**PROJECT IV PROPOSAL TITLE: RESTFUL WEB SERVICE BASED ‘SMART PARKING SYSTEM’ USING RASPBERRY PI.**

Prepared

By

Elvis Peter Hlongwane

Student Number: 202007970

Submitted in fulfillment of the requirements for the degree

**BACCALAUREUS TECHNOLOGIAE: INFORMATION TECHNOLOGY**

**COMPUTER SYSTEMS**

In the

**Department Computer Systems Engineering**

**FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY**

**TSHWANE UNIVERSITY OF TECHNOLOGY**

Supervisor: Professor P.A. Owolawi

**March, 2018**

# DECLARATION

I declare that, this proposal submitted to Tshwane University of Technology for the degree of **BACCALAUREUS TECHNOLOGIAE** Information Technology **Computer Systems** has not previously been submitted for a degree at this or any other university and that it is my work in design and in execution, and that all material contained herein has been duly acknowledged.

|  |
| --- |
|  |

**Abstract**

Visitors of the malls, shopping centres, residential complexes spend a lot of time in parking areas trying to locate empty parking slots. This end up creating many car traffic congestions. Parking management staff must walk around parking areas while performing monitoring and control duties. This is in-effective, and they also add to the problem of moving objects traffic congestions in parking areas.

There is a need for an automation of a parking slots allocation process, remotely monitoring of parking areas, and a need for an online booking of parking slots. These limitations are the focus of this project.

A ‘Restful Service Based Smart Parking System’ using Raspberry Pi 3 microprocessor is developed. The developed system is a five-layered architecture that consist of Client layer, Web service layer, Business layer, Enterprise Information System layer, and Hardware layer. The Client layer resides from end users’ side in the form of a browser or mobile application. It interacts with Web service layer by sending a request and receiving a response in the form of a Json message. Web service layer in turn interacts with Business layer by sending a request and receiving a response in the form of a local method invocation. Web service layer and Business layer reside in the same runtime environment. Business layer interfaces to Enterprise Information System layer through database connectivity mechanism. Business layer in turn interfaces to Hardware layer by sending and receiving JSON request and response messages. Hardware layer comprises of Raspberry Pi3 microprocessor, Ultrasonic sensors, Liquid Crystal Display, and a Potentiometer. Raspberry Pi3 microprocessor act as a CPU. It is installed with Apache Tomcat web server, to enable web server capabilities. It receives and processes JSON request from Business layer and sends Json respond back. The Ultrasonic Sensor is connected to Raspberry Pi3 microprocessor. It sends out a high-frequency sound pulse and then times how long it takes for the echo of the sound to reflect, Raspberry Pi3 microprocessor then uses this time to calculate the distance and work out if parking slot is occupied or not occupied. Using this information, an algorithm to determine if a parking area is completely empty, still has empty parking slots available, or parking is full has been developed and hosted within Raspberry Pi3 microprocessor. The status of the parking area is displayed on Liquid Crystal Display, which is also connected to the Raspberry Pi3 microprocessor. Potentiometer meter is connected to the Liquid Crystal Display to allow the contrast of the display to be adjustable if required.

The results after using the system demonstrate that visitors of the malls, shopping centres, public areas complexes can save a lot of time by remotely booking for parking slots online. Visitors can find parking slots with ease using navigation system that helps them navigate directly to their allocated parking slots. Parking management staff can remotely manage, monitor and control parking areas with ease. The systems help reduce car traffic congestions in parking areas.

**TABLE OF CONTENTS**

|  |  |
| --- | --- |
| **CONTENT** | **PAGE** |

[DECLARATION 2](#_Toc507107520)

[1. CHAPTER 1 6](#_Toc507107521)

[1.1. INTRODUCTION 6](#_Toc507107522)

[1.2. MOTIVATION FOR THE PROJECT 7](#_Toc507107523)

[1.3. PROBLEM STATEMENT 8](#_Toc507107524)

[1.4. AIM 8](#_Toc507107525)

[1.5. OBJECTIVES 8](#_Toc507107526)

[1.6. PROJECT SPECIFICATIONS AND LIMITATIONS 9](#_Toc507107527)

[2. CHAPTER 2 - LITERATURE REVIEW 10](#_Toc507107528)

[2.1. INTRODUCTION 10](#_Toc507107529)

[3. CHAPTER 3 – METHODOLOGY 12](#_Toc507107530)

[3.1. BLOCK DIAGRAM 12](#_Toc507107531)

[3.2. CIRCUIT DIAGRAM 13](#_Toc507107532)

[3.3. DESCRIPTION 14](#_Toc507107533)

[4. CHAPTER 4 - PROJECT PLAN 20](#_Toc507107534)

[4.1. PROJECT COST /BUDGET. 20](#_Toc507107535)

[5. CHAPTER 5 – EXPECTED RESULTS AND CONCLUSION 22](#_Toc507107536)

[5.1. INTRODUCTION 22](#_Toc507107537)

[REFFERENCE 23](#_Toc507107538)

**TABLE OF FIGURES**

[Figure 1.1: TYPICAL CAPMS 7](#_Toc507032071)

[Figure 3.1: BLOCK DIAGRAM OF RWSPMS . 12](#_Toc507032072)

[Figure 3.2: CIRCUIT DIAGRAM OF RWSPMS . 13](#_Toc507032072)

[Figure 3.3: USE CASE DIAGRAM OF RWSPMS . 15](#_Toc507032072)

[Figure 3.4: USER REGISTRATION ACTIVIty DIAGRAM 16](#_Toc507032071)

[Figure 3.5: USER DE-REGISTRATION ACTIVIty DIAGRAM . 16](#_Toc507032072)

[Figure 3.6: USER LOGIN ACTIVIty DIAGRAM . 16](#_Toc507032072)

