be fault.

Table [4.4] shows reserve location in the parking use case.

Table 4.4: Reserve location in the parking use-case.

Use Case: Reserve location in the parking.
Primary Actor: Driver.
Goal: • The driver can reserve location in the parking remotely and revoke it if is not occupied within 30 minutes.
Precondition: • Web application installed. • Internet access**.**
Trigger: • The driver can reserve location in the parking**.**
Scenario: • The driver checks his smart phone to see where the location of slot which he reserved from web. • The driver can reserve location in the parking. • If location is not occupied within 30 minutes then revoke the reservation.
Exception: • Smart phone may be fault.

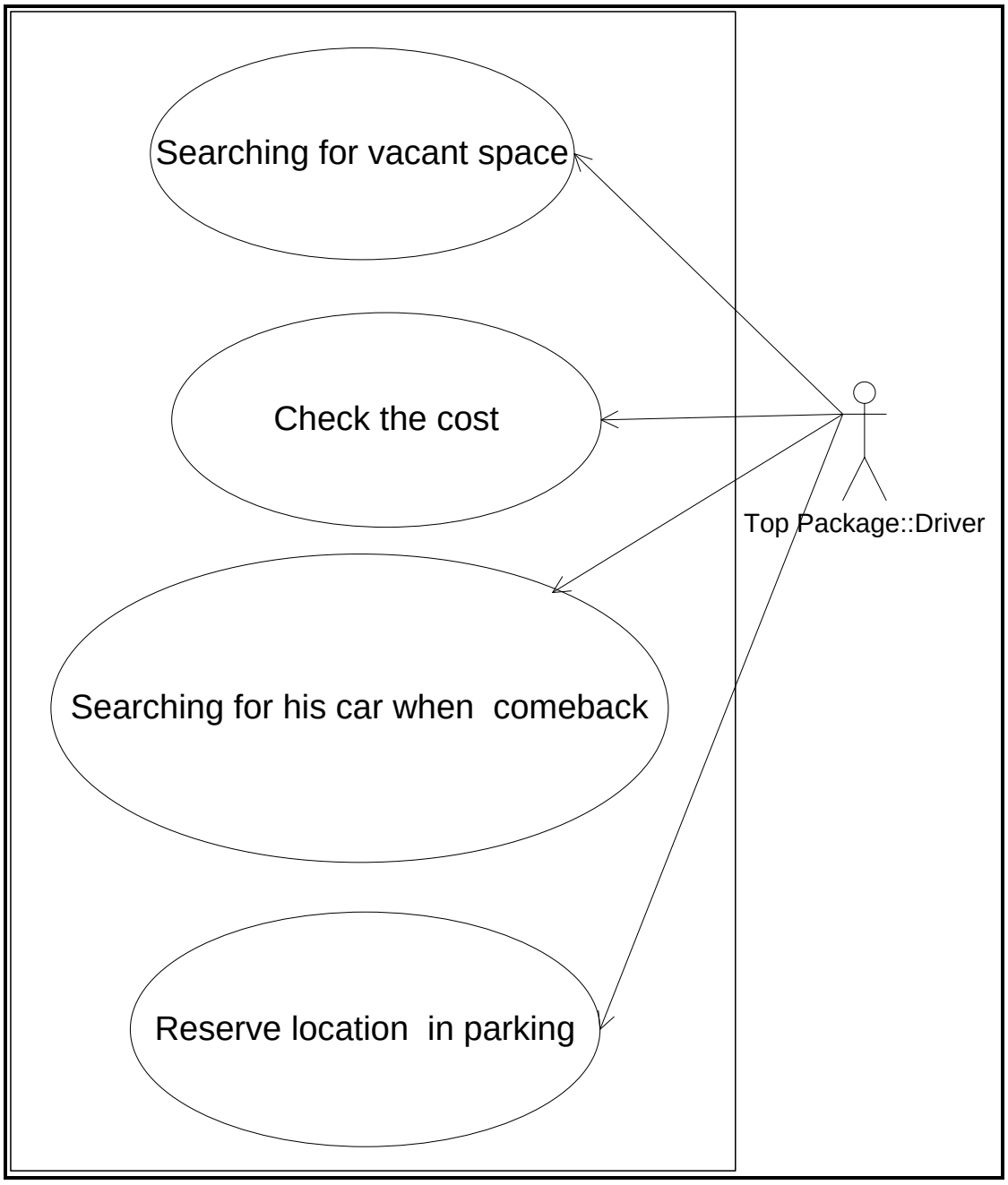

Figure 4.1: Use case diagram for driver

Table [4.5] shows gets reports and statistics use case.

Table 4.5: Get reports and statistics use case.

Use Case: • Get reports and statistics.
Primary Actor: • Parking manager.
Goal: • Parking manager should take information about the parking to develop it and determine the cost.
Precondition: • Data base to manipulate the information and data to classify them. • Installed screen with micro web server to view information and manage the parking.
Trigger: • Parking manager wants to take statistics information about the parking.
Scenario: • Parking manager need check reports for the parking. • All of this information will be available on screen.
Exception: -No exceptions.

Table [4.6] shows set the cost use case.

Table 4.6: Set the cost use case.

Use Case: • Set the cost.
Primary Actor: • Parking manager.
Goal: • Parking manager set the cost of parking during a specific time period.
Precondition: • Manger has an access to the system. • Interface between manager and the system"screen that connected to the micro web server".
Trigger: • Set the cost or change it from time to another
Scenario: • Manager log in to the system. • Manger can set or update the cost.
Exception: • No exceptions.

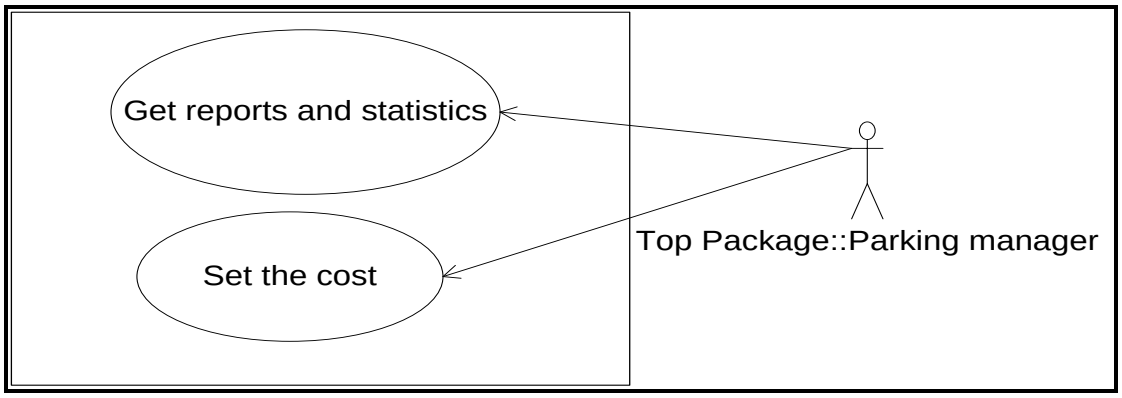

Figure 4.2: Use case diagram for manager

Table [4.7] shows vacant space detection use case.

Table 4.7: Vacant space detection use case.

Use Case: • Vacant space detection.
Primary Actor: • Sensor.
Goal: • Finding the free slot in the parking.
Precondition: • Installed presence sensors at each parking slot.
Trigger: • Detect the location of free slots.

Scenario:

- Driver enters the parking and checks the application on his smart phone.
- Sensors detect the free slots and send this data to the micro web server which in its turn sends data to smart phone which convert these data to suitable understandable form.

Exception:

- Error in communication between sensors and micro web server.

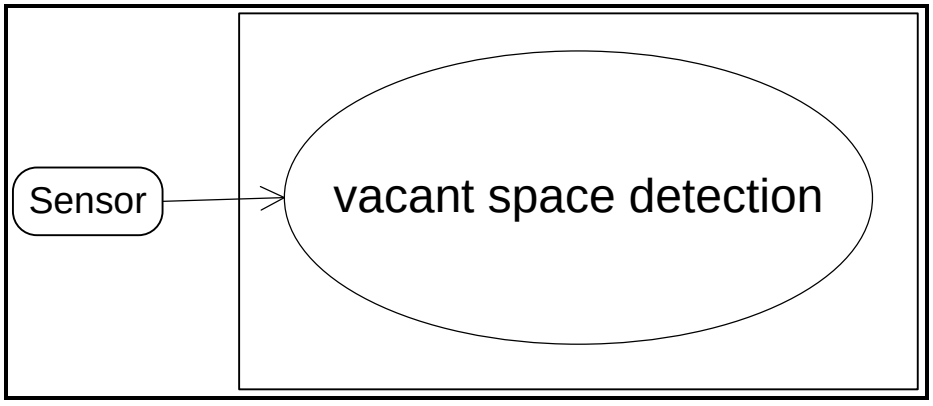

Figure 4.3: use case diagram for the sensor

4.2.3 Use-Case Diagram:

Figure (4.4) shows Use case diagram for the parking system.

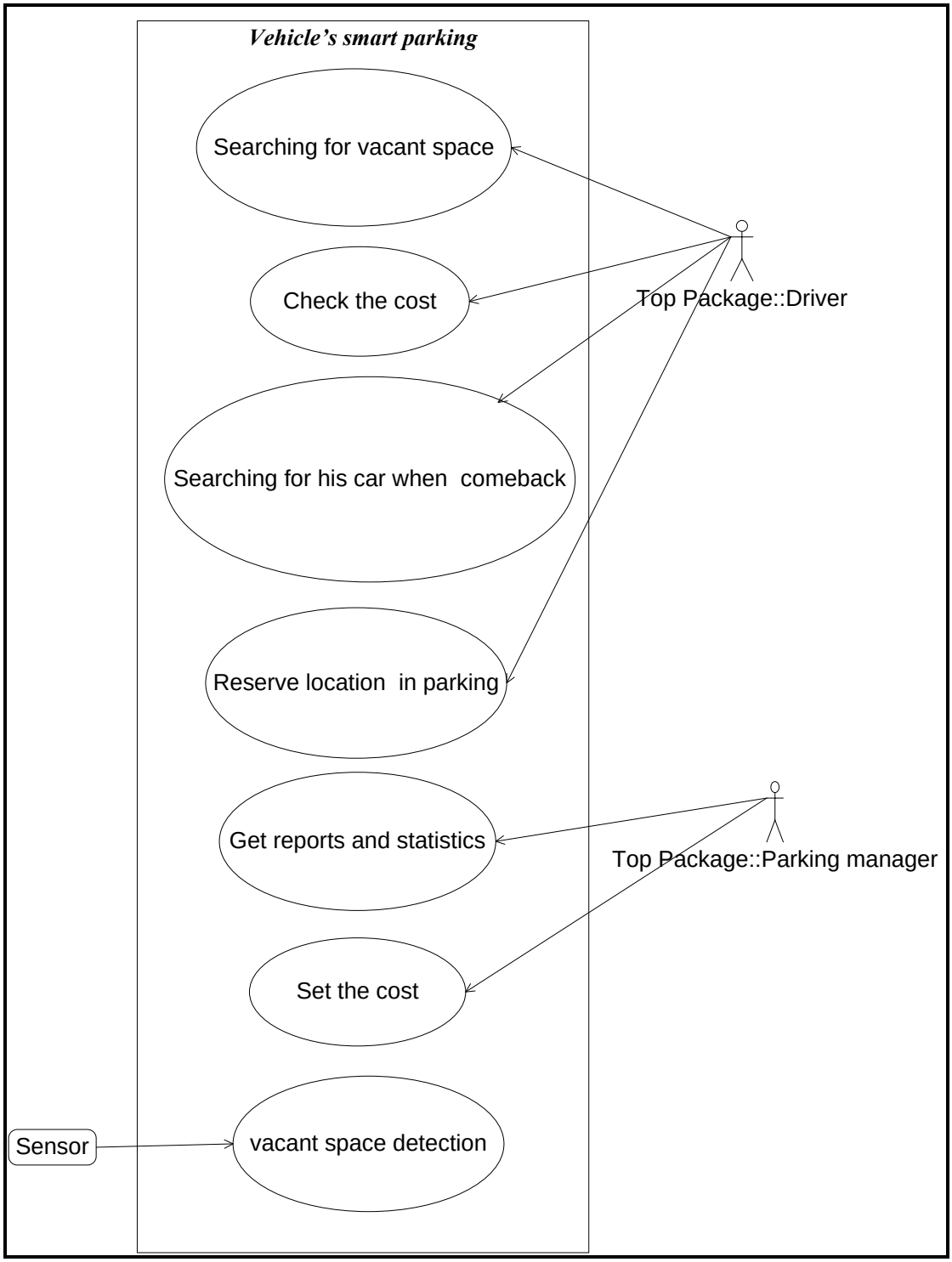

Figure 4.4: use case diagram for the system

4.2.4 Activity Diagram:

This section is shows activity diagram for all the system.

Figure (4.5) shows activity diagram for searching and detection for vacant space.

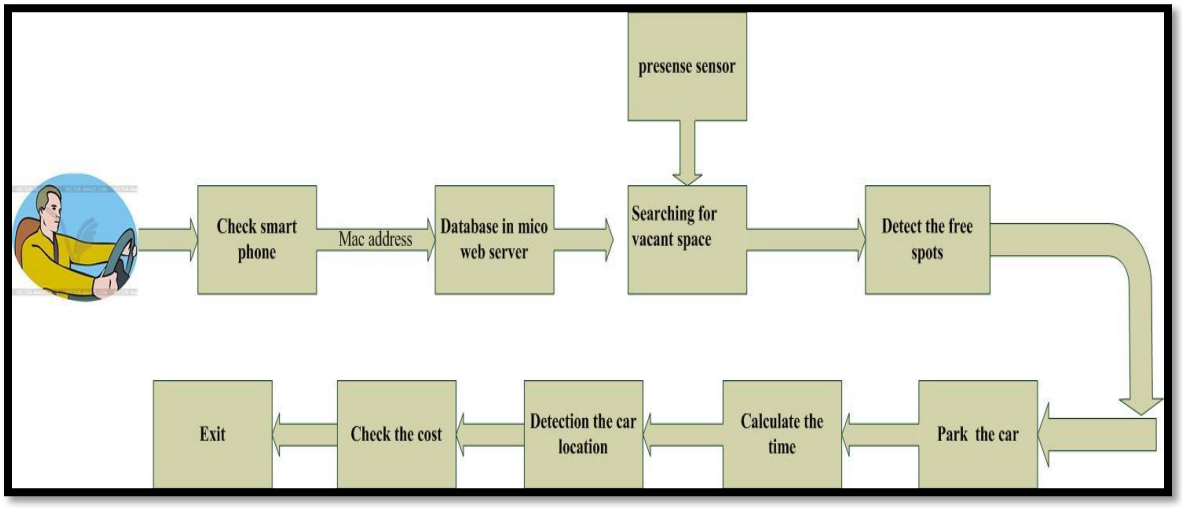

Figure 4.5: Activity Diagram

4.3 Class-Responsibility-Collaborator Modeling (CRC):

This section shows the class responsibilities collaborator, the CRC provides a simple means for identifying and organizing the classes that are relevant to system requirements.

