Abstract

We hereby have the utmost privilege to thank our Software Group Project teachers **Prof. Divyesh B. Patel** for their support and guidance. We would also like to thank to **Mr. A.D. Patel**, the Principal and **Mr. Amit Ganatra**, the Dean for giving us this opportunity to present this project. We would also like to thank our parents and my team for supporting and helping us to give my ideas, efforts in this project.

Acknowledgement

In Our project we have provided a Ultrasonic sensor to get the information about if car is parked or not in particular slot. And we have also provided Esp8266(wifi-module) to send the data to the cloud with the help of arduino. Our Project Car parking system using Arduino is basically developed for the special purpose to view that how much number of parking slots are filled/vacant in the parking arena so, user can view if there is slot available for parking or not. With the help of Arduino and Esp8266 the data will be updated to cloud (thingspeak) as soon as car leaves/enters the arena.

Table of Contents

1.0 INTRODUCTION

1.1 Project Summary

Car parking system using Arduino is basically developed for the special purpose to view that how much number of parking slots are filled/vacant in the parking arena. The data will be updated as soon car leaves/enters the arena through Esp8266 to cloud. There is a restriction that if the parking slots are filled then the vehicle is not allowed to enter the parking arena.

1.2 Purpose

The purpose of the project is to early detect the no. of filled/vacant slots in the parking arena to avoid the wastage of Users time for finding the parking slot for their vehicle.

1.3 Scope

The scope of this project is to make automated system to detect the no. of filled/vacant slots in the parking arena sending each data updation to cloud through the Esp8266.

1.4 Objective

- To design and build up a prototype of a car parking system.
- To acknowledge how to program Arduino and make it on any system.
- To send data to cloud using Esp8266.

2.0 PROJECT MANAGEMENT

2.1 Project Planning

2.1.1 Project Development Approach and Justification

The project development approach of this project is **agile model**. In Agile model every project needs to be handled differently and the existing methods need to be tailored to best suit the project requirements. In Agile, the tasks are divided to time boxes (small time frames) to deliver specific features for a release.

Iterative approach is taken and working software build is delivered after each iteration. Each build is incremental in terms of features, the final build holds all the features required by the customer.

2.1.2 Project Effort and Time, Cost Estimation

For Effort estimation of project,
Effort = 2.4 * (KLOC) ^1.05

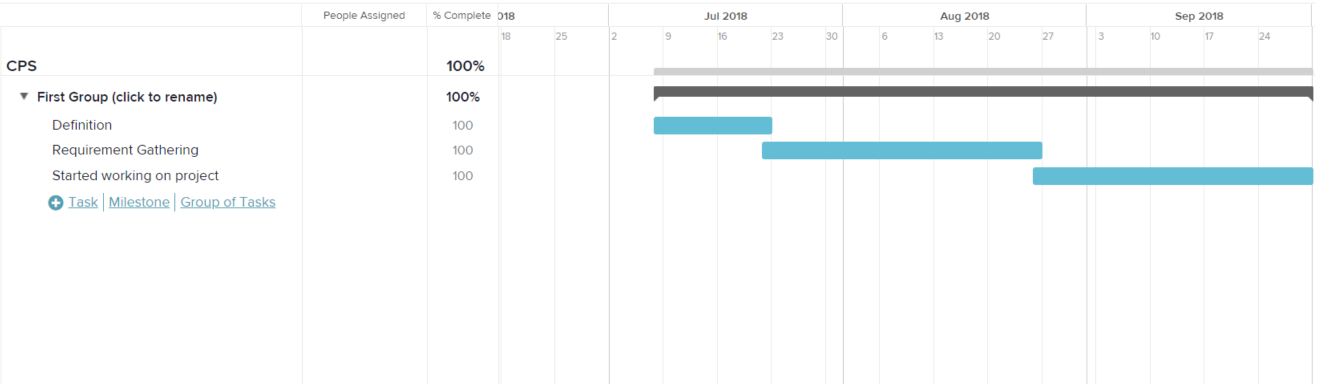
Effort = 2.4 * 0.069 ^ 1.05 = 0.151 PM

For Time estimation of project,
Time = 2.5 * Effort ^ 0.38
Time = 2.5 * 0.151 ^ 0.38 = 0.70 M

2.1.3 Roles and Responsibilities

Designer	Jayesh Bhutiya
Developer, Coder	Tarun Adavani
Documentation, Tester	Shivam Bhanvadia

2.2 Project Scheduling



	People Assigned	% Complete
CPS		100%
▼ First Group (click to rename)		100%
Definition		100
Requirement Gathering		100
Started working on project		100

3.0 SYSTEM REQUIREMENTS STUDY

3.1 User Characteristics

System Admin can choose the users of this system.