[Figure 3.7: MAKE A BOOKIng ACTIVIty DIAGRAM . 16](#_Toc507032072)

[Figure 3.8: CANCEL BOOKIng ACTIVIty DIAGRAM . 17](#_Toc507032072)

[Figure 3.9: MONITOR PARKIng AREA ACTIVIty DIAGRAM . 17](#_Toc507032072)

[Figure 3.10: APPLICATION LAYER CLASS DIAGRAM . 18](#_Toc507032072)

[Figure 3.11: RBP3 AND ULTRASONIC CONNECTIONS . 19](#_Toc507032072)

[Figure 4.1: PROJECT PLAN . 21](#_Toc507032072)

**LIST OF TABLES**

[Table 4.1: BuDGET 20](#_Toc507032077)

**ACRONYMS**

|  |  |
| --- | --- |
| **CAPMS** | **Cars Automated Parking management system** |
| **RWSSPS** | **RESTFUL WEB SERVICE BASED ‘SMART PARKING SYSTEM’ USING RASPBERRY PI** |
| **RBP3** | **Raspberry Pi 3** |
| **RFWS** | **Restful Web Services** |
| **CPU** | **Control Processing Unit** |
| **LCD** | **Liquid Crystal Display** |
| **LED** | **Light Emitting Diode** |
| **JEE** | **Java Enterprise Edition** |
| **HTTP** | **Hyper Text Transfer Protocol** |
| **PSO** | **Parking Spot Occupied** |
| **PSNO** | **Parking spot Not Occupied** |
| **JSON** | **JavaScript Object Notation** |
| **JRE** | **Java Runtime Environment** |
| **EIS** | **Enterprise Information System** |

* 1. **CHAPTER 1**
  2. **INTRODUCTION**

Automated systems are an emerging technology that is configured to enable the automatic, remote and/ or local control of the electronics devices. Automation is becoming the integral part of human daily activity and it is growing at a very fast pace and the technology to help achieve automation has advanced and still is. According to Wikipedia [1], automation is the technology by which a process or a procedure is performed without or with less human intervention. There are many systems that are built using this concept, ranging from hardware devices to software processes, and the integration of the two.

According to Wikipedia [1], automated systems are necessary in many instances, we are going to list a few examples below:

Home automation, the systems can help consumers save cost in energy consumptions [2]; they are flexible, easily accessible, and can be operated from anywhere in the world any time, and they are cost effective. In addition to the domestic usage, these systems can also be implemented in industrial buildings, governments, schools, and universities.

Smart Parking systems, which is the focus of this paper, customers can make bookings to use the parking areas of any outlets before they leave their homes, allowing them to use their time effectively. These systems can make use of sensors to calculate the time cars remained in parking areas and then calculate the amount the customers need to pay before they leave parking areas [3]. Customers can also make payments using convenient methods of payment, including traditional Visa, Master card method of payments. These systems can also allow the navigation to the parking slot allocated using navigation systems [4]. Fig. 1[17] depicts a generic design of Smart Parking system:

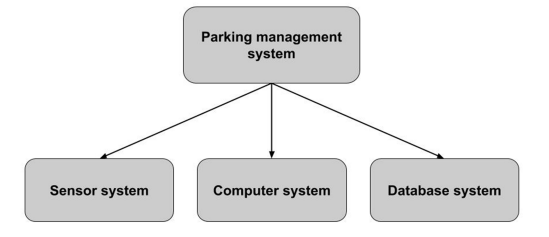


Fig. 1.1 Smart Parking system generic design

Smart Parking systems further explained:

Smart Parking systems collect real time data about available parking spaces of an outlet or public areas, process data, notify customer of the state of the parking on request, and autonomously process the payments by the customers. It involves low cost sensors, cameras, Wi-Fi, microcontrollers and microprocessors.

When Smart Parking systems are deployed, they would enable cars to reduce the time drivers spend driving around in the parking areas looking for empty parking slots. Car drivers looking for vacant parking slots cause more than 30 percent of traffic congestion [12]. Thorat et al also came to the similar conclusion that car drivers looking for vacant parking slots cause 30-40 percent of traffic congestion [16]. Smart Parking systems would help reduce traffic congestion in parking areas, and reduce the car emissions. F.I Shaikh et al [5], in their research stated that one-third of cars reach their destination and start circling around looking for a parking space, thus leading to problems like pollution and traffic congestion. They also stated that researchers have found that in one year, car cruising for parking created the equivalent of 38 times trips around the world, burning 1.7 lakh litres of fuel and producing 730 tons of CO2. H. Hao et al [6], indicated in their analysis that carbon dioxide emissions from global passenger cars were 2810 megatons in 2013, accounting for about 8.7% of global energy-related carbon dioxide emissions. As they further indicate that global car sales will more than double by 2050, this poses a serious harm to our environment. Aditya Basu [15] in his research estimated that 2,20 000 gallons of fuel can be saved by year 2018 and further 3, 000, 000 gallons can be saved by 2050, by successfully implementing Smart Parking systems.