Table 4.8: CRC model for sensor class

Class : Sensor	
Responsibility:	**Collaborator :**
Sensing the presence of a car in slot ( )	
Sends information to the micro-web server( )	Micro-web server

Table 4.9: CRC model for micro-web server class.

Class : Micro-web server	
Responsibility:	**Collaborator :**
Collect information from sensors and store data ( )	Sensors
Send information to the android application. ()	Android application

Table 4.10: CRC model for android application class.

Class : Android Application	
Responsibility:	**Collaborator :**
Display the vacant space on parking map ( )	Driver
Display the cost ( )	Driver
Detect the location of car( )	Driver

Table 4.11: CRC model for driver class.

Class : Driver	
Responsibility:	**Collaborator :**
Check the mobile for vacant space( )	Android application
Check the mobile for the cost of parking( )	Android application
Detecting the location of car ( )	Android application

Table 4.12: CRC model for parking manager class.

Class : Parking Manager	
Responsibility:	**Collaborator :**
Get report of information and statistics of the parking( )	Micro-web server
Set cost()	Micro-web server

4.4 Class Hierarchies and Relationships:

Class hierarchies and relationships; show the relation between all classes in the system as shown in Figure (4.5).

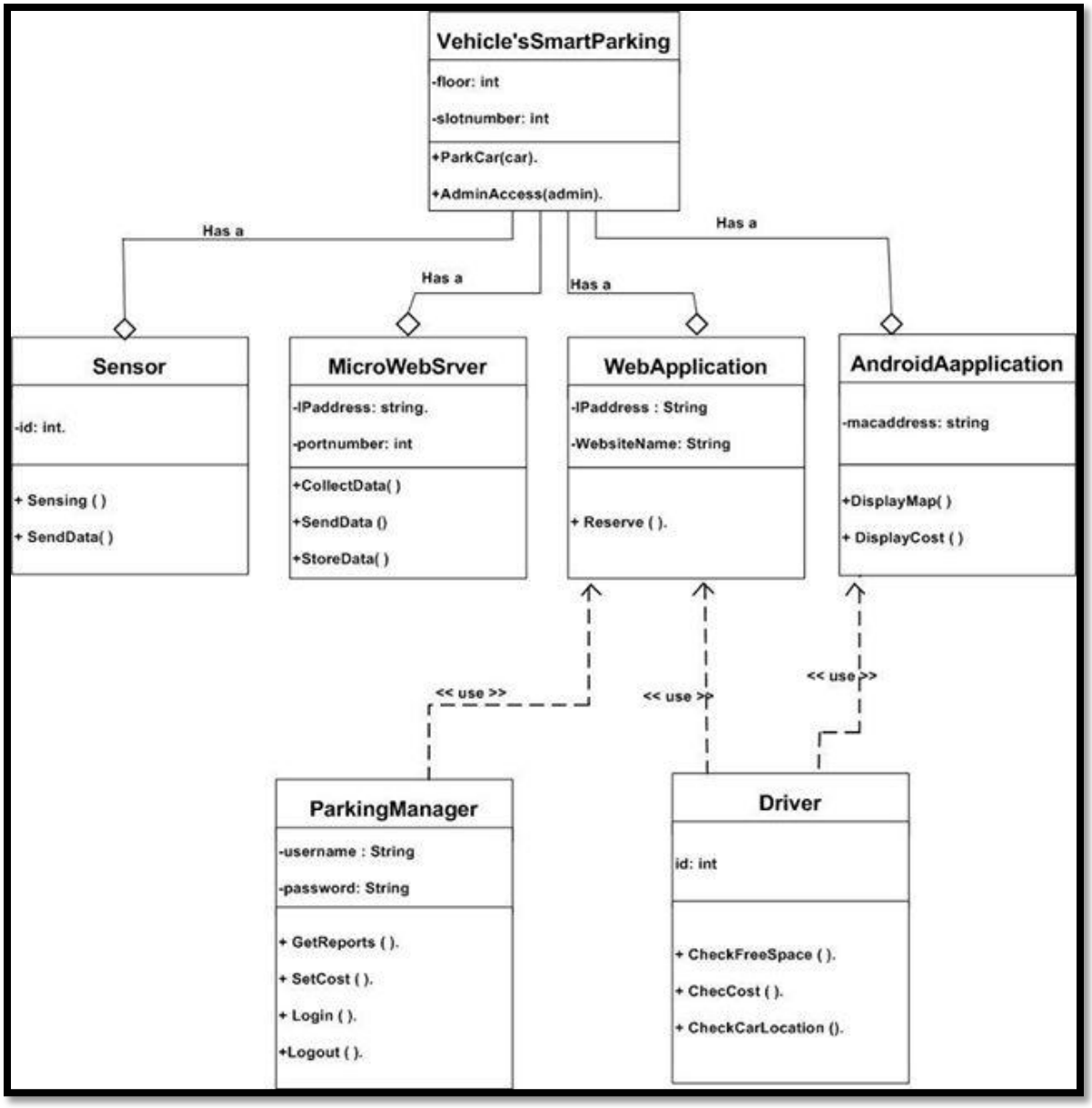

Figure 4.6: class hierarchies

4.5 Summary:

This chapter explained the requirements description of our project; it explained the requirements as scenarios using use-case diagram, class responsibilities collaborator, and finally the chapter represented class hierarchies and relationships.

Chapter Five:

System Design

5.1 Overview

5.2 Object Relational Model

5.3 State Behavioral Modeling

5.4 Subsystem Design

5.5 Class and Object Design

5.6 Object Interfacing

5.7 Hardware Interface Design

5.8 Summery

5.1 Overview:

In this chapter we will define the components, modules, interfaces, and data for the system to satisfy specified requirements.

The basic system abstraction diagram and how components are connected together are shown in Figure (5.1).

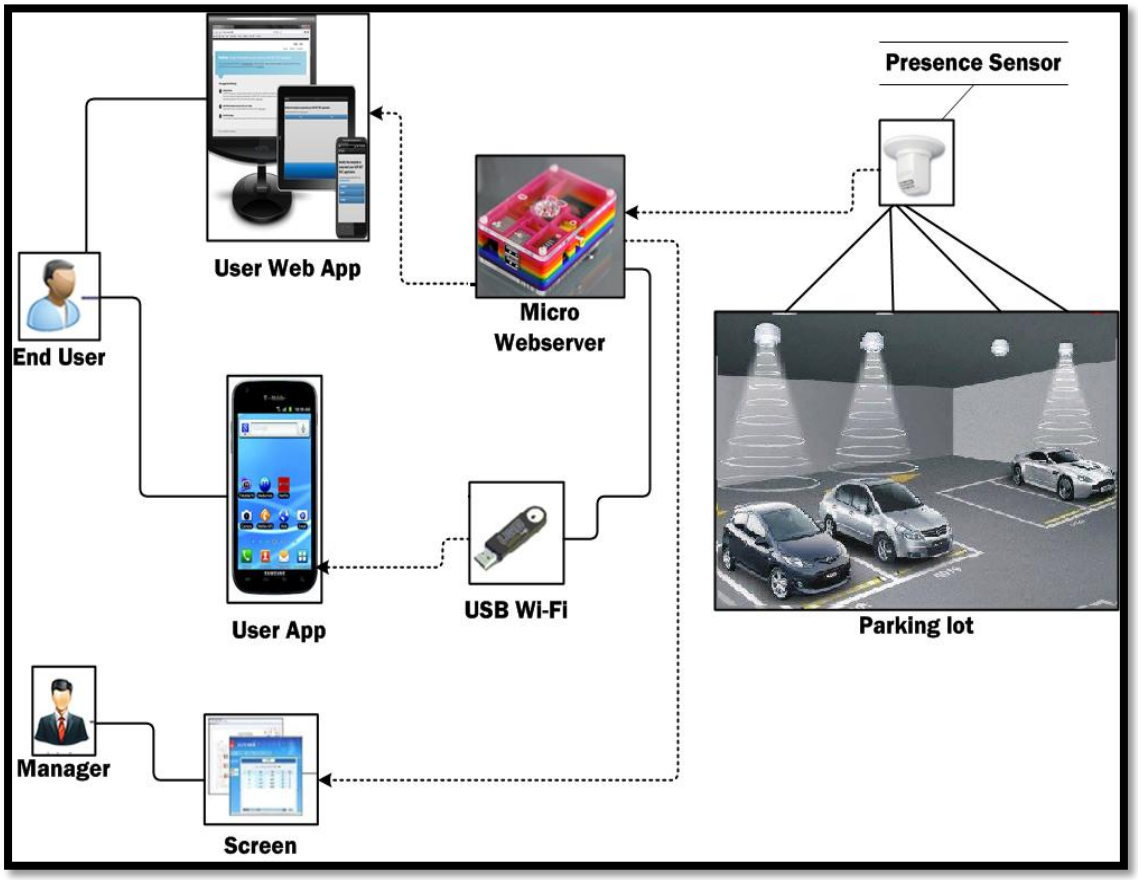

Figure 5.1 system block diagram

5.2 Object Relational Model:

Is a database management system (DBMS) similar to a relational database, but with an object-oriented database model: objects, classes and inheritance are directly supported in database schemas and in the query language. In addition, just as with pure relational systems, it supports extension of the data model with custom data-types and methods class diagram is shown in figure (5.2).[20]

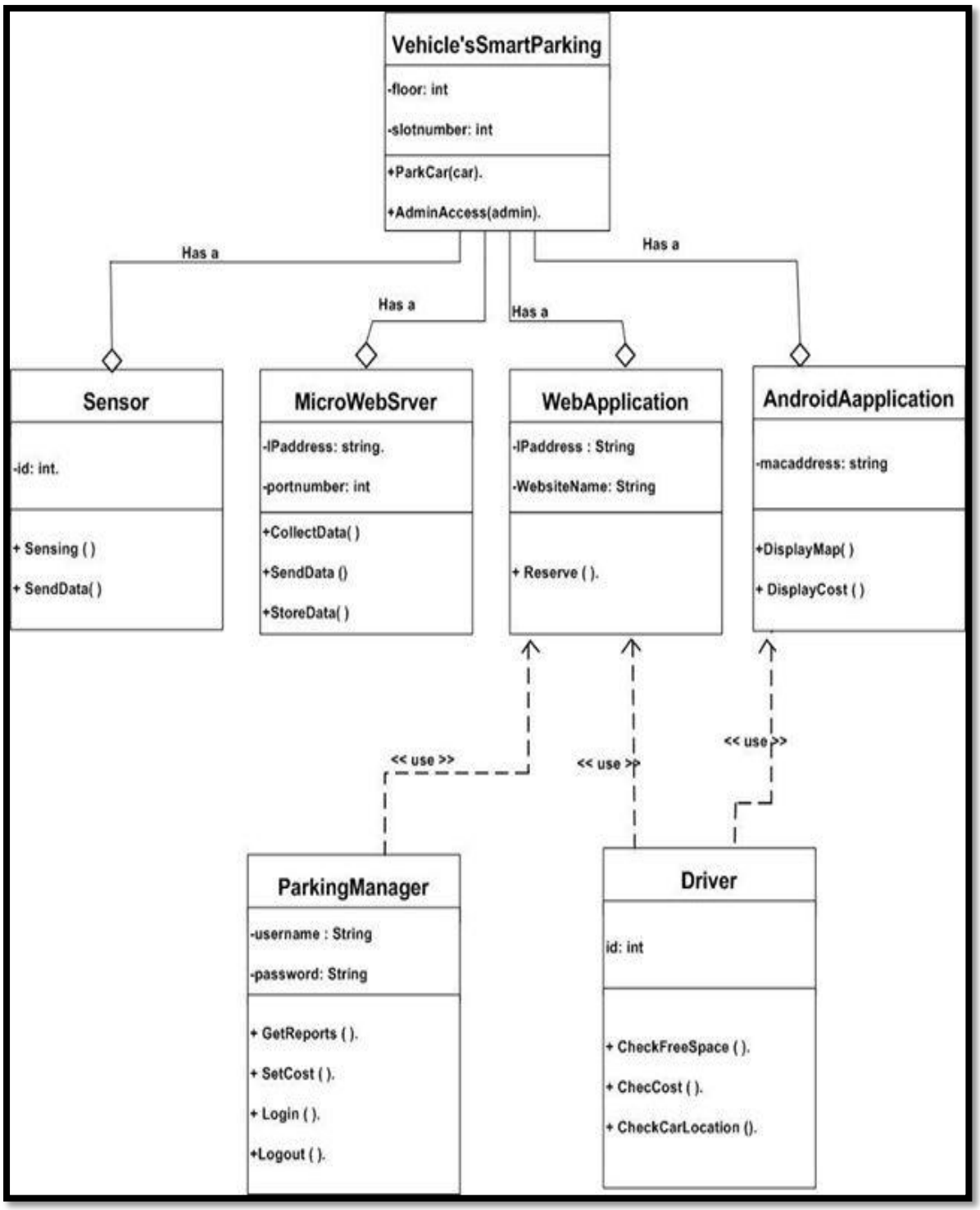

Figure 5.2: Class diagram

5.3 State Behavioral Model:

This section describes the overall system behavior by making focus on two aspects which are dataflow and state flow. A dataflow model shows how the data will processed as how it goes throw the system. State flow shows how the system responses to events and goes from one state to another.

5.3.1 Control Flow

Control flow is shown in figure (5.3).