3.2 Hardware and Software Requirements

Hardware Requirements:-

1. **Arduino uno r3**
2. **Esp8266**
3. **Ultrasonic sensor**
4. **Jumper wires**
5. **Bread board**
6. **Led**

CPU	Intel core-i3 5005U,2.0GHZ
RAM	4 GB DDR3
HDD	1 TB

Software Requirements:-

Programming language	C language
IDE	Arduino IDE 1.8.7

3.3 Assumptions and Dependencies

Assumptions:-

- System will work correctly in worst condition with 100% accuracy.
- The coding should be error free.
- The system should be user-friendly.
- The system should run 24-hours.
- Fast response.

Dependencies:-

- The specific hardware and software due to which the product will run.
- The end-user should have proper understanding of the system,
- System will use true data at this time,
- System will predict the no. of filled/vacant slots in parking arena.

4.0 SYSTEM ANALYSIS

4.1 Study of Current System

- At present the system is able to handle couple of parking slots in future will try to handle this system for larger no. of car parking.
- As per the present system, Led will notify the customer that whether the parking slots are empty/filled.
- At present the system is updating data to thingspeak.

4.2 Problem and Weaknesses of Current System

- This system will not be able to handle large no. of car parking at a time.
- There is no facility to pre-book the parking slot.

4.3 Requirements of New System

4.3.1 Functional Requirements

- Good User interface to input values
- System should predict the car parking for customer.

4.3.2 Non Functional Requirements

- The coding should be error free
- The system should be user-friendly and good GUI
- The system should have efficient security
- The coding should have proper comments

4.4 Feasibility Study

4.4.1 Does the system contribute to the overall objectives of the organization?

- Yes, System will do.

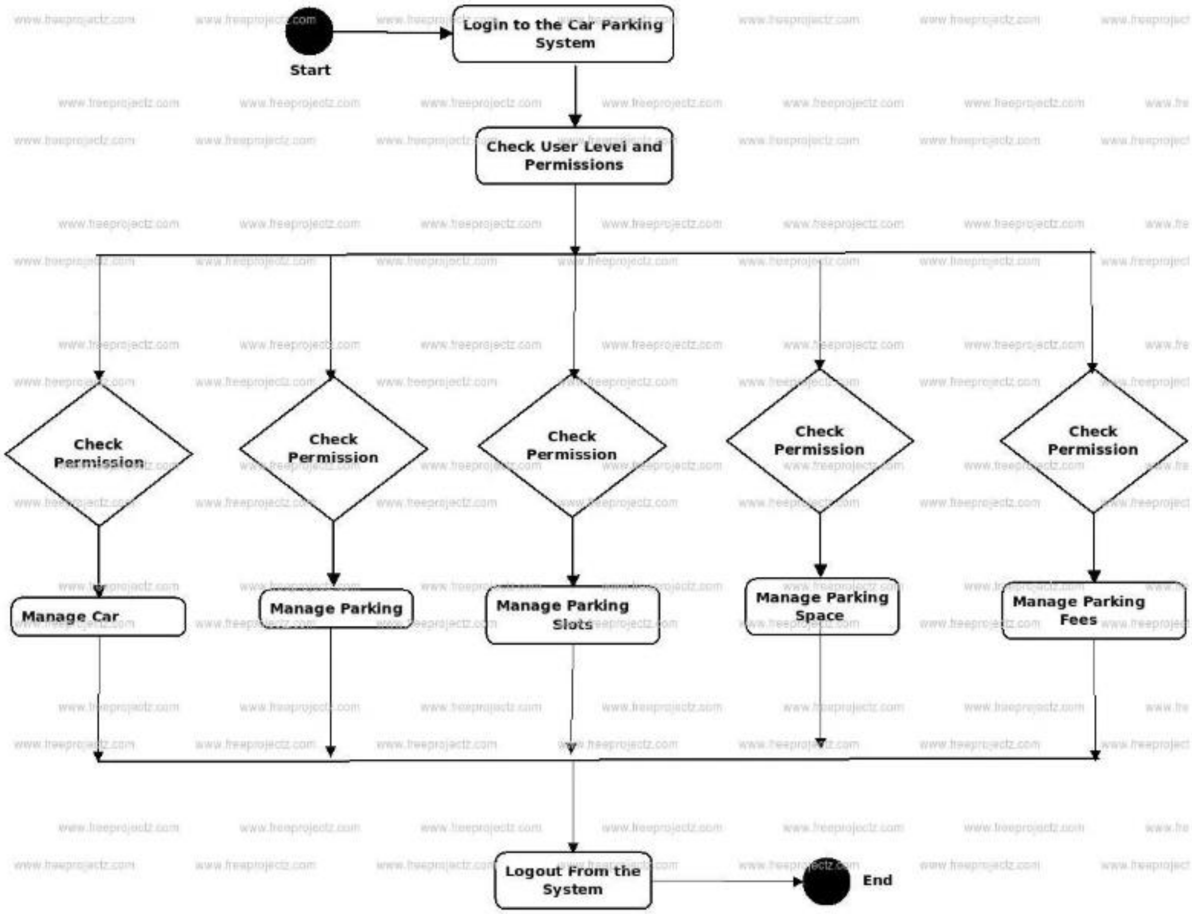
4.4.2 Can the system be implemented using the current technology and within the given cost and schedule constraints?