The statistics above highlights the a need for an automation of a parking slot allocation process, remotely monitoring of parking areas, and a need for an online booking of parking slots. To address this need, ‘Restful Service Based Smart Parking System’ using Raspberry Pi 3 microprocessor is developed. The developed system is a five-layered architecture that consist of Client layer, Webservice layer, Business layer, Enterprise Information System layer, and Hardware layer. The Client layer resides from end users’ side in the form of a browser or mobile application. It interacts with Web service layer by sending a request and receiving a response in the form of a JSON message. Web service layer in turn interacts with Business layer by sending a request and receiving a response in the form of a local method invocation. Web service layer and Business layer reside in the same runtime environment. Business layer interfaces to Enterprise Information System layer through database connectivity mechanism. Business layer in turn interfaces to Hardware layer by sending and receiving JSON request and response messages. Hardware layer comprises of Raspberry Pi3 microprocessor, Ultrasonic sensors, Liquid Crystal Display, and a Potentiometer. Raspberry Pi3 microprocessor act as a CPU and is installed with Apache Tomcat web server, to enable web server capabilities. It receives and processes Json request from Business layer and sends JSON respond back. The Ultrasonic Sensor is connected to Raspberry Pi3 microprocessor. It sends out a high-frequency sound pulse and then times how long it takes for the echo of the sound to reflect, Raspberry Pi3 microprocessor then uses this time to calculate the distance and work out if parking slot is occupied or not occupied. Using this information, an algorithm to determine if a parking area is completely empty, still has empty parking slots available, or parking is full has been developed and hosted within Raspberry Pi3 microprocessor. The status of the parking area is displayed on Liquid Crystal Display, which is also connected to the Raspberry Pi3 microprocessor. Potentiometer meter is connected to the Liquid Crystal Display to allow the contrast of the display to be adjustable if required.

Figure 1.1 depics RWSSPS System environment

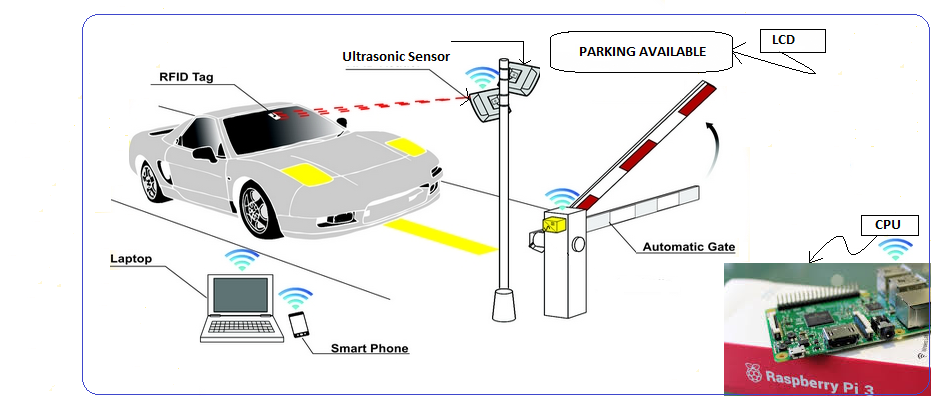


Figure 1.2 RWSSPS environment.

* 1. **MOTIVATION FOR THE PROJECT**

There are some issues that we have reviewed and saw the need for designing RWSSPS

1. *Environmental issues*

The more time spent driving in a parking area, the more the carbon dioxide emissions. The solution will help combat the scourge of global warming.

1. *Time management issues*

Time taken by the customers looking for empty parking slot will be reduced as they can be guided directly to their parking slot without any pain of getting lost and causing traffic congestions in the parking areas.

1. *Cost issues*

If customers know exactly where to park in the parking area, they can drive directly to their allocated parking slots without any unnecessary extra trips, thus saving on gasoline used by their vehicles. If customers spend less time in parking areas, it will convert into more time in shopping centers doing business, therefore more revenues for business owners and the government(in the form of tax collections). Alternatively, less time spent in parking might convert to less time spent in shopping centers, creating more volume for number of cars and customers driving in.

1. *Effort issues*

Maintenance team have to walk around monitoring parking areas to determine if there’s still an empty parking slots or whether parking area is full, manually places and remove ‘Parking Full/Available’ signs. This is a lot of effort and energy wasted as this can be automated.

* 1. **PROBLEM STATEMENT**

F.I Shaikh et al [5], in their research stated that one-third of cars reach their destination and start circling around looking for a parking space, thus leading to problems like pollution and traffic congestion. Customers are not able to make a booking for using parking areas in shopping complexes. They sometimes drive all the way only to find that all the parking slots have been occupied. This leads to illegal parking by car drivers and they end up being fined some amount of money, or incur some prescribed penalties as results.

Maintenance team have to walk around monitoring parking areas to determine if there’s still an empty parking slots or whether parking area is full, manually places and remove ‘Parking Full/Available’ signs. This is a lot of effort and energy wasted as this can be automated.

The solution is proposed to design RWSSPS to help solve these problems.

* 1. **AIM**

The aim of this project is to automate public parking slots booking and allocation, controlling, monitoring and management of cars in car parking areas.

* 1. **OBJECTIVES**

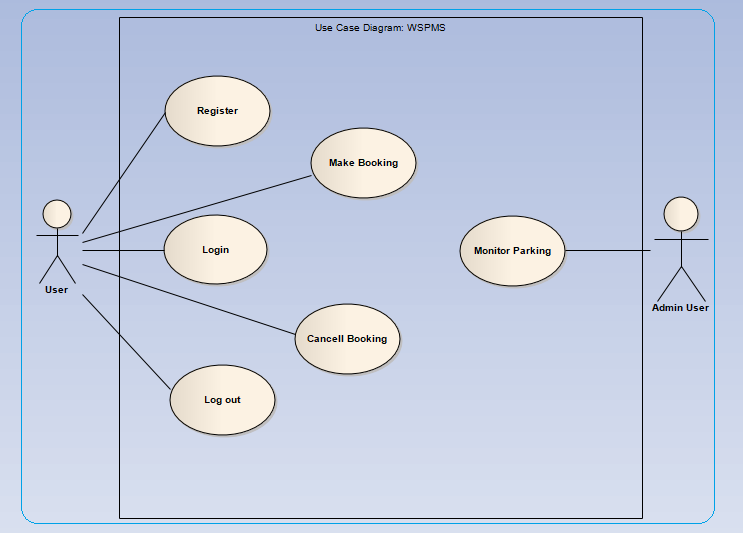
The purpose of this project is to allow car owners or customers to remotely book public parking slots in shopping complexes, and also allow shopping complexes maintenance team to control, monitor and manage cars in car parking areas. This will be achieved by implementing the following tasks:

1. Review different techniques used to design APMS and explain the limitation of each
2. Design RWSSPS and carry out circuit analysis of each section of the project
3. Simulate RWSSPS design using appropriate electronic design tool/software
4. Do cost analysis on the project
5. Package the project for demonstration for potential investors.