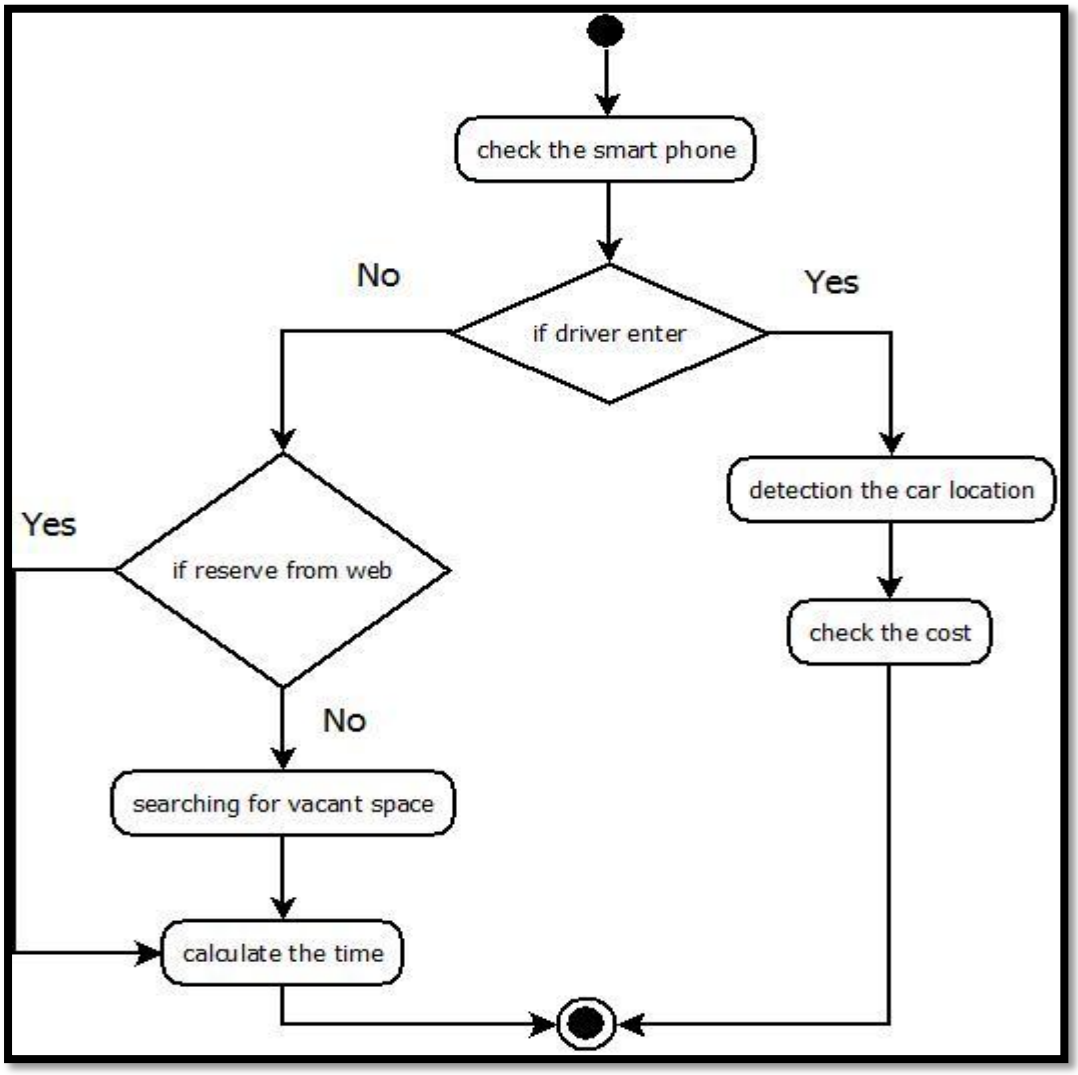

Figure 5.3: control flow

5.3.2 Data Flow

Data flow is shown in figure (5.4).

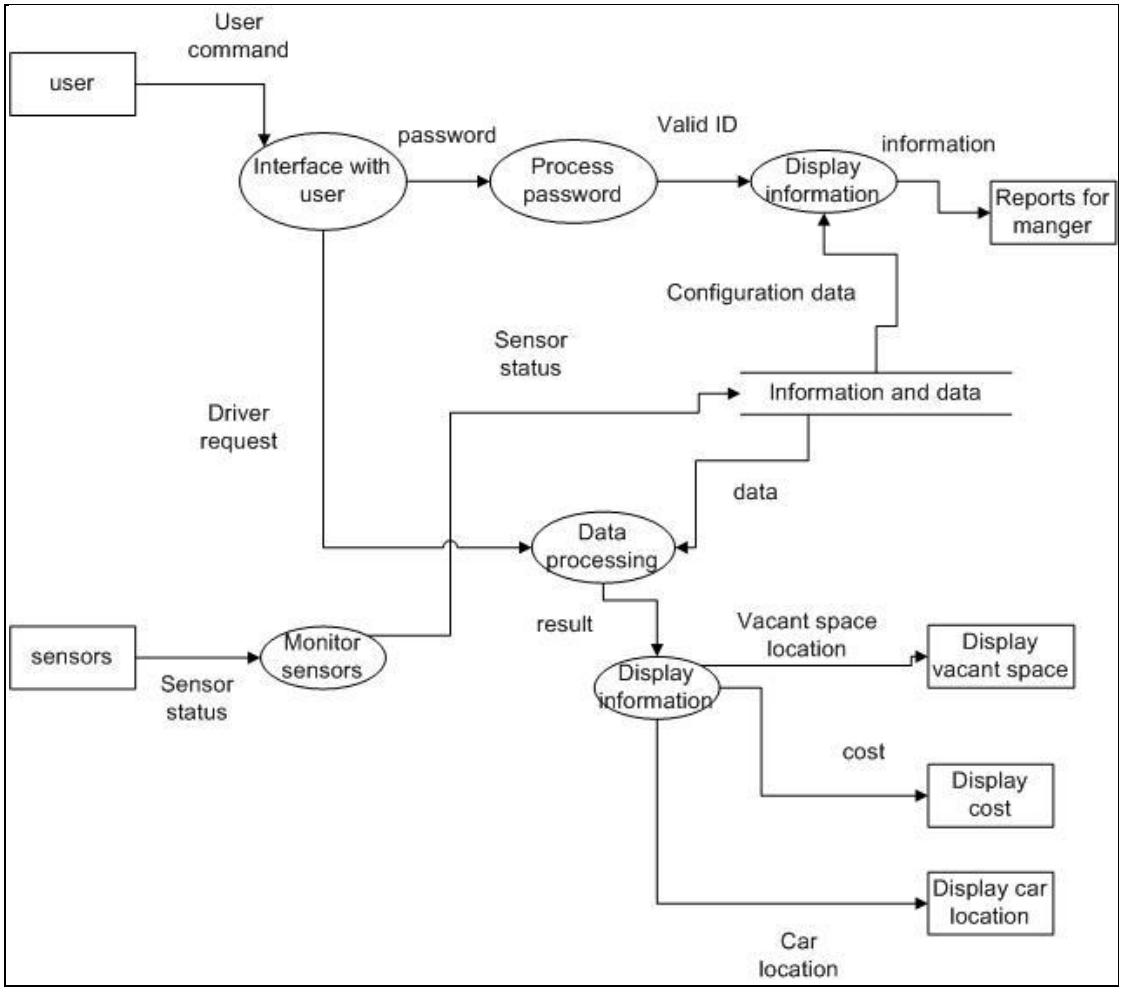

Figure 5.4: data flow

5.3.3 State Flow

Web application state diagram is shown in figure (5.5):

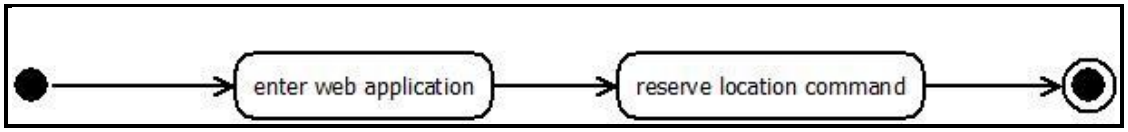

Figure 5.5: Web application state diagram

Parking manager state diagram is shown in figure (5.6):

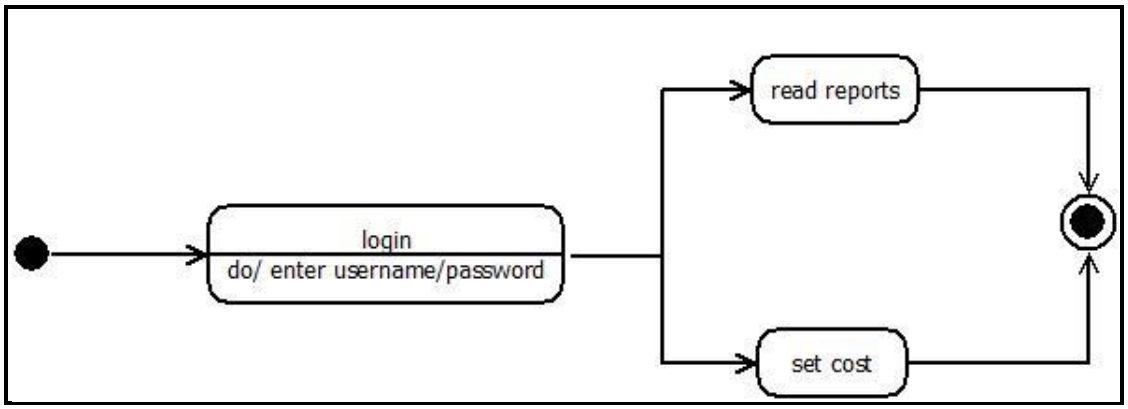

Figure 5.6: parking manager state diagram

Driver state diagram is shown in figure (5.7):

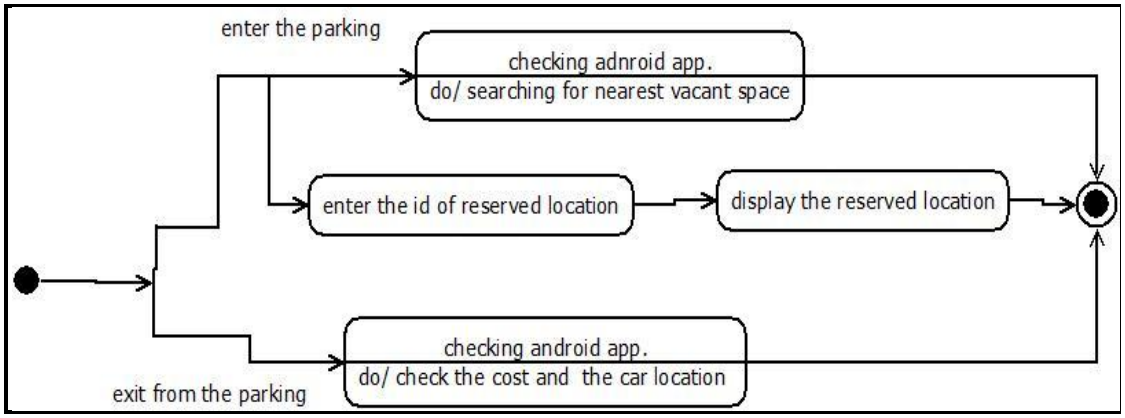

Figure 5.7: driver state diagram

Sensor state diagram is shown in figure (5.8):

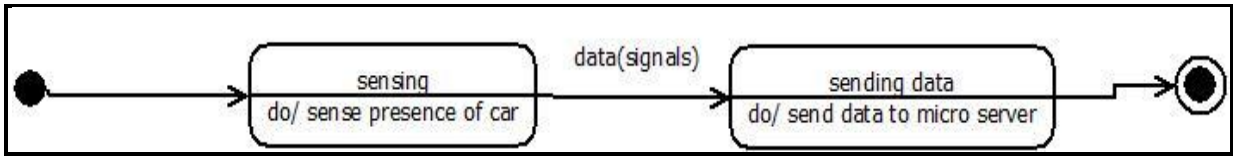

Figure 5.8: sensor state diagram

Micro web server state diagram is shown in figure (5.9):

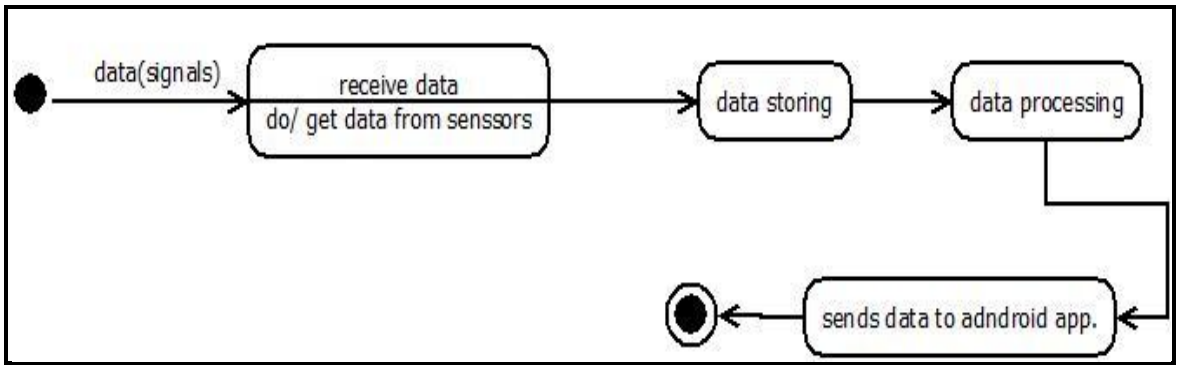

Figure 5.9: micro web server state diagram

Android application state diagram is shown in figure (5.10):

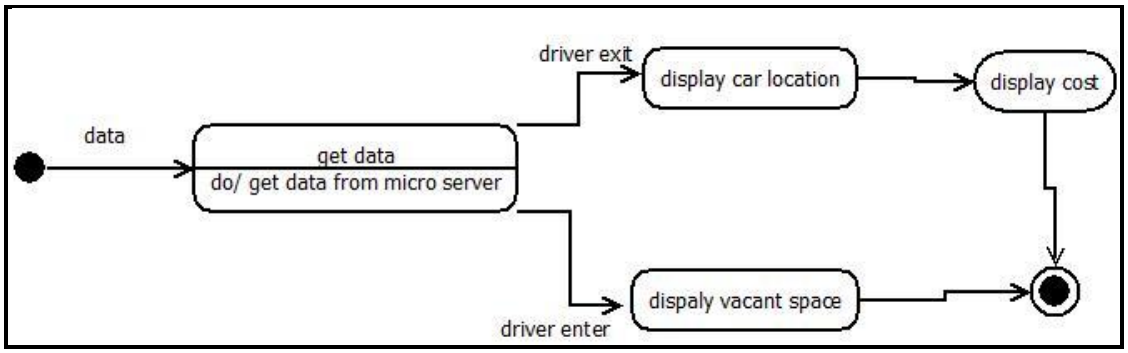

Figure 5.10: android application state diagram

5.4 Subsystem Design

This section shows the diagram of subsystem design. Figure (5.11), our system consists of three subsystems:

1. Driver Interaction: The Driver deal with the system using these tasks:

- Driver reserve location in parking slot.
- Searching for vacant space
- Detect the car location.
- Check the cost when return to the parking.

2. Parking Manager Interaction: The Parking manager deal with the system using two tasks:

- The Parking manager gets data and statistics information.
- The Parking manager set the cost of parking during a specific time period.

3. Detection the Vacant Space:

The system must detect the free slots in the garage by using presence sensor, which is deployed above all slots of the garage and must detect where is the nearest free slots to the driver, The system must collect information from sensor by using micro web server then display the nearest free slots to driver mobile.

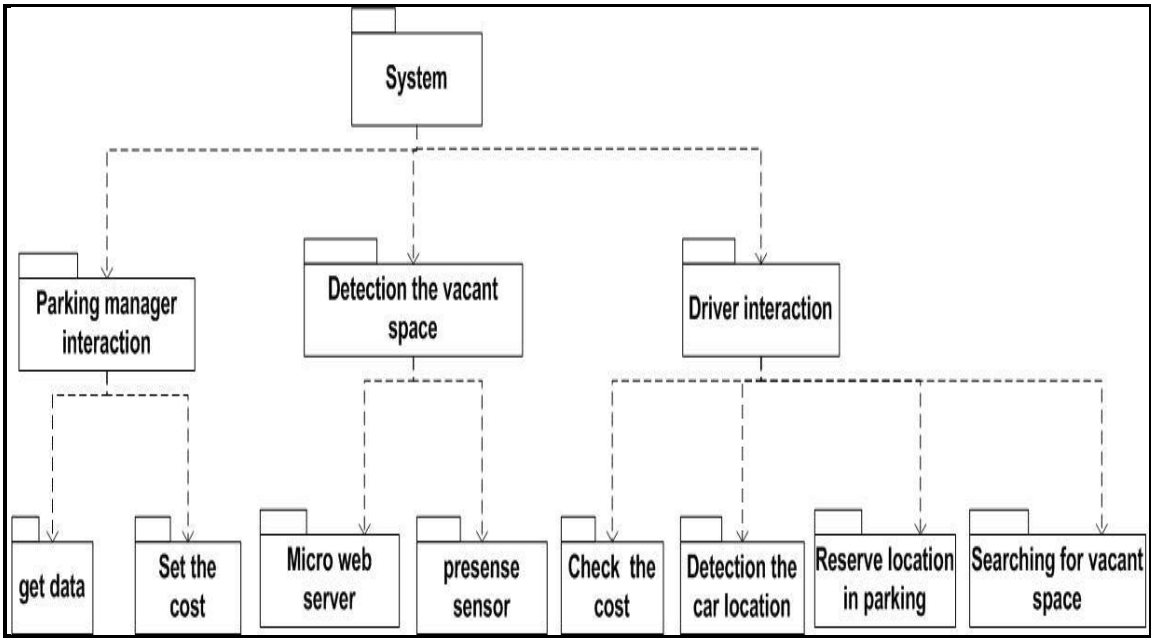

Figure 5.11: subsystem diagram

5.5 Class and Object Design:

This section gives more implementation details about each class and operations.