- Yes, System will be implemented using the current technology and within the given cost and schedule constraints.

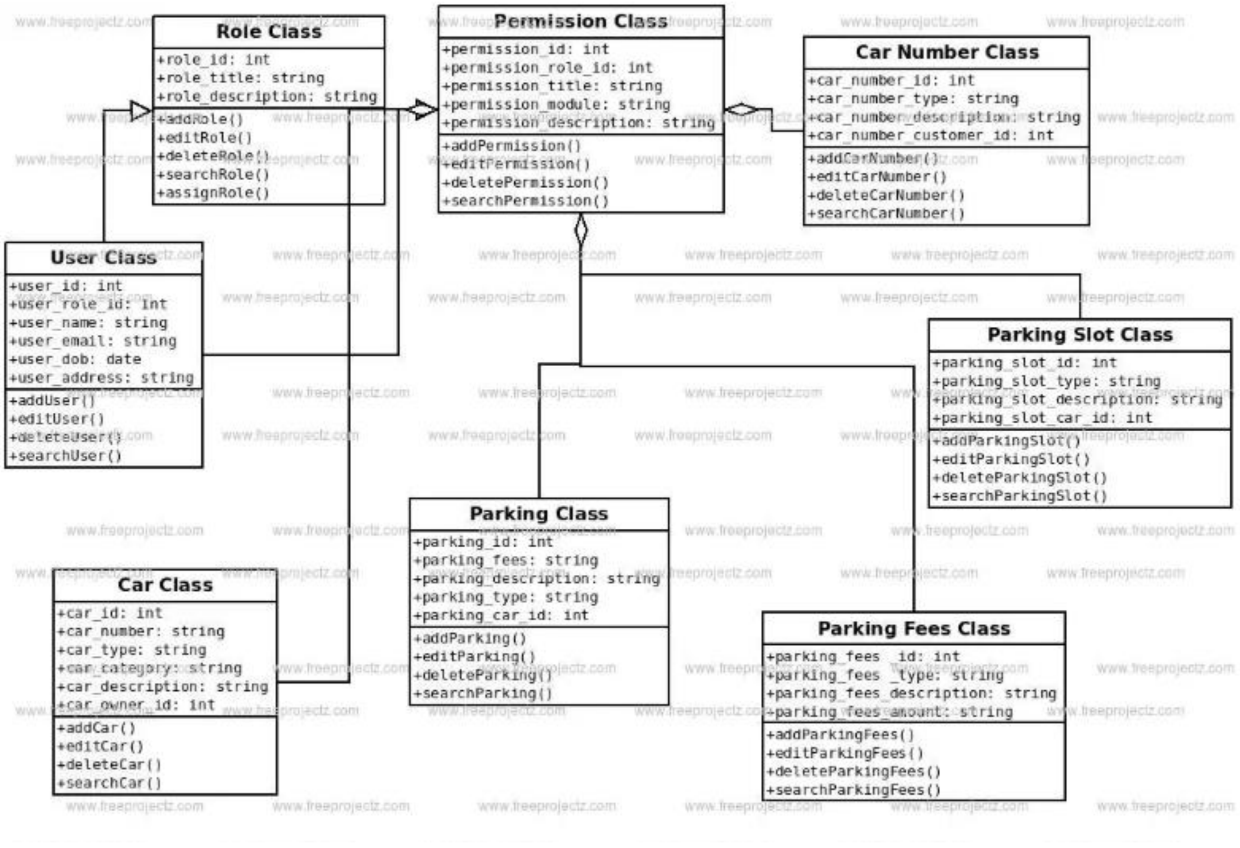
4.4.3 Can the system be integrated with other system which are already in place?