1.5.1 USE CASE

**Use case diagram**

The purpose of software development is to produce software by solving problems, but before the problem can be solved they need to be identified first. The problems identified in this work are those related to the RWSSPS. But this doesn’t really tell us much about what activities happens in the system? Who interacts with the system? What are system boundaries? These are called ‘use cases’. To make up the fully functional system, RWSSPS solution have the following use cases as seen in figure 3.2

Figure 3.3 Use case diagrams.

* 1. **PROJECT SPECIFICATIONS AND LIMITATIONS**

The specifications of this project are:

1. Any web browser or android application can be used to access the client application.
2. Central JEE container must be installed – this interfaces the client application layer and the hardware layer.
3. Hardware layer is made up of RBP3 Model, which is the size of the business card.
4. 2 Ultrasonic sensor and LCD are connected to the RBP3 Model
5. Voltage drawn by hardware devices cannot exceed 5V
   1. **CONTRIBUTION**

The primary objective of the RWSSPS is to automate public parking slots booking and allocation, controlling, monitoring and management of cars in car parking areas. So RWSSPS will help create convenience and safety environment for car drivers, convenience for shopping complexes management team, reduce environmental air pollution, and generate more revenues for both businesses and government.

RWSSPS contributes to different stakeholders in the form of convenience and/or revenues. Stakeholders involved are,

1. Car drivers/Customers

These stakeholders gain conventions by having the capability to reserve a parking spot beforehand, therefore saving on time and gasoline.

1. Shopping centers owners

The less time car drivers spend in parking areas the more time they will spend doing business inside shopping centers. Alternatively less time they will spend overall stay in the shopping centers, and therefore create space for other customers, thus creating more customer volume flowing in and out the shopping centers.

1. Business owners

The less time car drivers spend in parking areas the more time they will spend doing business inside shopping centers. Alternatively less time they will spend overall stay in the shopping centers, and therefore create space for other customers, thus creating more customer volume flowing in and out the shopping centers coming to do business.

1. Government

The more business is done the more economic activities, which is good for SARS in terms of revenue collections. This results to good economic growth of the country, which is what any government is striving for.

1. Community

If country’s economy is doing well, the community benefits in terms of number of jobs being created by both public and private sector. Communities also benefits from the healthy environment they are living in, the less carbon emission the better for the environment and the community.

1. **CHAPTER 2 - LITERATURE REVIEW**

**INTRODUCTION**

Automation is the use of scientific principles and techniques to develop machines that would perform operations that would normally be performed by humans. The etymology of the word itself is said to have been credited to having been first used by Delmar S. Harder in 1935, who was a plant manager for General motors; however the concept of automation is one that RWSSPS been around for millennia. One of the earliest instances of automation was credited to one of the Greek inventors by the name of Hero in 50 A.D, who is said to have created a system that would open temple doors when a priest lit a fire, in an automated manner [7].  
  
There are many instances and reasons for automation, but the main reason is to optimize productivity while minimizing effort of performing an operation. In the Industrial sector, automation is applied for increased productivity at less human labor than would be necessary if all the labor required was manual. This, however, should not be confused with mechanization, but rather as a step beyond; for mechanization involves machinery that could be used by human operators to assist in works that would require more muscular efforts, while automation minimizes or eliminates human sensory while providing speed and repeatability.

**HISTORY OF ACPMS**

According to Ramesh S. R, et al [8], automated parking system was actually first developed in 1925 by Max Miller in New York City. The designs original purpose was to simply lift a vehicle off the ground, such as in the case of a stalled or broken down car on a street. Ramesh S. R, et al claim that this system was never used.

According to [9], it was 1941 when O.A. Light created a device that allowed three cars to park vertically, three on each side for a total capacity of six. A year later, E.W. Austin invented the automated garage. His invention became the leader in automated parking during the 40s, 50s and 60s. These systems were called Bowsers, Pigeon Holes and Roto Parks [9].

In 1964, Eric Jaulmes invented what is most similar to the automated parking management systems of today. His system had a valet drive the car into an elevator. The elevator would then take the car to a predetermined spot and the valet would park the car in that space. Then on the return down, if it had been requested, the valet would stop at another spot to get a car to be returned. At the same time, the three former systems were revitalized to remove the valet altogether allowing the lift to tip the car into place and the opposite on retrieval [9].

By the mid 60's (and released in 1968), a "Vert-a-Park” system was developed by Bob Lichti, resembling a ferris wheel, the system allowed 22 cars to be parked in the same horizontal space of about two and a half; meanwhile it took up 90 feet vertically [9].

**SIMILAR WORK**

Further development and design has been made to continually improve the automated car parking systems. Lot of these systems has been developed using various technologies ranging from RFIDs, Sensor Network, Embedded System, MultiAgent Parking System, IOT, number plate extraction system, Automatic number plate recognition (ANPR) system.

Similar work has been done by various researchers as listed below:

* [4] proposed 'Automatic Smart Parking System using Internet of Things (IOT)'
* R. Sharma et al [10], proposed 'Automated Parking and Security System'
* Gavali et al [11], proposed 'Smart Parking System Using the Raspberry Pi and Android'
* N Moses et al [12], proposed ‘Smart Parking System for Monitoring Vacant Parking’
* S.S. Deshpande, and R.S. Gound [13], also proposed ‘An approach for smart parking system based on cloud using IoT’

The work of these researchers is appreciated and welcome, but none of these researchers addressed the issues of using scalable and flexible technologies like web services. This raises a concern once the system is developed and deployed to production for use, as their work designed the system with one standard user interface, either web, mobile application or fixed installed application only, but never combination of the three possibilities. Other issue these researchers failed to address is the ability to access the system remotely.