Table 5.1: vehicle's smart parking class.

Class name: Vehicle'sSmartParking
-floor: int -slotnumber: int
+ParkCar (car). + AdminAccess(admin)

Class name: vehicle's smart parking

The vehicle's smart parking class shows the attributes and the operations of the vehicle's smart parking.

Attributes:

- Floor: int: number of the floors in the park, identification floor of type positive int.
- Slotnumber: int: number of the places in each floor in the park, identification places of type positive int.

Functions:

+parkCar (car): is a public function which allow to the driver to park his car.

+AdminAccess (admin): is a public function which allow to the manger to access to the database.

Table 5.2: sensor class

Class name: Sensor
-id: int.
+ Sensing ( ) + SendData( )

.

Class name: wireless sensor

The wireless sensor class shows the attributes and the operations of the wireless sensor.

Attributes:

- Id: int: number of the sensor in the park, identification sensor of type positive int.

Functions:

+ Sensing ( ): is a public function which return the status of the slot, if car in slot or not.

+ SendData ( ): is a public function which sends the status of the slot to the micro-web server.

Table 5.3: micro-web server class.

Class name: MicroWebSrver
-IPaddress: string. -portnumber: int
+CollectData( ) +SendData () +StoreData( )

Class name: micro-web server

The micro-web server class shows the attributes and the operations of the micro-web server.

Attributes:

- IPaddress: string: is a micro-web server IP address, identification micro-web server of type string.

- Portnum: int: is a micro-web server port number, identify the port for our application, identification port of micro-web server of type int.

Functions:

+ Collectdata (): is a public function which collects information from sensors.

+ Senddata (): is a public function which send the nearest free slots location to the driver.

+ Storedata (): is a public function which store data and information about the park.

Table 5.4: android application class.

Class name: AndroidAapplication
-macaddress: string
+DisplayMap( ) + DisplayCost ( )

Class name: android application

The android application class shows the attributes and the operations of the android application.

Attributes:

- Macaddress: String: is macaddress of smart phone, identification macaddress of type string.

Functions:

+ Displaymap (): is a public function which returns the vacant space on parking map on driver's phone.

+ Displaycost ( ): a public function which returns the cost of time parking on driver's phone.

Table 5.5: driver class.

Class name: Driver
id: int
+ CheckFreeSpace ( ). + ChecCost ( ). + CheckCarLocation ().

Class name: android application

The driver class shows the attributes and the operations of the driver.

Attributes:

-id: int: is an identification number of the driver "MAC address of his smart phone".

Functions:

+ CheckFreeSpace ( ): is a function which driver check his mobile to see parking map to know where the free slots.

+ ChecCost ( ): is a function which driver checks his mobile to see the cost of time parking his car.

+ CheckCarLocation (): is a function which driver take his car from slot where parked in.

Table 5.6: parking manager class.

Class name: ParkingManager
-username : String -password: String
+ GetReports ( ). + SetCost ( ). + Login ( ). +Logout ( ).

Class name: android application

The parking manager class shows the attributes and the operations of the parking manager.

Attributes:

-username: String: is an identification username of the manger

-password: int: is an identification password of the manger.

Functions:

+ GetReports ( ): is a function which manger gets report and information about the park.

+ SetCost ( ): is a function which manger set the cost of parking time.

+ Login ( ): is a function which manger can access to the database of the park.

+ Logout (): is a function which manger logout from the database of the park.

Table 5.7: web application class.

Class name: WebApplication
-IPaddress : String -WebsiteName: String
+ Reserve ( ).

Class name: android application

The web application class shows the attributes and the operations of the web application.

Attributes:

- -IPaddress: String: is an identification of IP address of the web server.

-WebsiteName: String: is an identification of web site domain name.

Functions:

+ Reserve ( ) : is a public function which reserve slot for the car before the driver coming to parking slot.

5.6 Software Interface Design

This section shows the object interfacing and user interface design in the system.

5.6.1 Object Interfacing

This section is show Sequence diagram of Object interfacing. See figure (5.12).

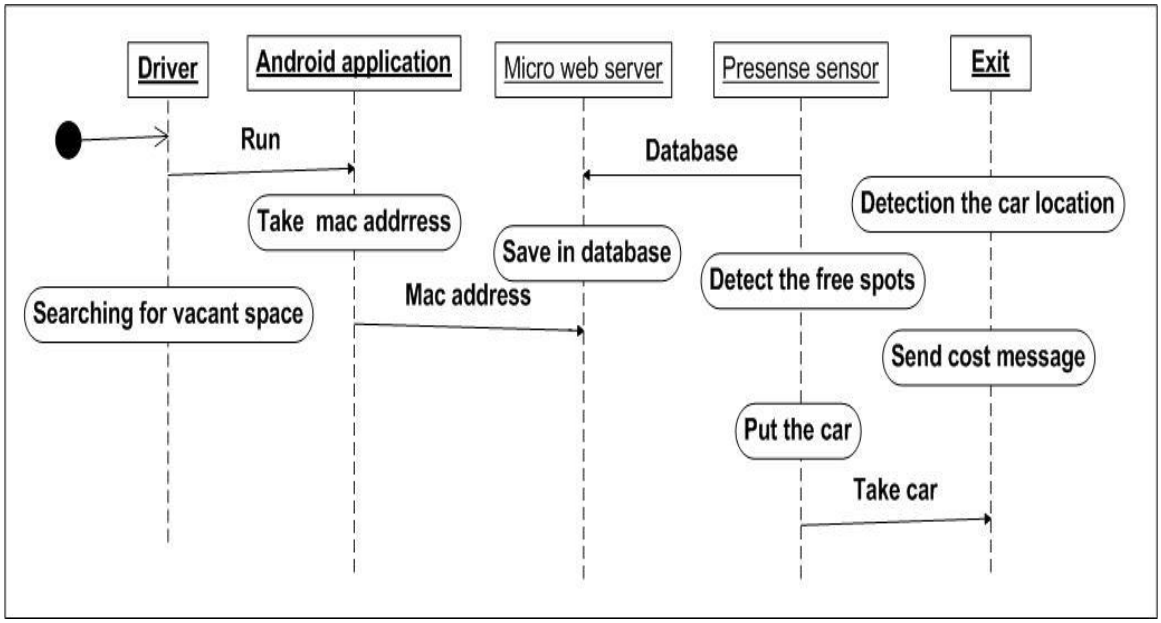

Figure 5.12: Sequence diagram of the system.

5.6.2 User Interface Design

This section shows user interface design in software. In our System the interface between driver and software is smart phone. The smart phone use to display the vacant space on parking map and display the cost.

1- The interface of detect the nearest free slots as shown in Figure (5.13).

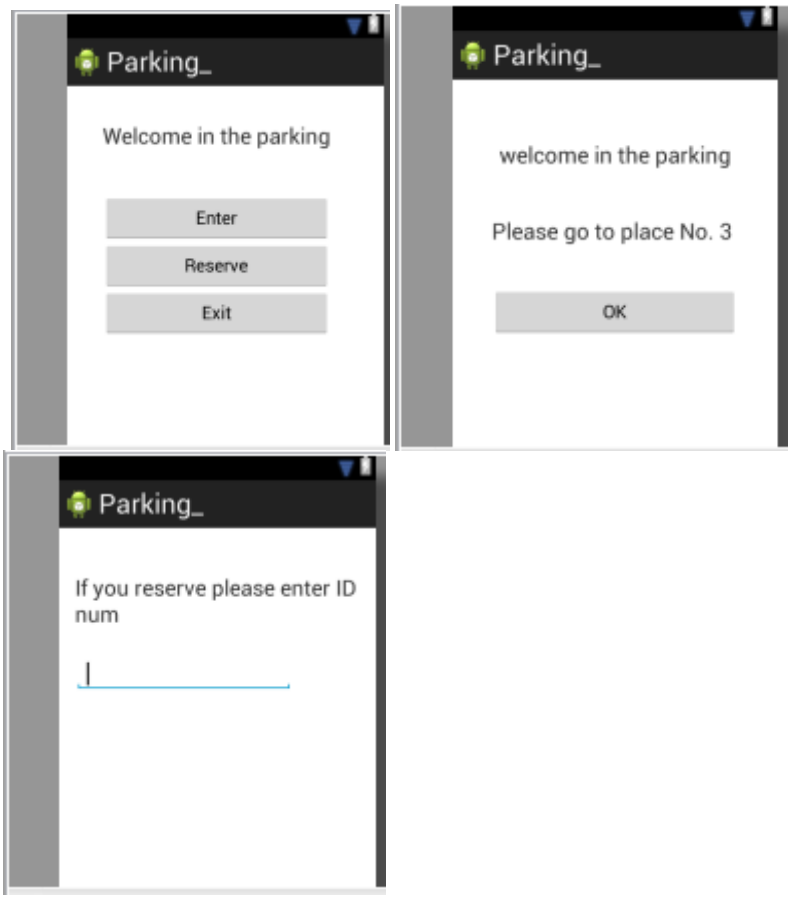

Figure 5.13: Android Interface (a)

2- The interface of Jdisplay message of the location of the car and the cost. as shown in figure(5.14)

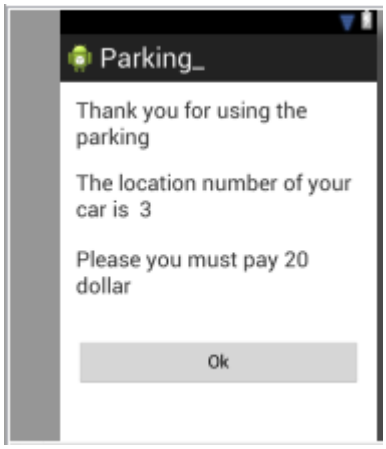

Figure 5.14: android Interface (b)

3-web application interface as shown in figure (5.15):

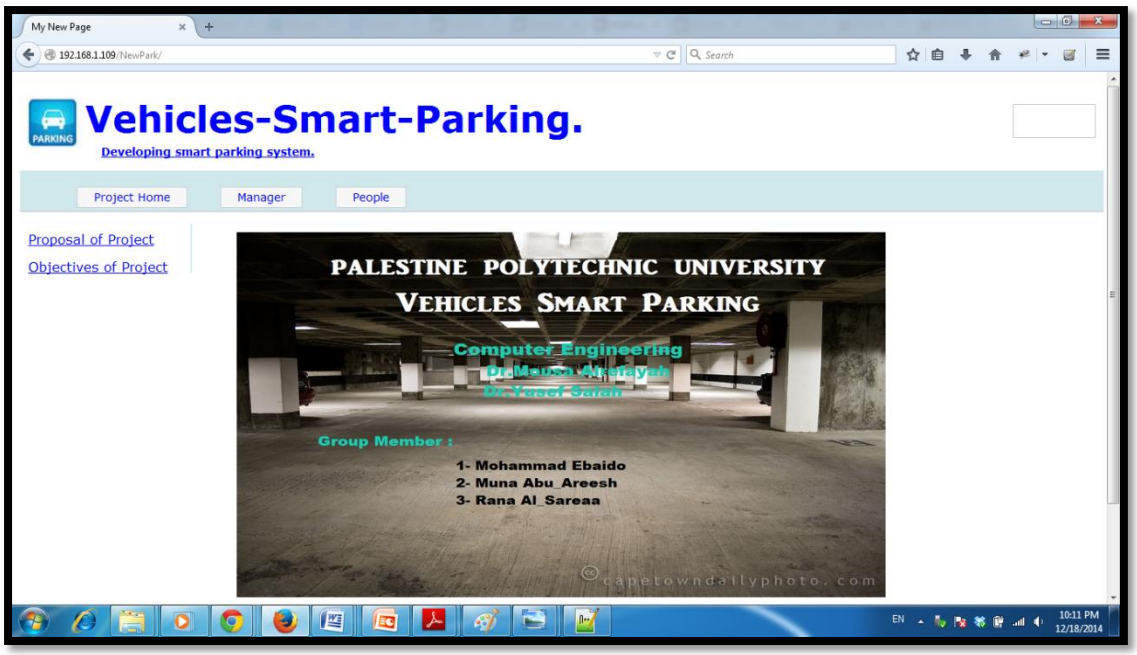

Figure 5.15: web application interface

5.7 Hardware Interface Design:

5.7.1 Component Interface:

The basic hardware components in the system are:

-sensors: detect the vacant space and then send data to micro server.

-micro server: acts as central device, receive data from sensors and send it to smart phone.

-smart phone: receive data from micro server and display the information to the driver.

The connection between all of these components and other is shown in figure (5.13).

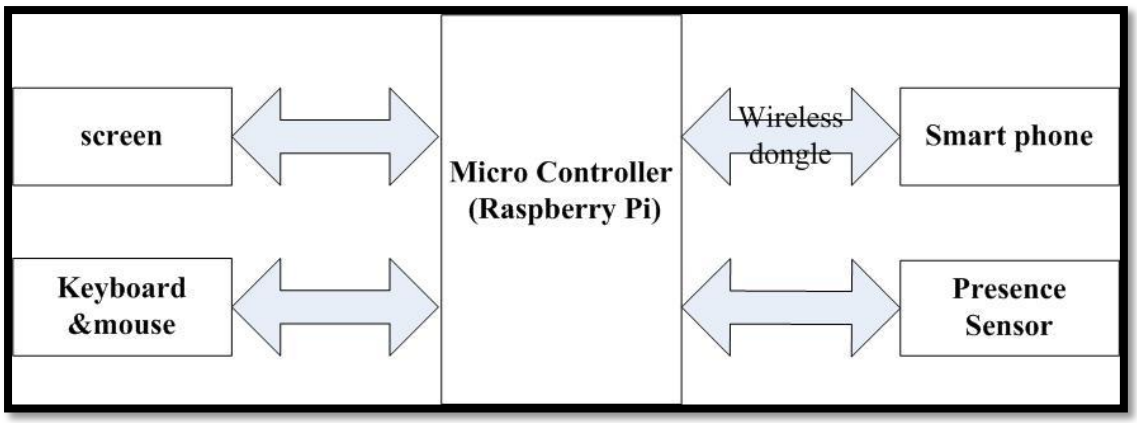

Figure 5.16: Hardware Interface

5.8 Summary

In this chapter we talked about objects-relational model for the software components, state behavioral modeling that includes control flow, data flow and state flow. The chapter shows subsystem design so that it has a description of each subsystem's tasks and components, software and hardware. Also the chapter describes the class and objects design, interface design is for both software and hardware components.