- Yes, System is able to integrate with other systems.

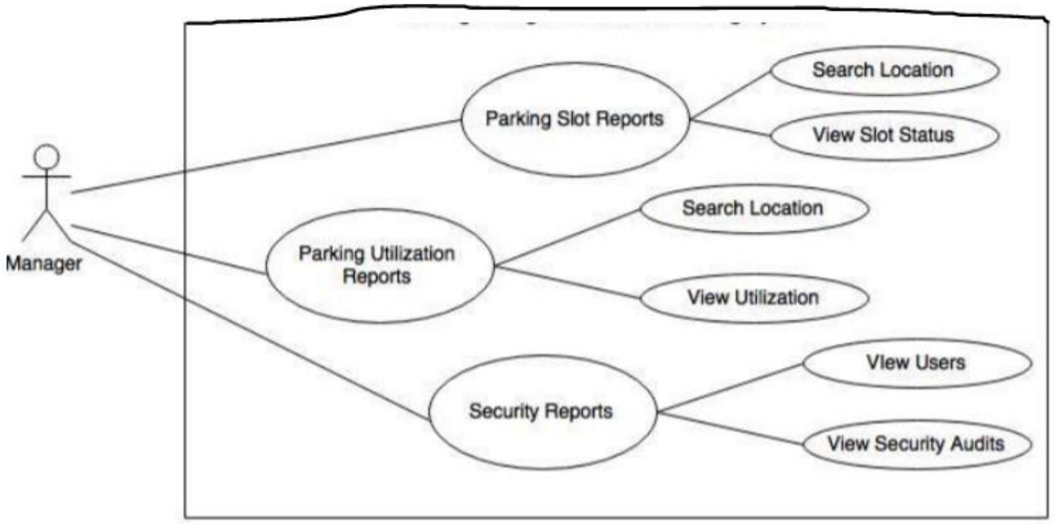
4.5 Activity/Process In New System (Use event table)