The proposed work will address these issues. RWSSPS is RFWS; therefore it can be accessed from anywhere, anytime by any kind of a client application.

1. **CHAPTER 3 – METHODOLOGY**

RWSSPS is a five-layered architecture and they are:

1. Client Layer
2. Web service layer
3. Business/Application layer
4. EIS layer
5. Hardware layer

The sketch below, figure 3.1 depicts the architectural layers of RWSSPS.

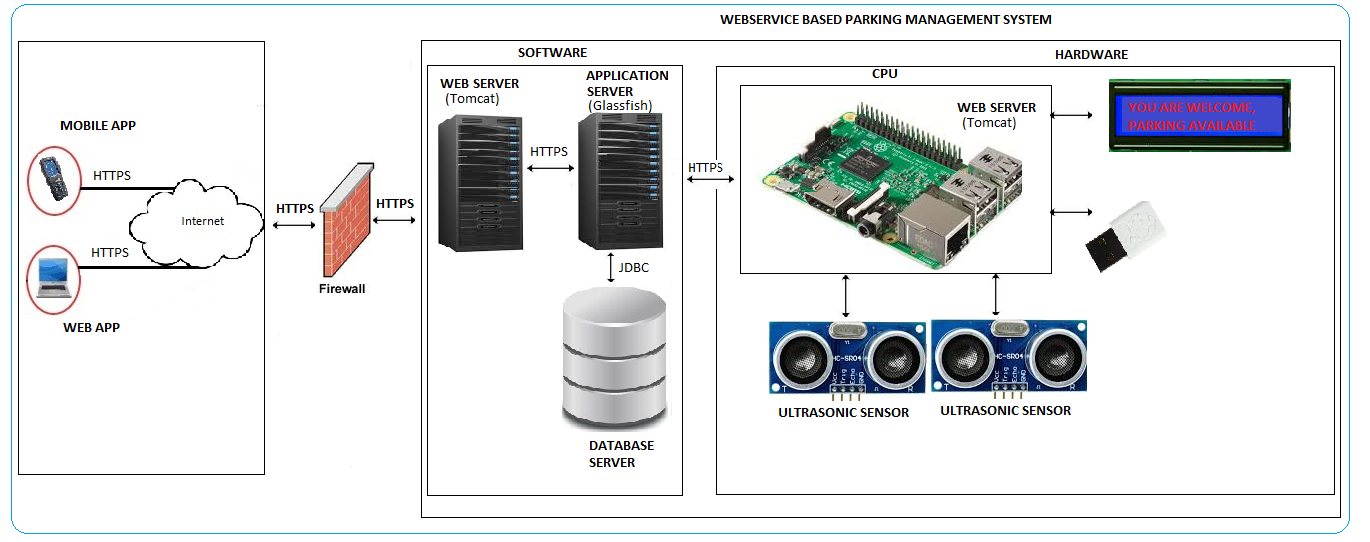


Figure 3.1 Architectural layers of RWSSPS

These layers are categorized into hardware and software categories. Client, Web service, Business, and EIS layers fall under software category, and the hardware layer falls under hardware category.

* 1. **LAYERS**

The system processes the requests from left to right, and the responses from right to left. We discuss the flow below by looking at how each layer handle their requests and responses.

3.1.1. **Client (Web/Mobile) layer**

This layer represents the interface layer (browser or mobile application) that users use to perform system’s activities defined in section 3.1.1. It interacts with Web service layer by sending requests and receiving responses in the form of a JSON messages. It is responsible for serving user interface via the HTTP(s) protocol to clients. The Web server sends out web pages

in response to requests from browsers or mobile apps. A request is generated when a user clicks a link on a web page in the browser or mobile app. The request is then forwarded to the application layer and it wait for the application server response. The request and response is HTTP based.

This layer deploys the following **technologies**:

3.1.2.1 Angular6

The Client layer process involves:

*Process*:

1. User logs into the RWSSPS application through a personal computer or mobile device via a web browser, or mobile application

3.1.2. **Web service layer**

The Web service layer resides in RWSSPS web server. It receives the HTTP requests from the Client layer and sends HTTP responses back in the form of JSON messages. Web service layer in turn interacts with Business layer by sending a requests and receiving a responses in the form of a local method invocation. Web service layer and Business layer reside in the same JRE.

This layer deploys the following **technologies**:

3.1.2.1 Jersy

*Process*:

1. Web service layer receives HTTP request from Client layer in the form of JSON messages
2. Process request
3. Forward the request to Application layer for further processing by invoking local method call
4. Receive the local method call-response from Application layer
5. Transform the response to JSON message
6. Send back the HTTP response to Client layer in the form JSON message

3.1.3 **Application layer**

Application layer interfaces to EIS layer through database connectivity mechanism. Application layer also interfaces to Hardware layer by sending HTTP requests and receiving HTTP responses in the form of JSON messages.

**Application layer**

Application layer is the integrator of the entire RWSSPS system. It integrates all the layers in the system. It integrates with Client layer and Hardware layer using HTTP protocol. It also integrate with EIS using JDBC.

This layer deploys the following **technologies**:

3.1.3.1 JEE

3.1.3.2 Glassfish application server

The following diagram, figure 3.3.1 illustrates the class diagram of the application layer in the system.