Chapter Six

System Implementation

6.1 Overview:

In this chapter we will show and explain the implementation of the system. We are also going to explain each stage in preparing and executing the project.

6.2 System Implementation:

This section explains the detailed description of the project hardware, software and system units.

6.2.1 Hardware connection and Interface

1) Sensors Connections

Here we are using our parking model which consists of the two floors, where each floor has four sensors. These sensors are connected to the raspberry pi. Each sensor is connected to the General Purpose Input Output (GPIO) of the raspberry pi through three pins (VCC, GND and SIGNAL). The sensor provide a GND signal to the raspberry pi, if there is an object in its range. Otherwise the sensor will provide a 5 volts signal as shown in figure (6.1).

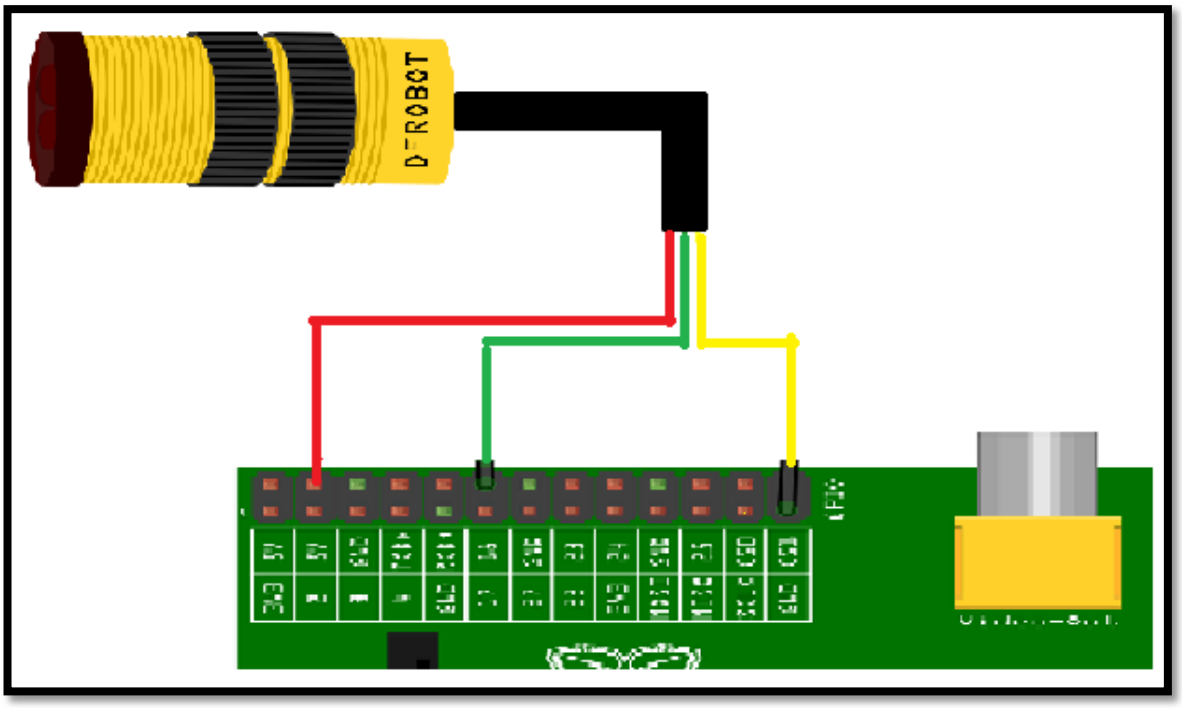

Figure 6.1: Raspberry pi with sensor

2) Wireless Dongle Connection

We used the wireless dongle to make a wireless connection between the raspberry pi and android application on the smart phone. This connection was done through a socket programming (IP address and port number). This is shown in figure (6.2).

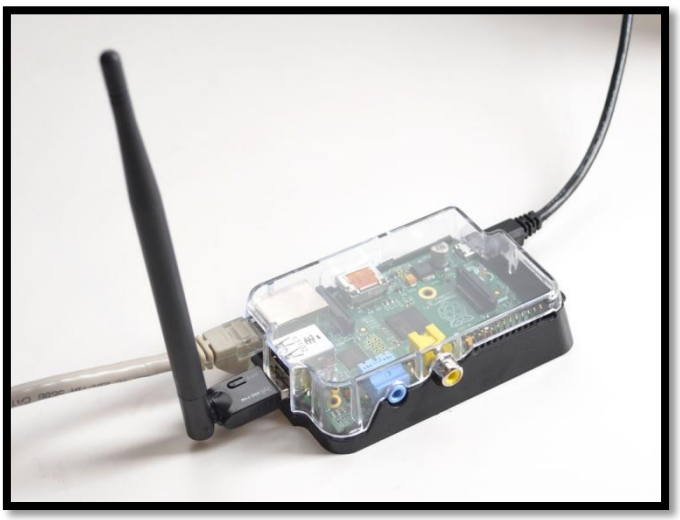

Figure 6.2: Raspberry pi with wireless dongle

3) Screen Keyboard and Mouse Connection

The raspberry pi has an operating system (Raspbian). The system is accessed through a screen, a keyboard, and a mouse. The screen is connected to the raspberry pi a High-Definition Multimedia Interface (HDMI) cable. The keyboard and mouse are connected to via a USB cable as shown in figure (6.3).

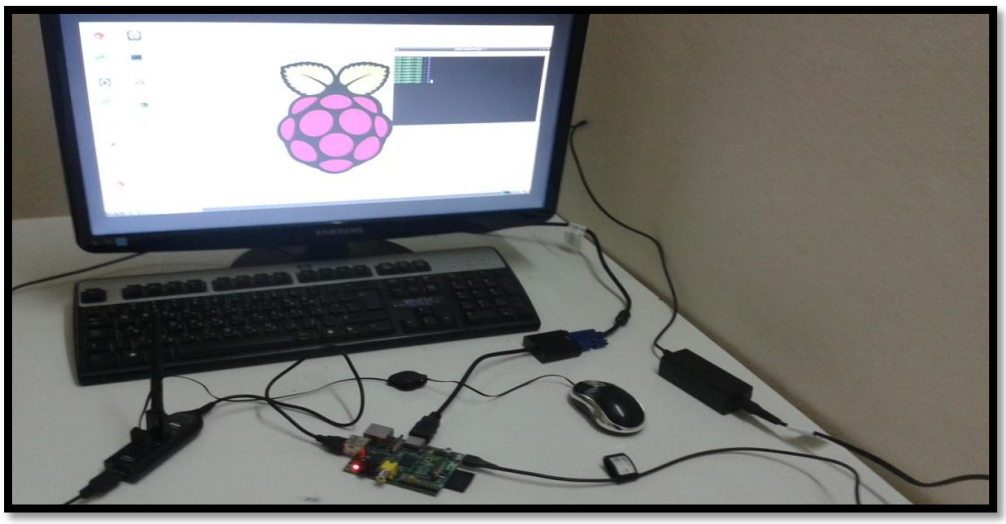

Figure 6.3: Raspberry pi and peripherals

6.2.2 Software Methods and Interface

1) Python Programming Language

We used python programming language in the raspberry pi as the main language in our system. We used python language for:

1- Control GPIO pins; to get data from the sensors.

2- Connect to the database and perform the queries we need.

3- Socket programming at raspberry pi side.

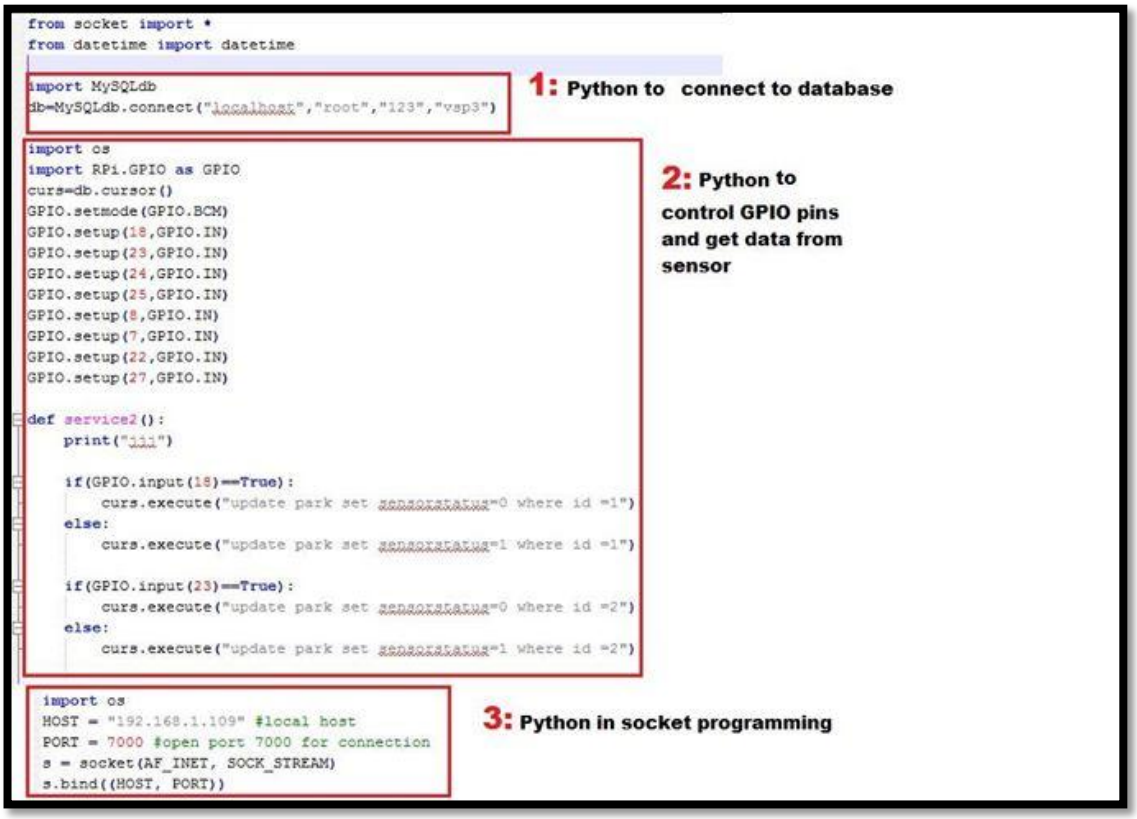

Figure 6.4: Python Programming

2) Android Application

We developed an android application which is implemented using eclipse software that should be installed at driver's smart phone which acts as a client for our system. In the android application we used many methods and classes:

1- Wife manager: which is used to get the macaddress of the smart phone in order to send it to the raspberry pi and then save it in the database.

2- Socket programming at the client side.

3- AsyncTask: is used to establish a continue connection in the background.

4- Animaton class: in order to give our application harmony with users in a graphically way.

The application contains three buttons as shown in figure (6.5). The first one is used to reserve the nearest free slot in the parking lot. It display figure (6.6) to guide the driver to his reservation. If there is no free slot the application will inform the user that there is no free slot as shown in figure (6.7). If the driver used the web to reserve a slot, he\she should choose the second button. After clicking on the application it will ask the driver to enter the number associated with his parking slot as shown in figure (6.8), then figure (6.6) will appear to show him\her exactly where to park. The last button used when the driver exit to show where his\her car is and how much the cost is as shown in figure (6.6).

Figure 6.5: android (a)

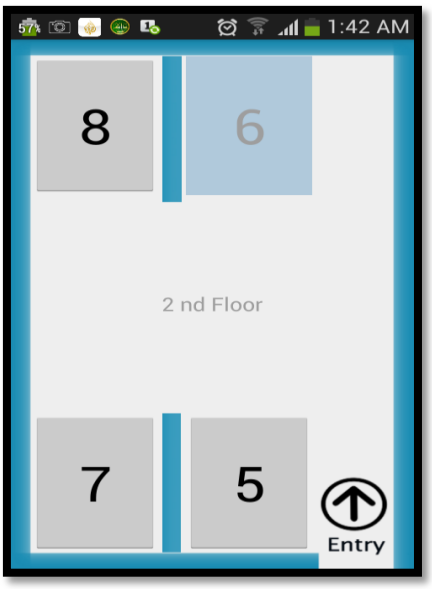

Figure 6.6: android (b)

Figure 6.7: android (c)

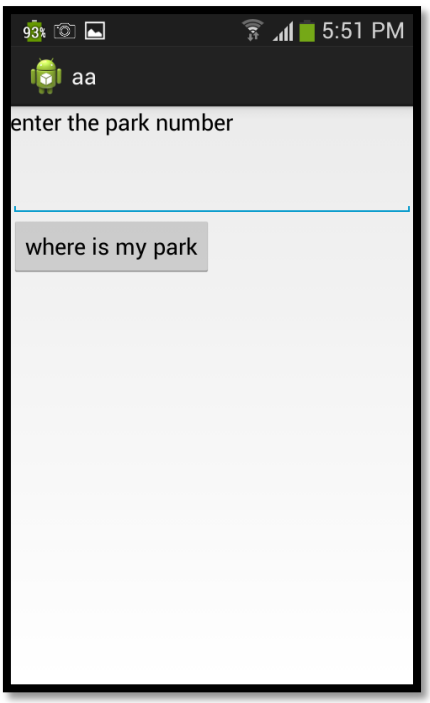

Figure 6.8: android (d)

3) Web application

In our system raspberry pi acts as a web server, when setting up a web site using scripting and descriptive programming languages, we used:

1- HTML as descriptive language.
2- CSS to design the style of the web page.
3- Php as server-side scripting language.
4- Mysql with php to deal with database.
5- Apache server.

The web site has three pages; the first page "Home Page" that describe the project and its objectives as shown in figure (6.9). The second page "Manager" that ask the manager to login to set the cost as shown in figure (6.10) and display reports of the parking as shown in figure (6.11). The third page "People" for drivers to reserve free slot in the parking in addition the driver can download the android application for his smart phone as shown in figure (6.11).