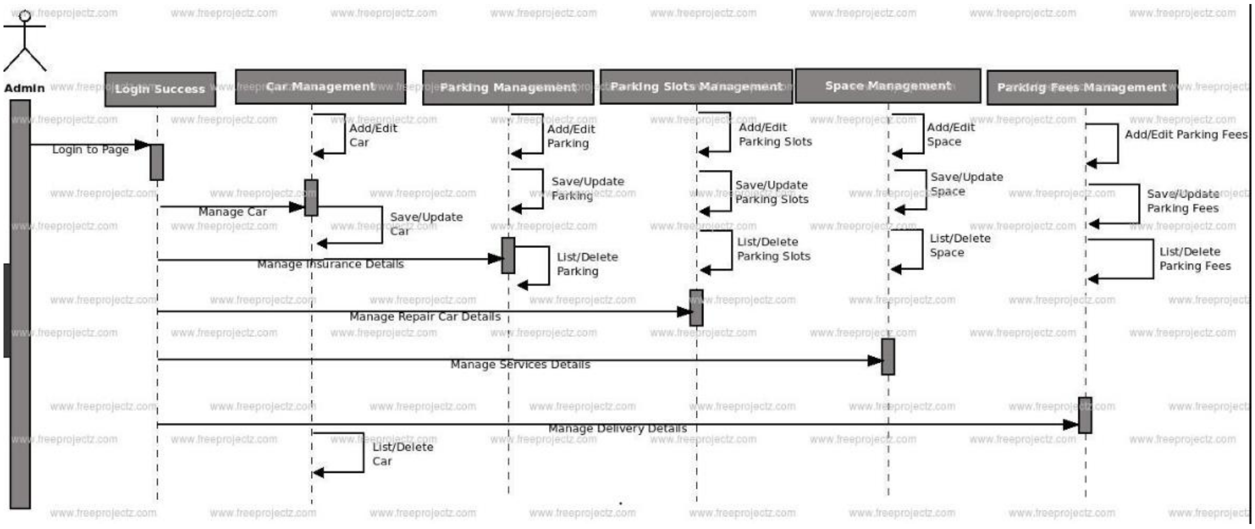
4.6 Class Diagram



4.7 System Activity(Use case and/or scenario diagram)



4.8 Sequence Diagram



www.freeprojectz.com

5.0 SYSTEM DESIGN

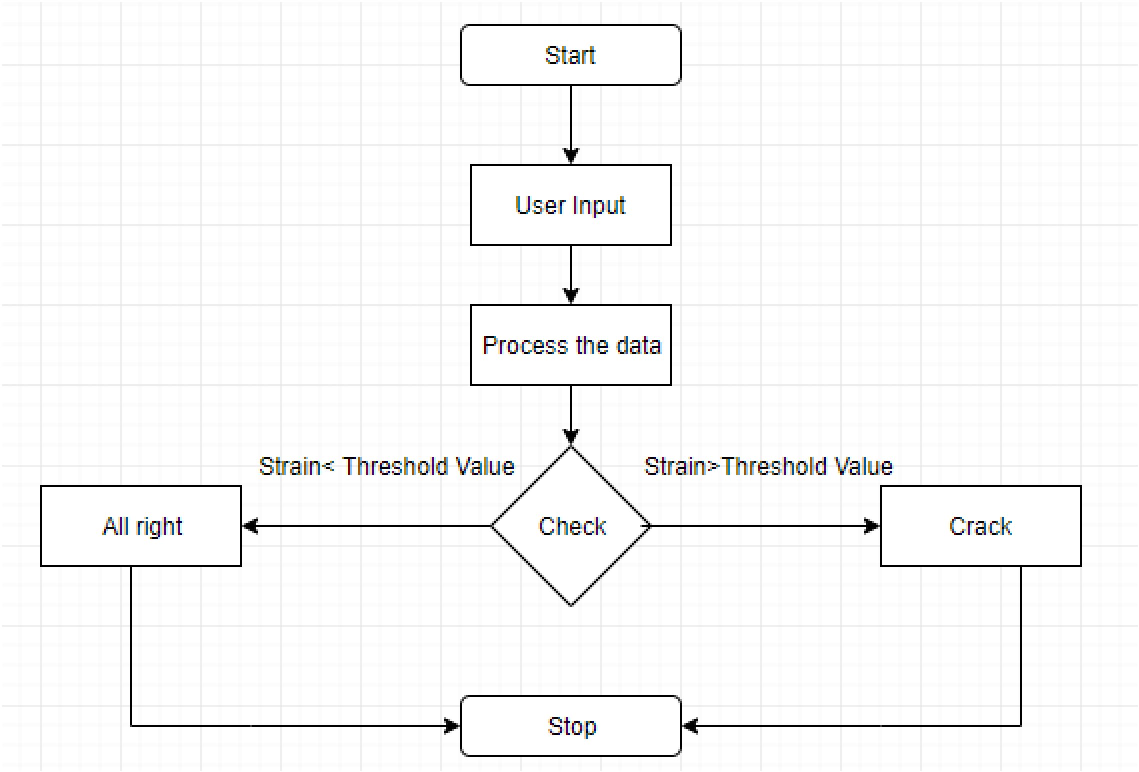
5.1 System Application Design

5.1.1 Method Pseudo code

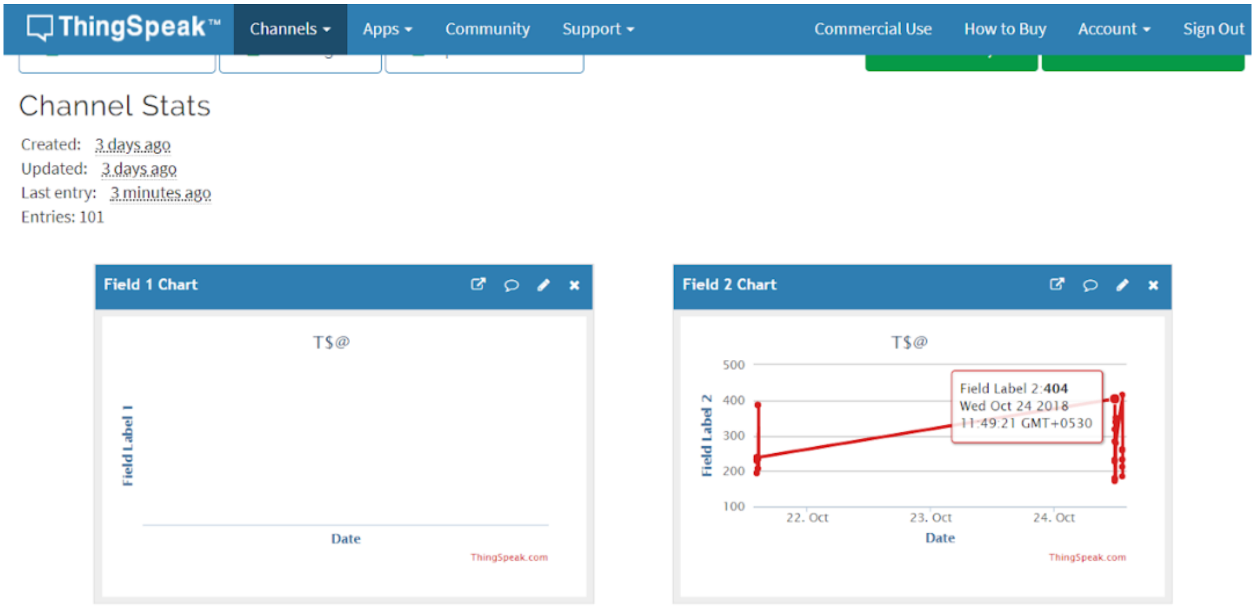
```
Step-1: Take input from parking slot
Step-2: Predict the value
Step-3: If value > 0
                Print("slot empty")
            Else
                Print("slot filled")
```

5.2 Input/output and Interface Design

5.2.1 State Transition/UML Diagram

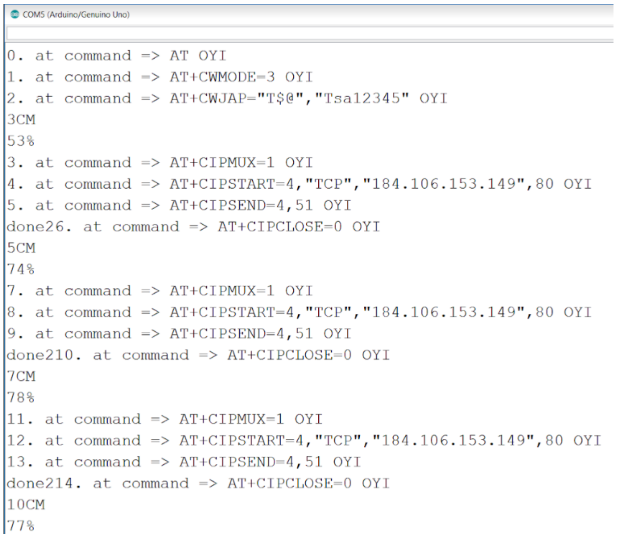


5.2.2 Samples Of Forms, Reports and Interface

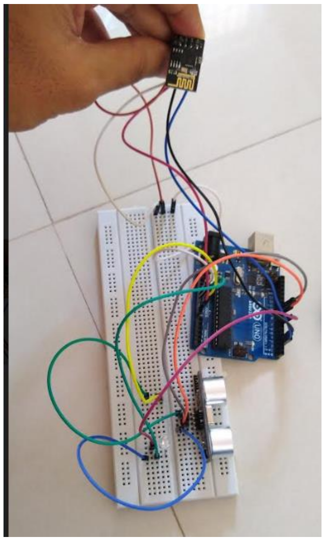


Channel Stats

Created: 3 days ago
Updated: 3 days ago
Last entry: 3 minutes ago
Entries: 101



```
0. at command => AT OYI
1. at command => AT+CWMODE=3 OYI
2. at command => AT+CWJAP="T$@","Tsa12345" OYI
3CM
53%
3. at command => AT+CIPMUX=1 OYI
4. at command => AT+CIPSTART=4,"TCP","184.106.153.149",80 OYI
5. at command => AT+CIPSEND=4,51 OYI
done26. at command => AT+CIPCLOSE=0 OYI
5CM
74%
7. at command => AT+CIPMUX=1 OYI
8. at command => AT+CIPSTART=4,"TCP","184.106.153.149",80 OYI
9. at command => AT+CIPSEND=4,51 OYI
done210. at command => AT+CIPCLOSE=0 OYI
7CM
78%
11. at command => AT+CIPMUX=1 OYI
12. at command => AT+CIPSTART=4,"TCP","184.106.153.149",80 OYI
13. at command => AT+CIPSEND=4,51 OYI
done214. at command => AT+CIPCLOSE=0 OYI
10CM
77%
```