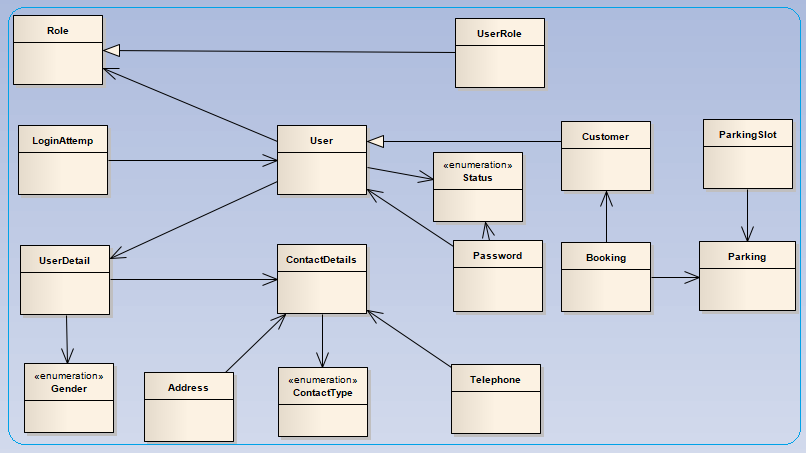
****

Figure 3.10:Application layer class diagram

The application layer process flow is as follows:

*Process*:

1. Application layer receives method call from Web service layer
2. Process the method call
3. Send a CRUD operation request to EIS layer when applicable
4. Receive and process a CRUD operation response from EIS layer if applicable
5. Send a request to Hardware layer when applicable
6. Receive and process a response from Hardware layer if applicable
7. Send back the method call-response to Web service layer

3.1.4 **EIS layer**

This layer makes use of SQL Database to store data. The database is secured behind the firewall, and use the roles based mechanism to authenticate and authorize users. This layer integrates with the application layer through JDBC. The system uses Application layer class diagram in fig 3.7 for database entities.

This layer deploys the following **technologies**:

3.1.4.1 JPA

3.1.4.2 SQL

3.1.4.2 MySQL

*Process*:

1. EIS receives a CRUD operation request from Application layer
2. Process a CRUD operation request
3. Send back CRUD operation response to Application layer

3.1.4 **Hardware layer**

This layer represents the hardware components of RWSSPS. Hardware layer comprises of Raspberry Pi3 microprocessor, Ultrasonic sensors, Liquid Crystal Display, and a Potentiometer.

* RBP3

The RP3 is a very powerful computer having the dimensions of a business card [14].

In RWSSPS, RBP3 is the CPU of the hardware layer. Other components are connected and controlled from here. It also communicates with JEE application layer.

* Ultrasonic Sensor

The ultrasonic sensor is a distance measurement sensor, which uses ultrasonic waves. The speed for the ultrasonic waves is taken to be 1540 m/sec and using this and the time required the wave to go and come back to the sensor the distance is measured [3].

Ultrasonic has four pins, which will be connected to the RP3,

Pin1: will be connected to 5V

Pin2: TRIGGER, will be connected to one of the GPIOs and triggers the sensor

Pin3: ECHO, will be connected to one of the GPIOs. This pin outputs pulse in the form of +5v. This amount of voltage cannot be connected to RP3 directly, so a voltage divider circuit must be built. To achieve this 2 resistance, 1k ohms and 2k ohms must be connected in series used to reduce the ECHO output to 3.3V that can be handled be RP3. Fig 3.9 below depicts the sample RP3-Ultrasonic connection.

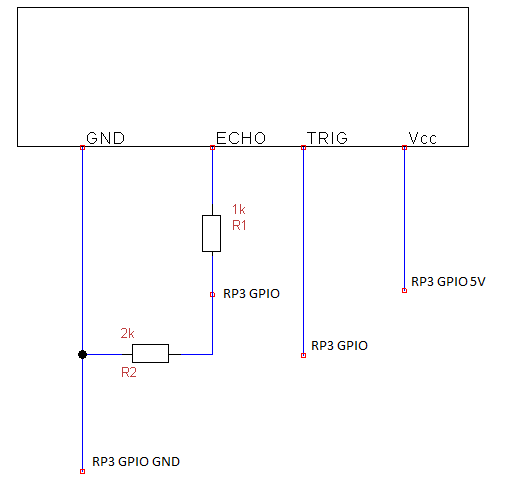


Figure 3.11 Ultrasonic RP3 connections

Applying the voltage reduction formula, the value of R2 was calculated,

Vout = Vin x (R2/(R1 + R2))

(Vout/Vin) = R2/(R1 + R2)

Since we know the ultrasonic ECHO pulse output of +5V, and RP3 GPIO expects 3.3V, we will assume 1st resistence of 1k ohms and calculate the second closest we can use, therefore

3.3/5 = (R2/(1000 + R2)),

Therefore R2 = 1941 ohms

PIN4: GND will connect to ground.

* LCD

LCD will be used to communicate to the customers the status of the parking. It is powered up using RP3 +5v pin2 through the 1000uF capacity.

* Power Supply

Power supply used here will be 5v to power up the RP3. This can be done in 2 ways:

1. Power will be sourced directly from USB adapter

The following diagram, figure 3.9 illustrates the circuit diagram of the system.