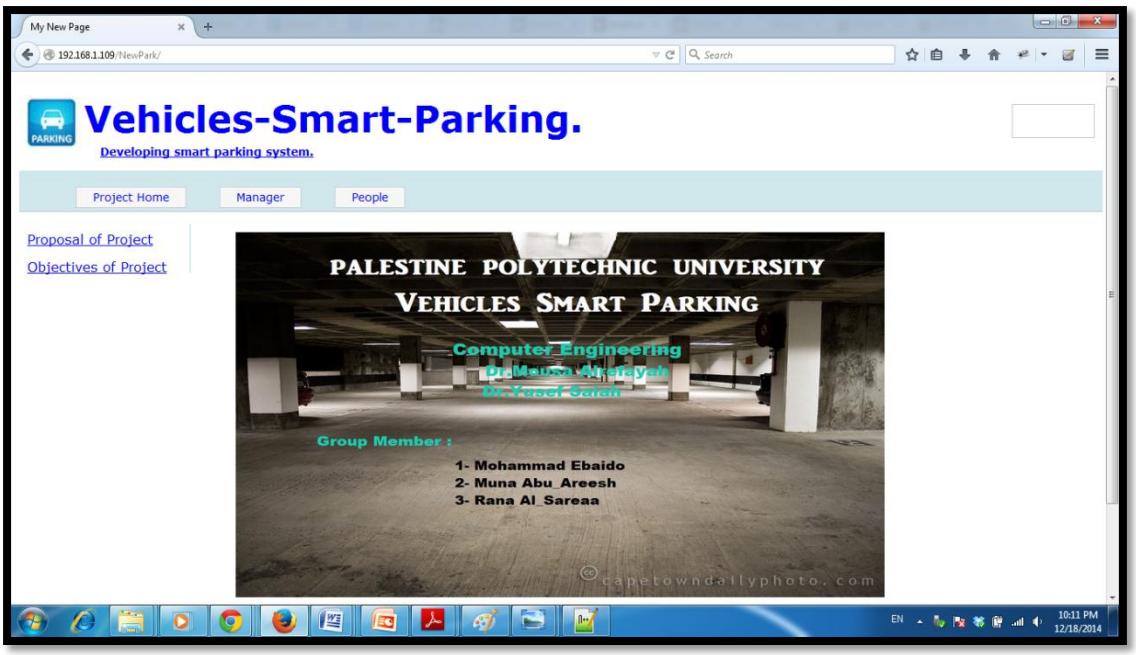

Figure 6.9: Home Page

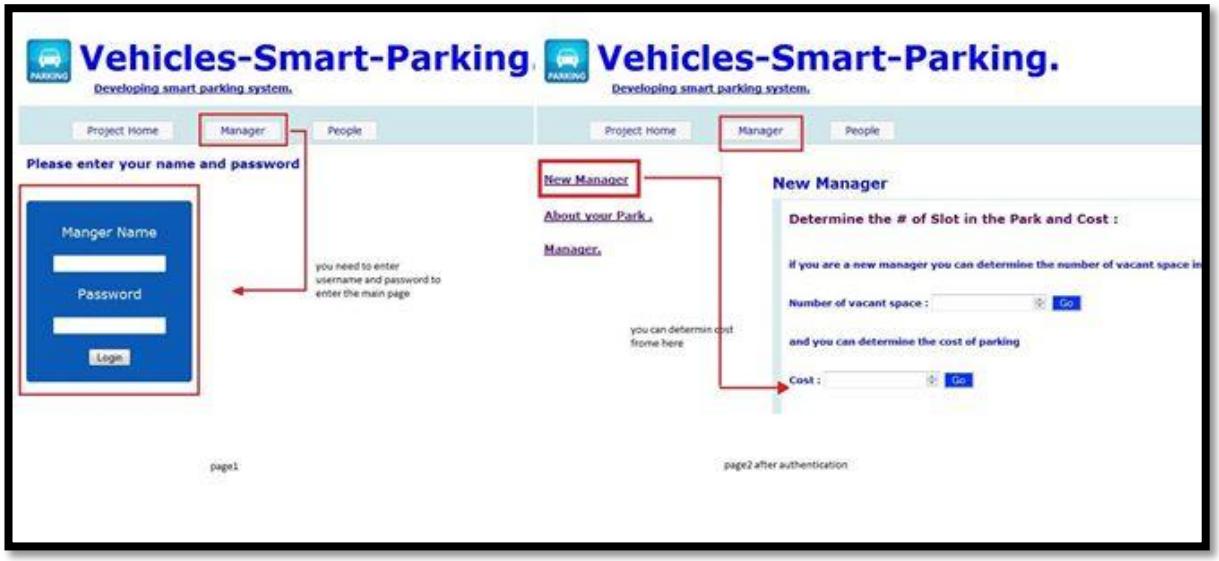

Figure 6.10: Manager (a)

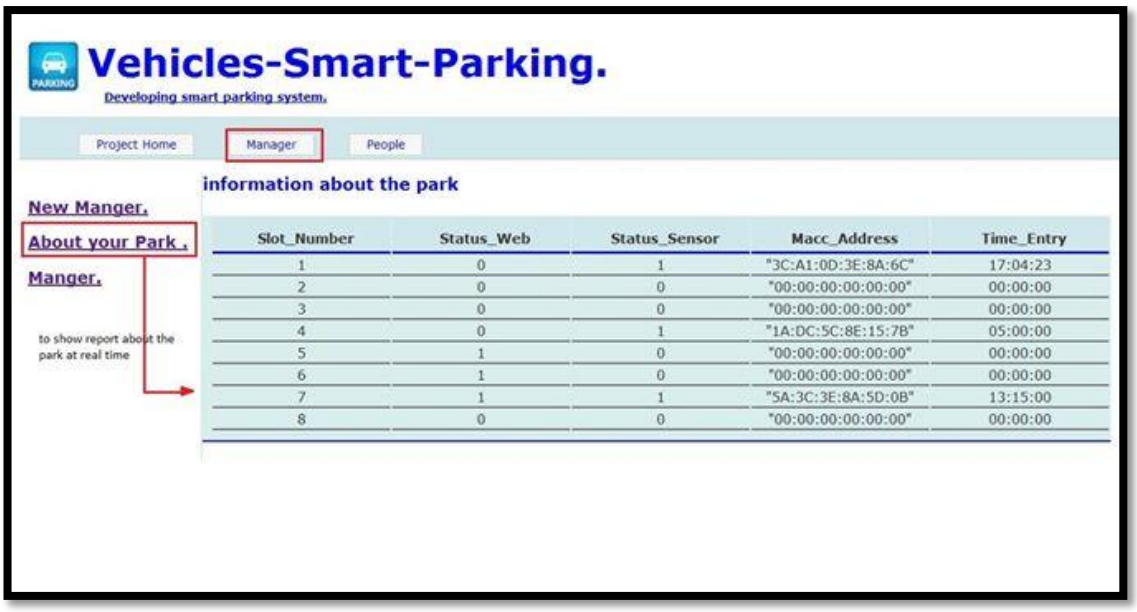

Figure 6.11: Manager (b)

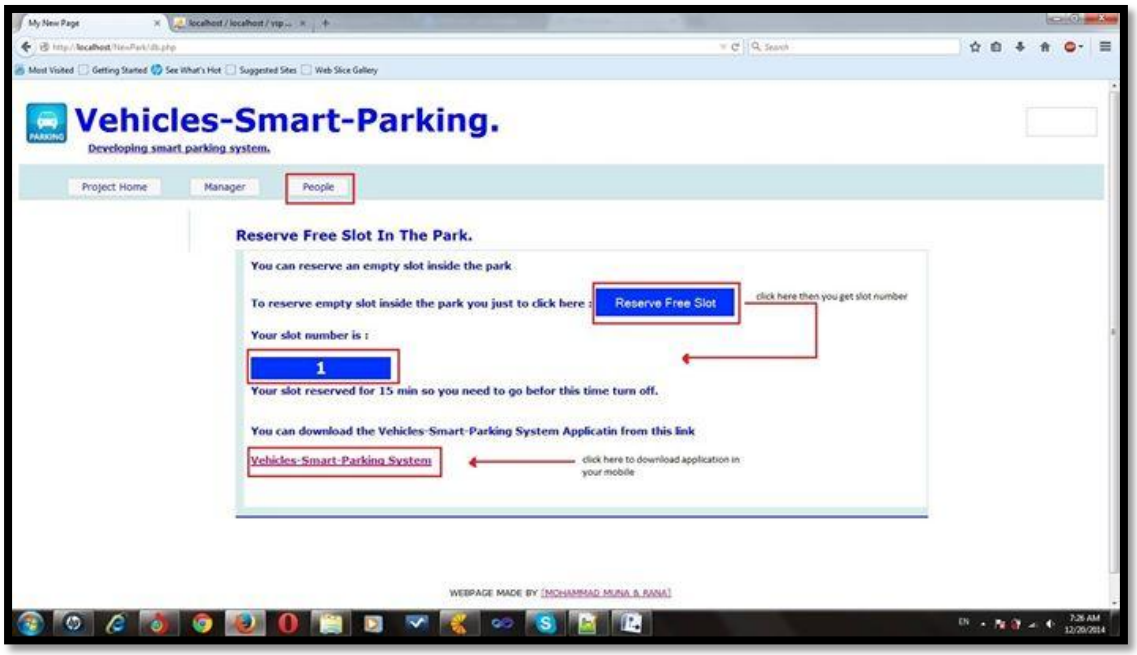

Figure 6.12: People

4) Mysql

We used mysql as database management system. We have two tables in our system "park" and "manager" table.

Park table consists of five columns (id, status, sensorstatus, macaddress, and time entry) as shown in figure (6.12).

- Status: present if the slot is reserved from web or not.
- Sensorstatus: present if the slot is occupied or not.
- Macadress: store macaddress of smart phone.
- Entry time: store the time when the driver park the car.

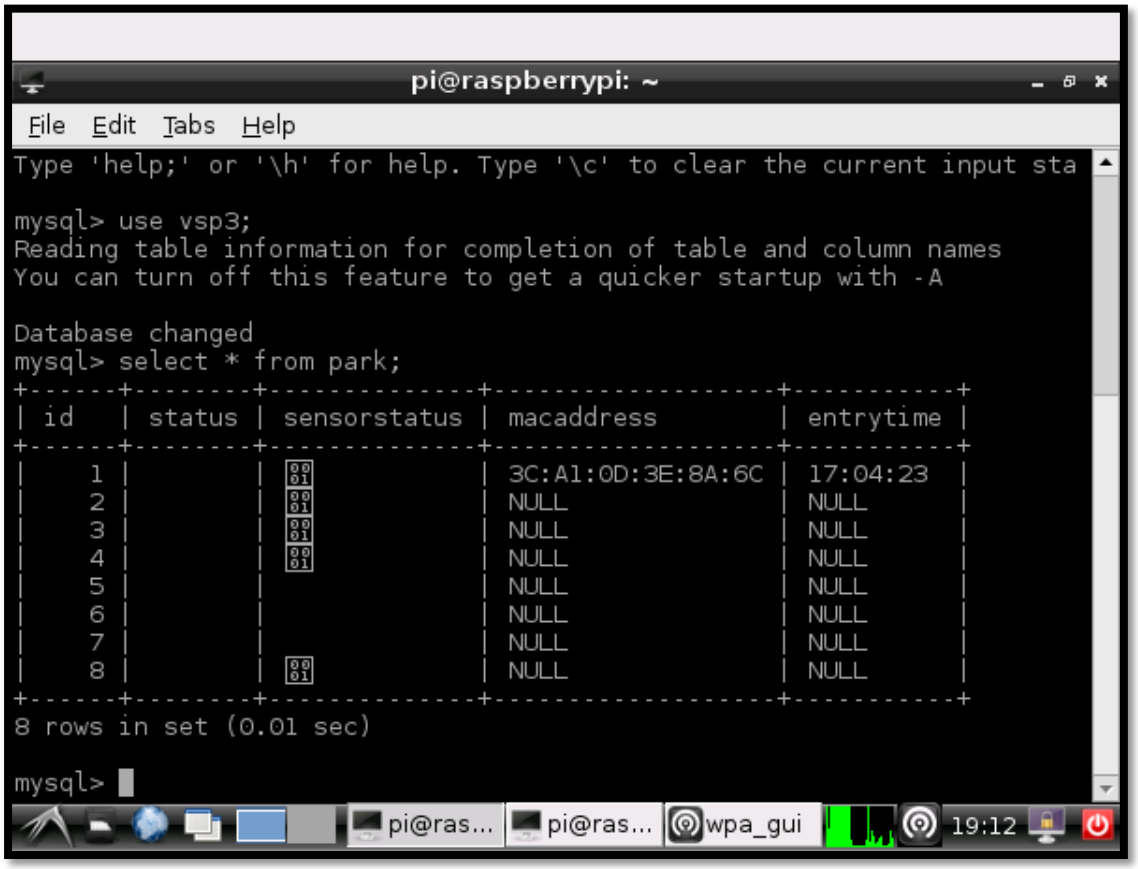

Figure 6.13: Park Table

Manager table which consist of four columns (id, username, password, and cost) as shown in figure (6.13).

- Username: store manager name.
- Password: store manager password.
- Cost: store the cost of parking per hour.

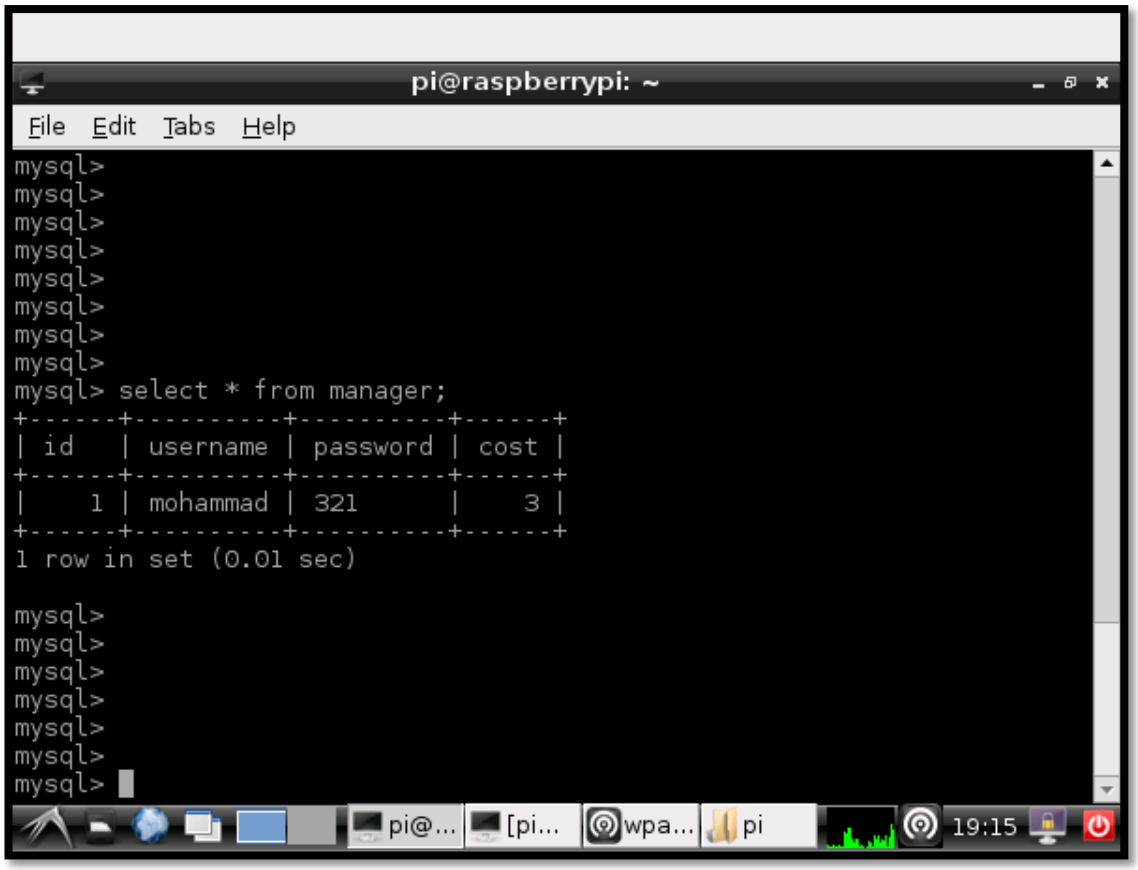

Figure 6.14: Manager Table

6.3 Integrated System Implementation:

The final stage is to connect all the previous subsystems together to get the final circuit. The whole system implemented together as shown in the figure (6.14).