6.0 IMPLEMENTATION PLANNING

6.1 Program/Modules Specification

There are two program modules in this project:-

- First module is for showing the customer whether parking slots are full/empty.
- Second module is for data uploading to thingspeak.

6.2 Coding Standards

The guidelines for good coding are listed below:-

- All the variables, functions, and methods should be assigned names that make the code more understandable to the reader. By using meaningful names, the code can be self-explanatory, thus, minimizing the effort of writing comments for variables.
- The names of functions should be meaningful and should describe the purpose of the function with clarity and briefness. Like variables, the names should be self-explanatory so that no additional description about the task of that function is required.
- Comments should be used with important segments of code and code segments that are difficult to understand.
- Comments should be separated from the code to enhance readability of the software code.

7.0 TESTING

7.1 Testing Plan

For the Testing of this project there are main two modules. To test both modules we will use both Black-box and white-box testing. It will reveal all the errors in the program.

7.2 Testing Strategy

We will test this project by both Black-box and White-box testing. For this project we will use dummy test data.

7.3Testing Methods

Black-box testing: Black box testing is the Software testing method which is used to test the software without knowing the internal structure of code or program. The main purpose of the Black Box is to check whether the software is working as per expected in requirement document & whether it is meeting the user expectations or not. Types of **Black Box Testing Techniques:** Following black box testing techniques are used for testing the software application.

- Boundary Value Analysis (BVA)
- Equivalence Class Partitioning
- Decision Table based testing
- Cause-Effect Graphing Technique
- Error Guessing

White-box testing: White Box Testing is the testing of a software solution's internal coding and infrastructure. It focuses primarily on strengthening security, the flow of inputs and outputs through the application, and improving design and usability. White box testing is also known as Clear Box testing, Open Box testing, Structural testing, Transparent Box testing, Code-Based testing, and Glass Box testing.

7.4 Test Suites Design

For test suit we will take a car and check that whether the led displays the result. Then will check that whether the data is being updated in the thingspeak.

7.5 Test Cases

Black Box Testing:-

Test no	Function Name	Test Case (condition)	Expected Results	Actual Result	Pass/ Fail
1	Display	Display the result on the LED	Should display the result on LED	Displays the result on LED	Pass
2	Display	Upload data to thingspeak	Should display the data upload on thingspeak	Displays the data upload on thingspeak	Pass

White Box Testing Example:-

Test no	Condition	Input/Test Data	Expected Result
1	If distance less than 40 cm	LED should glows	LED glows
2	If distance more than 40 cm	LED shouldn’t glow	LED doesn’t glow

8.0 CONCLUSION AND DISCUSSION

8.1 Self Analysis of Project Viabilities

- This project can be applicable to the real life project
- This project can make all parking system automatic

8.2 Problem Encountered and Possible Solutions

- We need efficient amount of knowledge to handle the data upload on thingspeak and display the led which helps the customer to know whether parking slot is vacant/fill.

8.3 Summary

Car parking system using Arduino is actually used to make the automated parking system which helps customer to know whether parking slots are empty/filled so that they can easily park their vehicle which saves their time. We have used a ESP8266 wifi module to pass data to cloud with help of thingspeak. Which helps to update each time car enters/leave that particular parking system.

9.0 LIMITATIONS AND FUTURE ENHANCEMENT

Limitations of the project:-

- It is working on smaller data.
- This is basic model of actual system.
- Cost increases as we increase the parking slots in the parking system.

Future enhancement of the project:-

- We can apply this project to any parking slots .
- We are trying to analyze for the larger amount of data.

10.0 REFERENCE

- *https://www.youtube.com/watch?v=u8e8zw7_Oi0*
- *https://www.robotsthenextspeciesonearth.com/p/blog-page_38.html*
- *https://create.arduino.cc/projecthub/eka-labs/smart-car-parking-system-8a5530*
- *https://www.youtube.com/watch?v=u8e8zw7_Oi0*