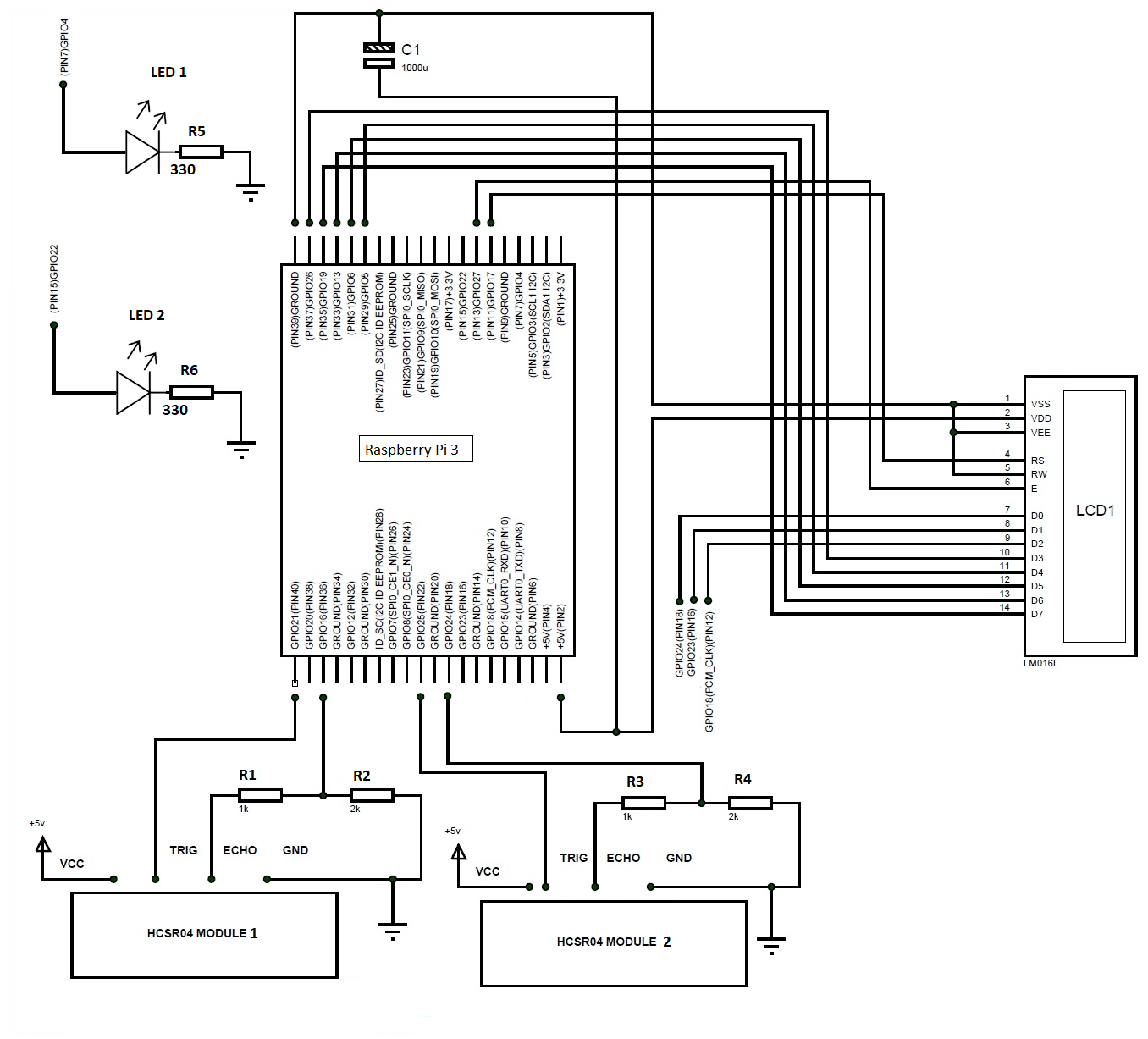


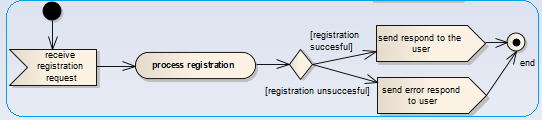
Figure 3.2 Circuit diagram of the RWSSPS

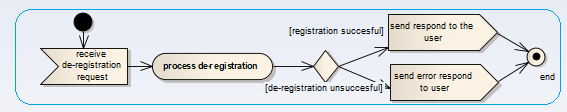
* 1. **RWSSPS PROCESS FLOW**

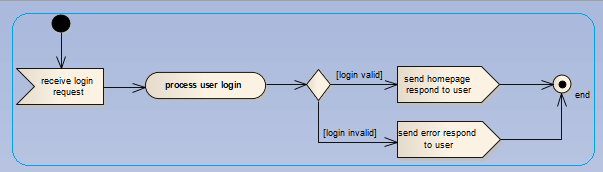
To illustrate the RWSSPS process flow, we create activity diagrams for Registration, ‘De-Registration’, ‘Make a booking’, ‘Cancel a booking’, and ‘Monitor Parking’ use cases.

**Activity diagrams:**

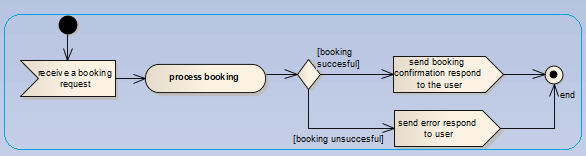
The following activity flow diagrams illustrate the visual process of uses cases. The following use cases are used to give an example.