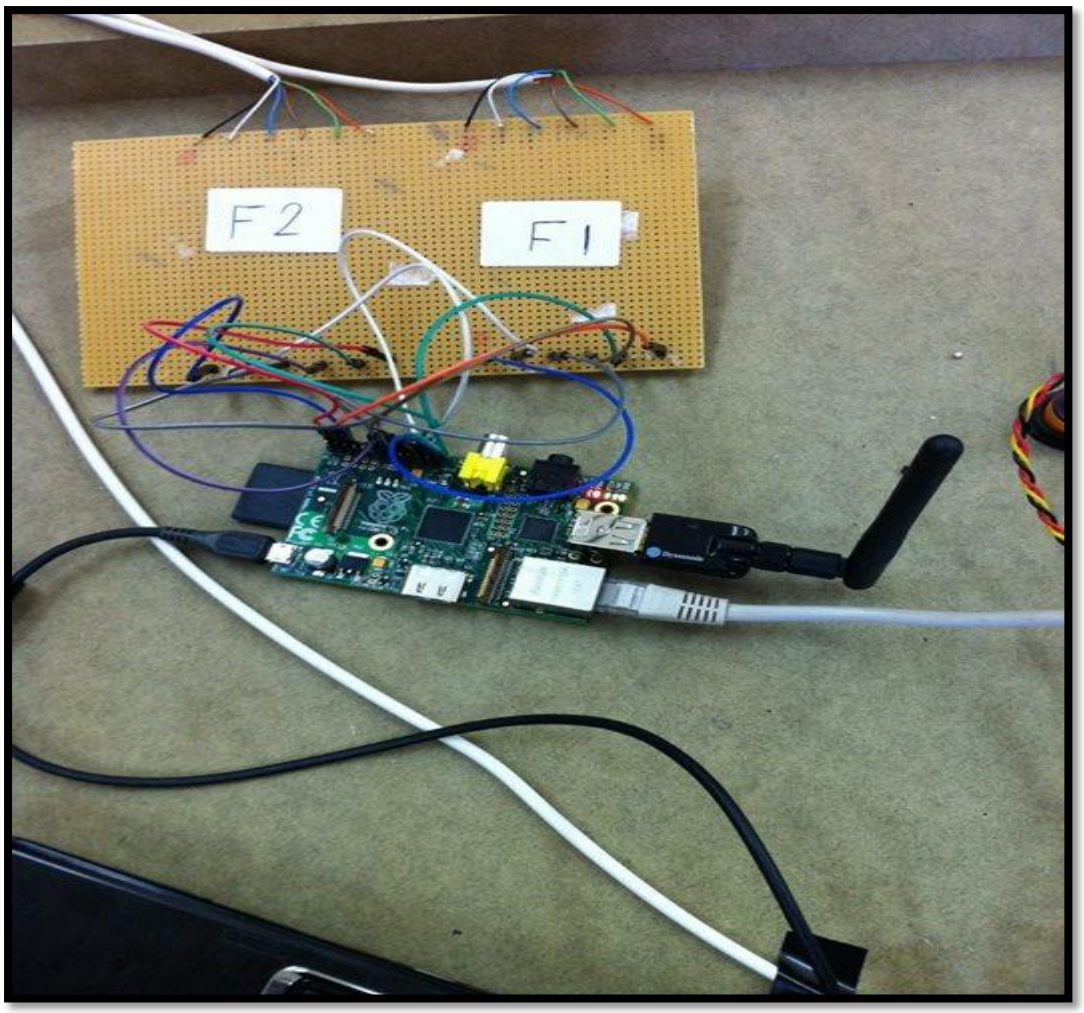

Figure 6.15: Integrated System

6.4 Summary:

In this chapter we explained all the work in details, in addition to figures and tables that shows all steps of the project work. Each stage had been explained individually.

Chapter Seven

System Testing

7.1 Overview

7.2 System Testing

7.3 Summary

7.1 Overview:

In this chapter project implementation was done step by step, each step was tested directly and results were recorded. Results of each hardware design and software codes will be presented. Each subsystem will be tested alone and results will be documented, then the whole integrated system together. Figures and tables will be provided in the testing chapter.

7.2 System Testing:

This section will provide the testing and all results that accrued during the testing process.

7.2.1 Presence Sensor Testing:

In this section we will elaborate about the presence sensors connection with the raspberry pi microcontroller. First when we connected the sensor with raspberry pi, it shows the result on LEDs. The next figure shows the connection between the raspberry pi and presence sensors as shown in figure (7.1)..

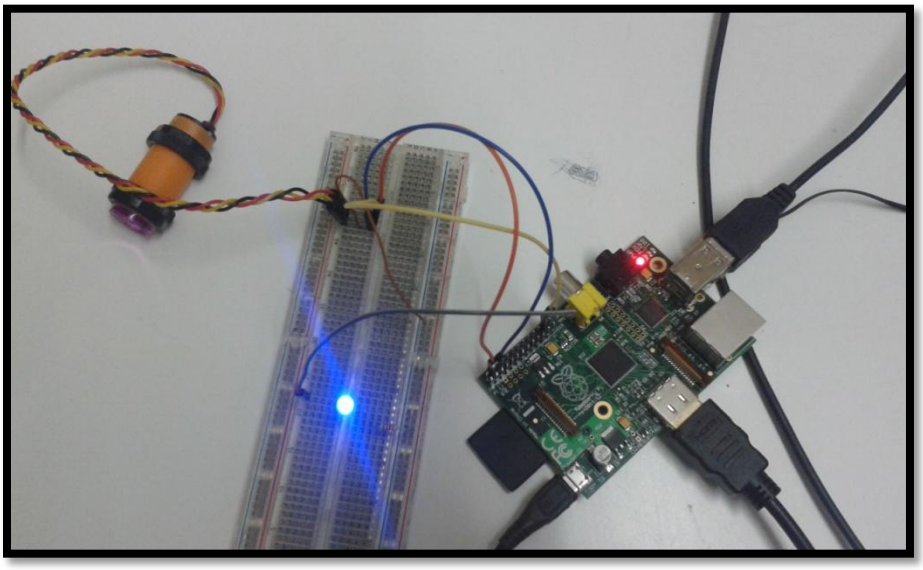

Figure 7.1: presence sensor testing

Finally we tested all sensors with raspberry pi and got the result on the terminal as shown in figures (7.2-7.5):

If the parking is empty the system will select the first slot, as shown in figure (7.3):

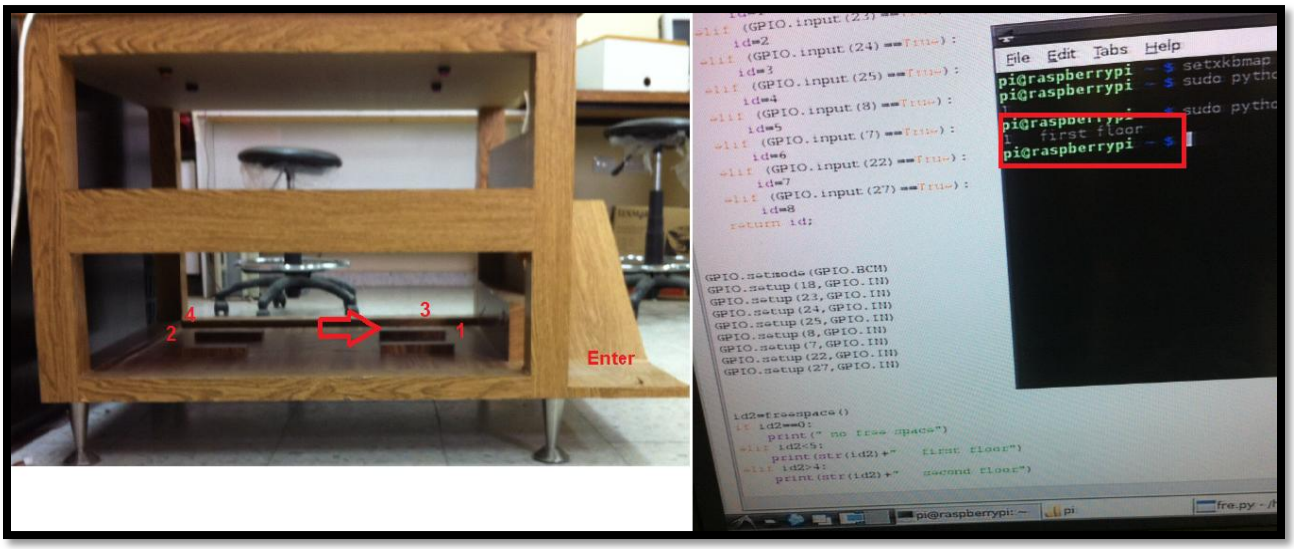

Figure 7.2: all presence sensors testing (a)

The system select the nearest vacant space in the parking from available free slots, as shown in figures (7.3 and 7.4)

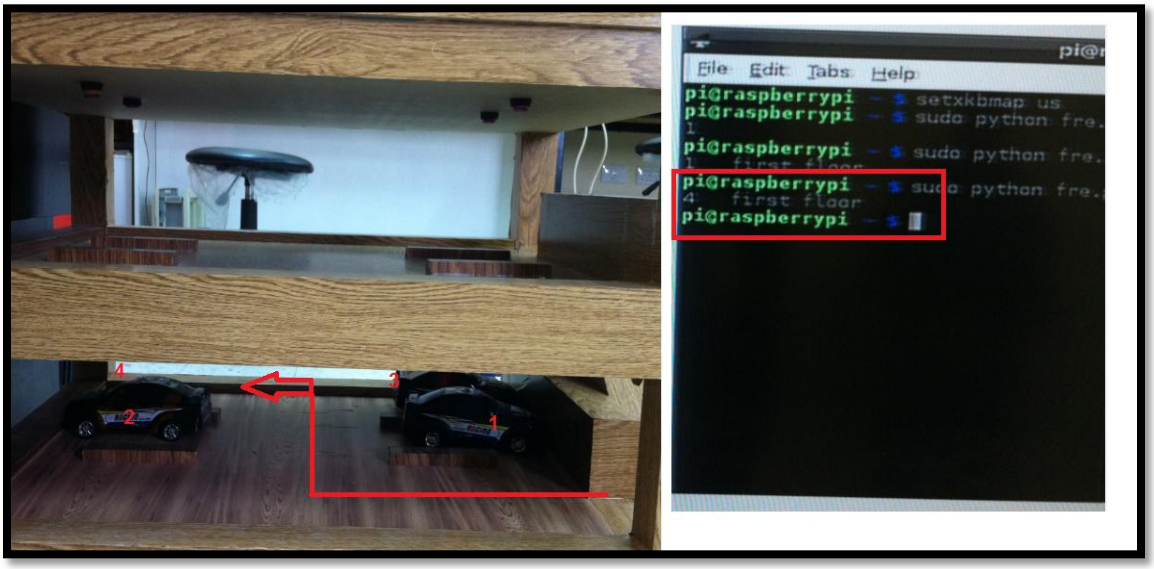

Figure 7.3: all presence sensors testing (b)

Figure 7.4: all presence sensors testing (c)

When the parking is full, the system shows that there is no available free space, as shown in figure (7.5)

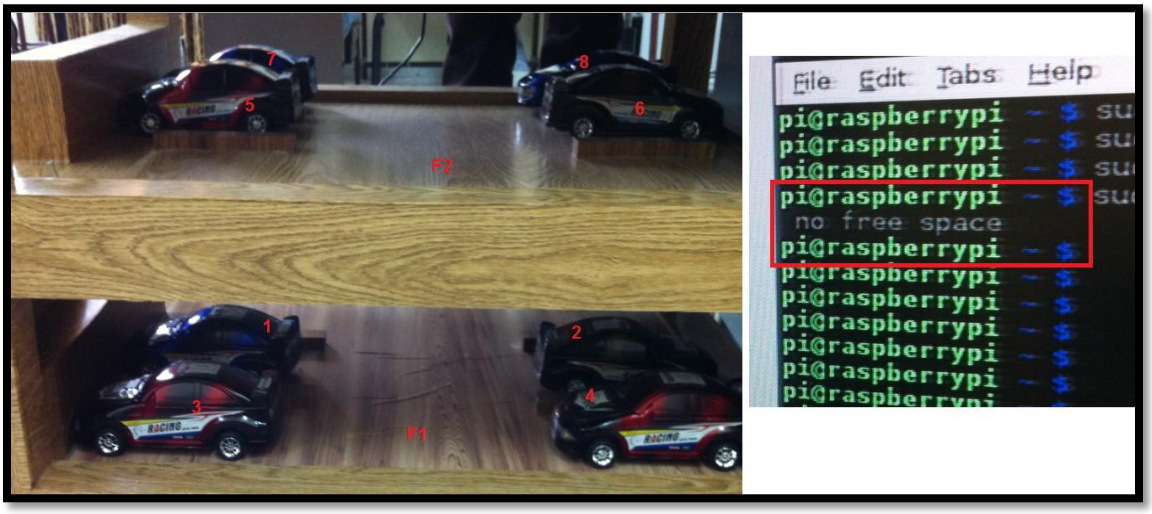

Figure 7.5: all presence sensors testing (d)

7.2.2 Raspberry pi Testing

In this subsection we will elaborate about the raspberry pi connection with the other system components.

1) Wireless Connection Testing:

This phase include wireless connection between raspberry pi and android application. First we get MAC address from android application of smart phone and display it as shown in figure (7.6).