**Use case: Registration use case**Figure 3.4 Registration activity diagram

**Use case: De-registration use case** ****Figure 3.5 De-registration activity diagram

**Use case: Login **Figure 3.6 Login activity diagram

**Use case: Make booking** **use case**

Figure 3.7 Make booking activity diagram

**Use case: Cancel booking** **use case**

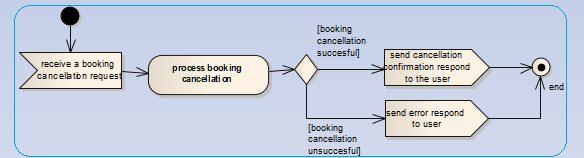


Figure 3.8 Cancel booking activity diagram

**Use case: Monitor parking use case**

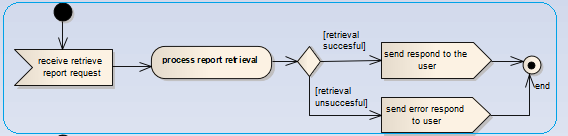


Figure 3.9 Monitor parking activity diagram

1. **CHAPTER 5 – EXPECTED RESULTS AND CONCLUSION**

**Expected results**

1. RWSSPS users will have to first register before they can start using the system.
2. Registration process will take place through the Web or Android user interface application.
3. Once users are registered they can login into the system, selected a shopping center, browse the parking area for available parking slots. Users can then be able to make a booking if parking slots are available.
4. Once they have a booking confirmed they can drive thought the shopping center, and directly through their parking allocated parking slot.
5. When parking in an allocated slot, red LED will be lit indicating that the parking slot is occupied
6. The green LED will be lit once the car vacates the parking slot.
7. When the parking is full, LCD will display the ‘Parking Area Full’ message in the parking area entrance, otherwise ‘WELCOME' will be displayed.

**Conclusion**

The RWSSPS is associated with high rapidly changing technology in these current times. Such applications with the automated parking management system would help the car drivers to save a lot of time, help shopping complexes management teams to monitor, and effectively manage parking areas. RWSSPS is a flexible system without boundaries; it can be operated from anywhere, at any time, and at a lower cost. RWSSPS is not limited to shopping complexes; it can be used in firms, companies, schools, churches, or any building with public parking areas.

**REFFEENCES**

[1]Wikipedia, “Automation.” Internet: https://en.wikipedia.org/wiki/Automation, Mar 09, 2018 [Mar. 05, 2018].

[2]Wikipedia, “Home automation.” Internet: https://en.wikipedia.org/wiki/Home\_automation , Mar 09, 2018 [Mar. 05, 2018].

[3] PI, M.A.E Viegas, Z. Rajani, V. Vaz, J. Kumari. (2017, Jan.). “Smart Reverse Parking using Raspberry.” International Journal of Science Technology & Engineering. [On-line]. 3(07), pp. 99-102. Available: http://www.ijste.org/articles/IJSTEV3I7070.pdf [Feb. 12, 2018].

[4] S.R Basavaraju. (2015, Dec.). “Automatic Smart Parking System using Internet of Things (IOT).” International Journal of Scientific and Research Publications. [On-line]. 5(12), pp. 629-632. Available: http://www.ijsrp.org/research-paper-1215/ijsrp-p4898.pdf[Feb. 12, 2018].

[5] F.I Shaikh, P.N Jadhav, S.P Bandarkar, O.P Kulkarni, N.B. Shardoor. (2016, Apr.). “Smart Parking System Based on Embedded System and Sensor Network.” International Journal of Computer Applications. [On-line]. 140(12), pp. 45-51. Available: http://www.ijcaonline.org/research/volume140/number12/shaikh-2016-ijca-909532.pdf [Feb. 12, 2018].

[6] H Hao, J Sarkis, Y Geng. (2016, Apr.). “Carbon footprint of global passenger cars: Scenarios through 2050.” Energy. [On-line]. 101, pp. 121-131. Available: https://www.researchgate.net/publication/297791279\_Carbon\_footprint\_of\_global\_passenger\_cars\_Scenarios\_through\_2050 [Feb. 12, 2018].

[7]History, “Types Of Automation.” Internet: http://science.jrank.org/pages/679/Automation.html, Mar 09, 2018.

[8] S.R Ramesh, I Muthuveera, K. Naveenkumar, S. K Tamilarasu. (2017, Mar.). “Rotary Parking System for Multi-Purpose Vehicle.” International Conference on Latest Innovations in Applied Science, Engineering and Technology. [On-line]. pp. 154-159. Available: https://www.ijraset.com/fileserve.php?FID=6964 [Feb. 12, 2018].

[9]KLAUS multiparking Inc, “The History Behind the Automated Parking Management System.” Internet: http://www.parkingmanagementsystem.com/, Mar 10, 2018.

[10] R Sharma, K.P Jasani, A Jaiswal, P.R Bais. (2017). “Automated Parking and Security System.” International Journal of Advance Research, Ideas and Innovations in Technology. [On-line]. 3(2), pp. 709-715. Available: https://www.ijariit.com/manuscripts/v3i2/V3I2-1370.pdf [Feb. 12, 2018].

[11] A. Gavali, P Kunnure, S Jadhav, T Tate, V Patil. (2017, Jun). “Smart Parking System Using the Raspberry Pi and Android.” International Journal of Computer Science and Information Technology Research. [On-line]. 5(2), pp. 48-52. Available: www.researchpublish.com [Feb. 12, 2018].

[12] N Moses, Y.D Chincholkar. (2016, Jun). “Smart Parking System for Monitoring Vacant Parking.” International Journal of Advanced Research in Computer and Communication Engineering. [On-line]. 5(6), pp. 717-720. Available: https://www.ijarcce.com/upload/2016/june-16/IJARCCE%20159.pdf [Feb. 12, 2018].

[13] SS. Deshpande, RS. Gound. (2016, Jun). “An approach for smart parking system based on cloud using IoT.” Proceedings of the Second International Conference on Research in Intelligent and Computing in Engineering. [On-line]. 10(2300- 5963), pp. 161-163. Available: https://annals-csis.org/Volume\_10/drp/pdf/80.pdf[Feb. 12, 2018].

[14] SS. Deshpande, RS. Gound. (2014, Nov). “A Review Paper on Raspberry Pi.” International Journal of Current Engineering and Technology. [On-line]. 4(6). Available: http://inpressco.com/wp-content/uploads/2014/12/Paper53818-3819.pdf [Feb. 12, 2018].

[15] Aditya Basu – Happiest Minds

[16] Thorat – IoT Based Smart Parking Systems

[17] Prashanth – A review