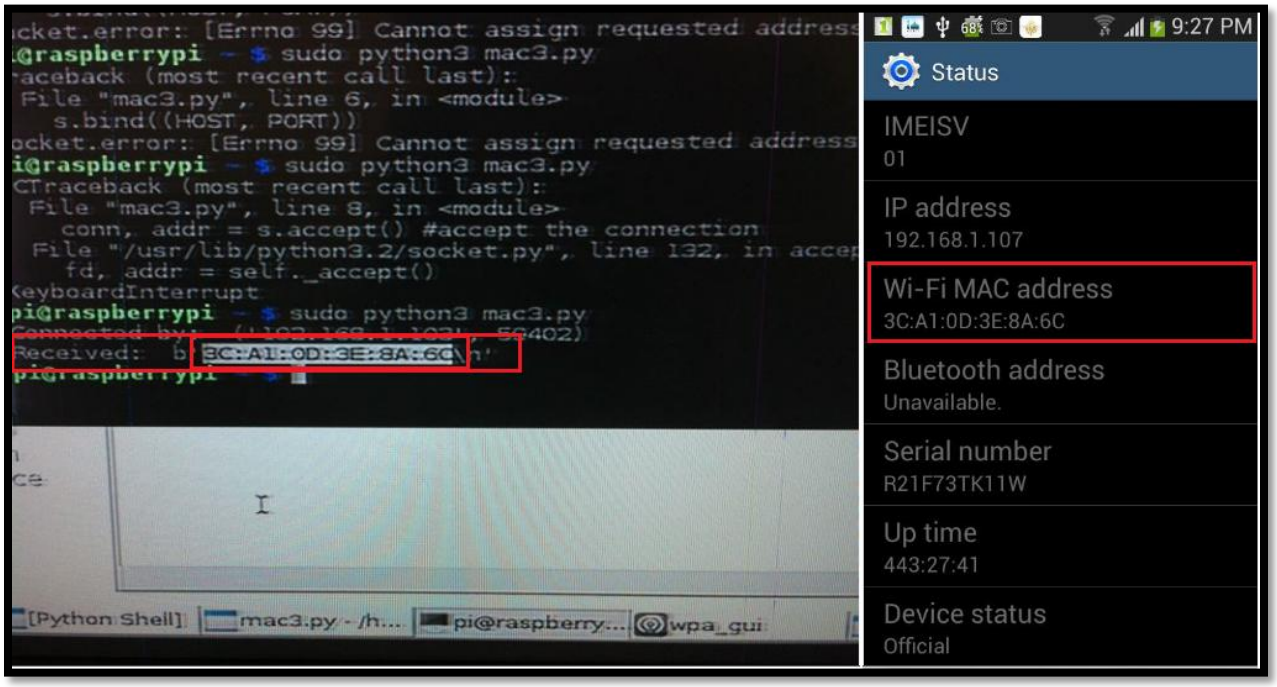

Figure 7.6: Wireless Connection Testing

2) Database Testing:

We created a table and perform some queries at Linux terminal, as shown in figure (7.7):

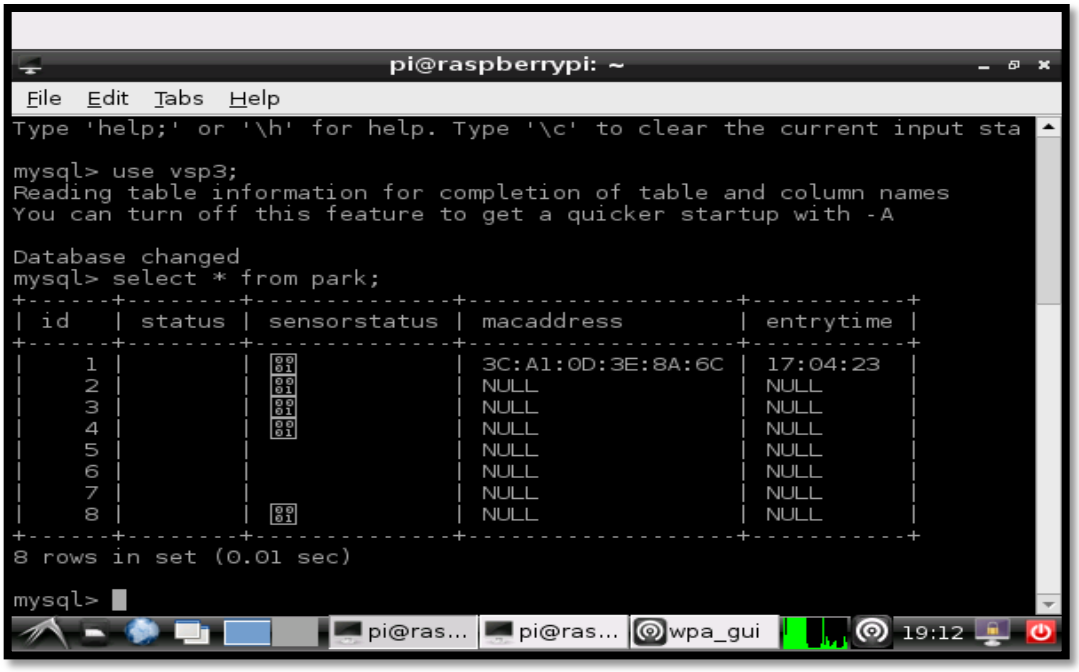

Figure 7.7: Database Testing

7.2.3 Mobile Application Testing:

This phase includes the wireless communication sending and receiving data from raspberry pi as shown in figure (7.8).

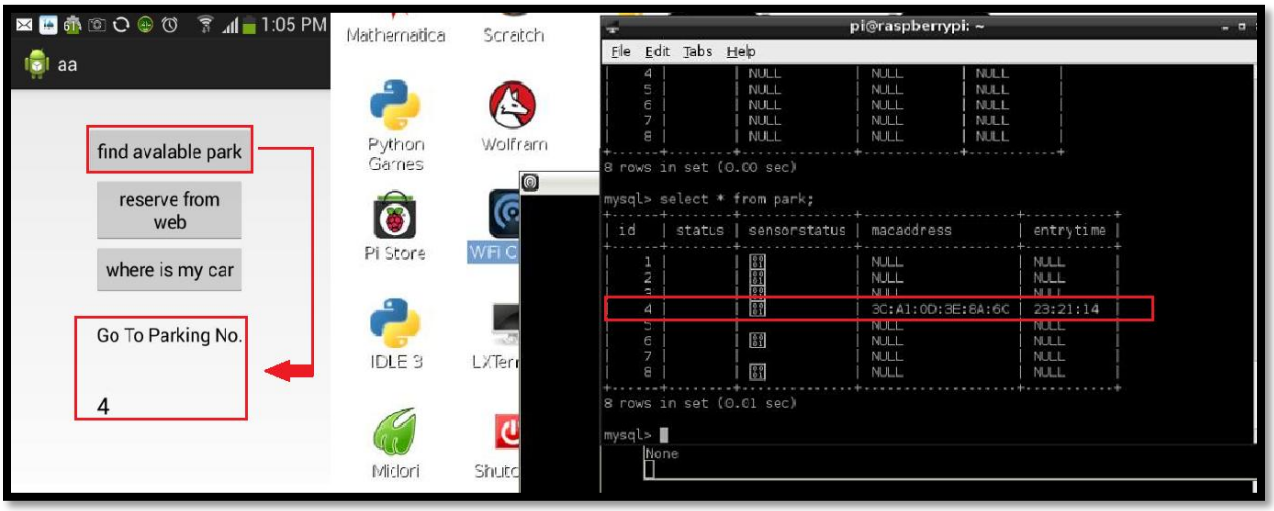

Figure 7.8: android communication testing (a)

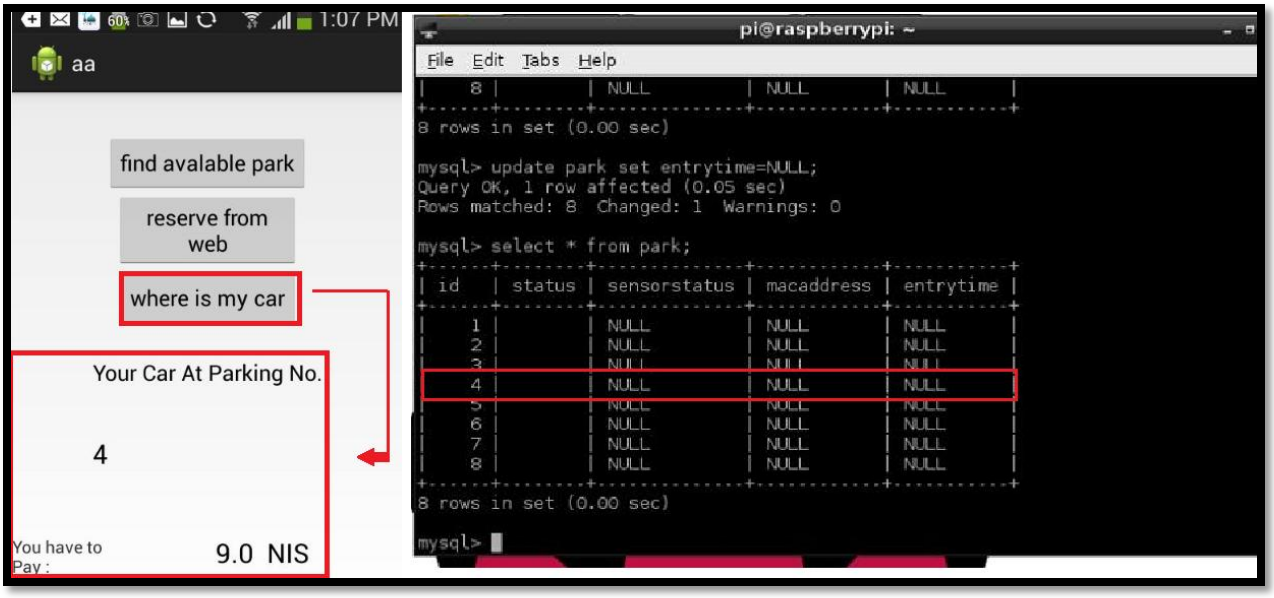

Figure 7.9: android communication testing (b)

7.2.4 Web Application Testing:

This testing process was done by the user accesses the web page and allocate location in the parking remotely. We tested the web site first at local host on raspberry pi as shown in figure (7.10).

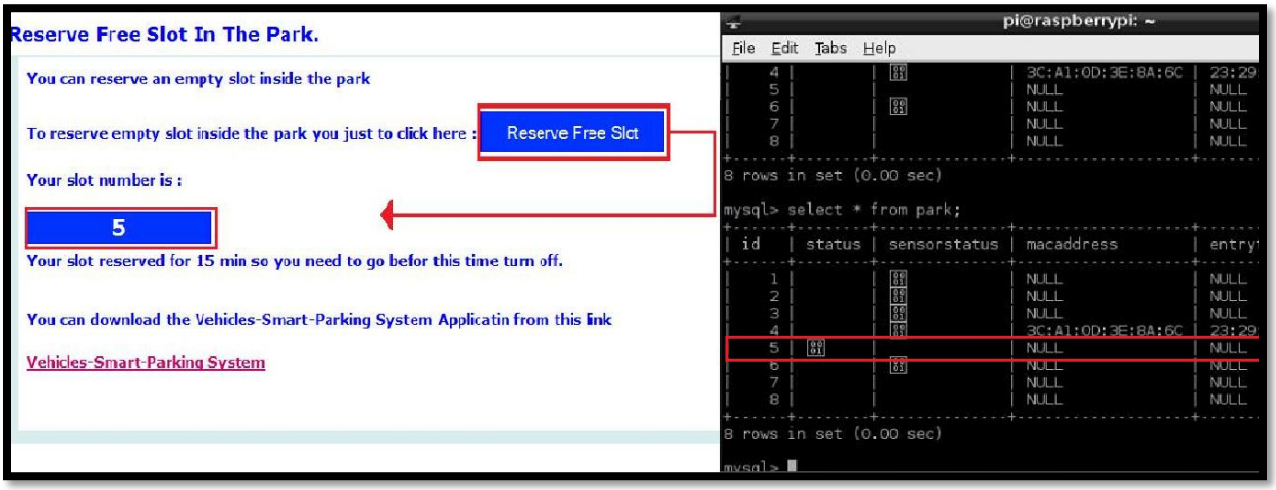

Figure 7.10: web application testing

7.3 Summary:

In this chapter we started by testing the raspberry pi which showed the results on LEDs. We also showed wireless connections between the raspberry pi and android application that is provided with the database testing by figurers. Each part is examined as individual part. At the end the whole system is tested as one part.

Chapter Eight

Conclusions and Future Work

8.1 Overview

8.2 Conclusions

8.3 Future Work

8.1 Chapter Overview:

In this chapter we will consider all conclusions of the projects, in addition to future work of the project.

8.2 Conclusions:

The project has been done as a step to develop the parking system and reduce the human role in wasting their time looking for a parking spot.

The system will direct and guide drivers to the most convenient available parking space in the parking facility in the shortest amount of time and least effort. The system consists of web and android application on the drivers mobile, which allows the driver to know where the nearest available parking spaces is. The system can display the location of the driver's car on his smart phone through the android application when one comes back to get ones car. In addition, the system allows the driver to reserve a slot in the parking from anyplace where the internet is available.

Almost all the goals of our system are discussed and the way of achieving it. An automatic parking system has been built in order to make the parking process easier, accurate, and reduces the needed time.

Our experience and skills from the implementation of the project are:-

- Developing android applications.
- Experience in networks and socket programming.
- Learn new programming language “python”.
- Experience in Linux operating system specially “Rasbian” and its terminal commands.
- Web site development.
- Mysql database management system.
- Dealing with sensors and micro controller “raspberry pi”

8.3 Future Work:

After our hard work on this project, and facing many problems during the implementation; we as project team see that the following points may be a good improvement in the future. To make it more reliable and suitable for real life application:

- ✓ Authentication for entering the parking.
- ✓ Expand the idea of collecting the fees in computerized way.
- ✓ Use wireless sensors.
- ✓ Make monitoring system for more security.

References:

[1] http://en.wikipedia.org/wiki/Parking/.

[2] http://en.wikipedia.org/wiki/Sensor/.

[3] http://whatis.techtarget.com/definition/microcontroller/ .

[4] http://www.dummies.com/how-to/content/looking-at-the-android-operating-system0.html/ .

[5] http://en.wikipedia.org/wiki/Android_(operating_system)/.

[6]: http://en.wikipedia.org/wiki/Wireless_USB/.

[7]:Riham Walled Zallom , Jehan Khamayseh , and Maysa Al-Mohtaseb "Smart Car Parking System" , Polytechnic Palestine University, Hebron, 2007.

[8] Shadi Arafeh ,Shery Wazwaz and Mahmoud Al-juneidi , Wireless operated parking system Palestine polytechnic university,Hebron, 2011.

[9] Rafa' Atya & Najat Sanoury "smart parking" ,Al-Najah national university,2010.

[10] \Image processing , Idris M.Y.I, Tamil E.M, Razak Z, Noors N.M, Kin L.W ," Smart Parking System using Image Processing Techniques in Wireless Sensor Network Environment
", Information Technology Journal01/2009;

[11] \Microcontroller, http://en.wikipedia.org/wiki/Microcontroller, 2000.

[12] \Server, http://en.wikipedia.org/wiki/Server_(computing).

[13] Raspberry Pi," http://uk.rsonline.com/web/generalDisplay.html?id=raspberrypi.".

[14] \ 8051microcontroller," http://en.wikipedia.org/wiki/Intel_MCS-51"

[15] \Image processing problems, Amin Kianpisheh, Norlia Mustaffa, Pakapan Limtrairut and Pantea Keikhosrokiani," Smart Parking System (SPS) Architecture Using Ultrasonic Detector ",Universiti Sains Malaysia(USM), Malaysia

[16]\ Raspberry pi, http://www.raspberrypi.org/

[17]\USB dongle, https://www.adafruit.com/products/1012

[18]\ Raspberry pi software, http://en.wikipedia.org/wiki/Raspberry_Pi

[19]\webapplication, http://www.acunetix.com/websitesecurity/web-applications/

[20]\object relational mmodel , http://www.princeton.edu/~achaney/tmve/wiki100k/docs/Object-relational_database.html