CHAPTER 1

INTRODUCTION

The car license plate recognition identification is an important application in the field of Intelligent Transport System (ITS) and Electronic toll collection (ETC). The objective is to extract and recognize vehicle registration numbers from car images, process the image data finally utilize for access record and prepare electronic bill. Electronic toll collection (ETC) or Electronic Car parking payment is one of the major research topics in intelligent transportation system (ITS). ETC is an implementation of a road pricing concept in order to create benefits such as increasing the capacity of toll stations, reducing a toll paying time, enhancing the convenience and safety of travelers, and minimizing air pollution and fuel consumption. It enables freeway to parking lot, toll plaza, and bridge, tunnel, and turnpike operators to save on staffing costs while reducing delay for travelers and improve overall traffic performance and parking system. Moreover, monitoring the vehicle traffic and the management of parking areas are the most labor-intensive job. Therefore the research on systematic full automatic parking system is proposed. It differs from conventional parking system, no magnetic card is used to record the entry and exit time of. Also, it is designed in such a way that it has the ability of giving out the information regarding parking free spaces to users before entering the parking spaces.

In the recent years, License Plate Recognition (LPR) are having a wide impact in people's life as their scope is to improve transportation safety and mobility and to enhance productivity through the use of advanced technologies. This system is useful in many fields and places such as parking lots, private and public entrances, border control, theft and vandalism control. For the past two decades, there have been various studies for number plate recognition in images. Early day's number plate recognition systems employed detection methods based on techniques such as edge analysis, color analysis, and the Adaboost training Ernst introduced a face detection method based on local structure patterns computed by the Modified Census Transform (MCT).

SMART PARKING SYSTEM

Using intelligent sensors can help parking lots to realize in time parking space inventory control. Nowadays, the parking problem is more than a nuisance. Parking managers and relevant researchers tried hard to solve that problem. Induction loop, camera and magnetic sensors have been installed in intelligent parking systems, thus cars can be counted easily in the entrance and exit and the occupation of every single parking space can be detected.

Meanwhile, ultrasonic sensor fitted on the side of a car is used to detect parked cars and vacant spots. Thus, it can be more efficient to do parking survey in ground or underground parking lots.

Automated Parking System using RFID Technology Project is mainly based on the RFID technology. There has been a considerable amount of reduction in transaction costs and decrease in stock shortage with the use of Radio Frequency Identification (RFID) technology in automation. Most of the RFID networks include a wide range of automation technologies.

These technologies are RFID readers, RFID writers, RFID barcode scanners, and RFID controllers. In this study, a solution has been provided for the problems encountered in parking-lot management systems via RFID technology. RFID readers, RFID labels, computers, barriers and software are used as for the main components of the RFID technology.

The software has been handled for the management, controlling, and operation tasks for parking lots. Check-ins and check-outs of the parking-lots will be under control with RFID readers, labels and barriers. Check-ins and check-outs will be handled in a fast manner without having to stop the cars so that traffic jam problem will be avoided during these processes

Since there won't be any waiting during check-ins and check-outs the formation of emission gas as a result of such waiting will be avoided. Therefore, by this project we develop a parking system for an organization to have automated parking system for making best use of space, decreasing the man power and providing authentication for the vehicles from avoiding the theft.

CHAPTER 2
LITERATURE SURVEY ON NATIONAL AND INTERNATIONAL SCENARIO

In legacy parking management systems, only the administrator has information about the parking spaces occupied by vehicles. Since the existing parking system cannot use the active information exchanging, it did not provide useful parking information for drivers. To solve this problem, smart sensors and the middleware for handing them are needed.

The vehicle parking location service has been proposed on the using of RFID devices. In this service, the drivers have to receive an RFID tag on the entrance of parking lot. The tag provides the vehicle location service for drivers through the RFID reader of parking space. However, this approach is inconvenient because the driver must receive the RFID tag in the entrance. In addition, the cost for RFID tag is needed.

In the year 2006, 458,293 new registered vehicles were reported compared to the year 1999 where there were only 296,716 new registered vehicles, which makes it a rough estimate of 54.5% increase in a span of 7 years (Malaysian Ministry of Transportation, 2007). Referring to the aforesaid statistics provided by the Malaysian Ministry of Transportation, the current transportation infrastructure and car park facilities are deemed insufficient in sustaining the influx of vehicles on the road.

Therefore, problems such as traffic congestion and insufficient parking space inevitably crops up. In Asia, the situation are made worse by the fact that the roads are significantly narrower compared to the West (Inaba et al., 2001). Various measures have been taken in the attempt to overcome the traffic problems. Although, the problem can be addressed via many methods, the paper focuses on the car park management system introduced, which is the smart par.

2.1 General features of Car Parking System

Enable the driver to collect ticket upon entrance: car Parking system should be able to allow the driver to get his ticket after he press the button of the gate barrier.

The system should record the entire cars that pass through the entrance.

The system should allow the gate to open whenever a driver has press the button and take his ticket.

Allow the drivers to make payment: if it's of commercial use, the system should enable the drivers to make payment of their charges before exiting.

Allow the driver to exit: if the driver has paid his charges and require exiting, the system should open the gate to allow him exiting.

In [3], the authors presents the generic concept of using cloud-based intelligent car parking services in smart cities as an important application of the Internet of Things (IoT) paradigm. This type of services will become an integral part of a generic IoT operational platform for smart cities due to its pure business-oriented features. A high-level view of the proposed middleware is outlined and the corresponding operational platform is illustrated. To demonstrate the provision of car parking services, based on the proposed middleware, a cloud-based intelligent car parking system for use within a university campus is described along with details of its design, implementation, and operation

In [4], the authors present the results of a survey on the needs of drivers from parking infrastructures from a smart services perspective. As smart parking systems are becoming a necessity in today's urban areas, we discuss the latest trends in parking availability monitoring, parking reservation and dynamic pricing schemes. We also examine how these schemes can be integrated forming technologically advanced parking infrastructures whose aim is to benefit both the drivers and the parking operators alike.

As in [5], authors have been working on a new application – smart parking – to test their IoT Tools. Using Nwave's car parking sensors (which use the Weightless-N wireless protocol that Nominet contributed to), and their IoT Tools they have been able to quickly build a system that allows them to monitor car park usage at Nominet's Oxford HQ in real-time.

In [6] the authors adopt Elliptic Curve Cryptography (ECC) as an attractive alternative to conventional public key cryptography, such as RSA. ECC is an ideal candidate for

implementation on constrained devices where the major computational resources, i.e., speed, memory are limited and low-power wireless communication protocols are employed.

In [7], parking lots have largely been ignored when it comes to technological innovations and ideas. However with the number of cars on the street increasing, and along with it the time taken to park, the parking problem will now have to be addressed effectively.

2.2 Old Car Parking System

Old car parking system was developed using 8051 microcontroller. It has the sections: Display section, Keyboard, indicator & Beeper section, Lift & motor section, Sensor section, LCD section. Program is written using 8051 microcontroller. Two 8255 IC's are connected to 8051. All circuits are interfaced with 8255. The display section displays the floor number along with the number of cars which has been already parked in that particular floor. So whenever a car is ready to either come down or go up, the program either decrements the count or increments the count automatically according to the going up or coming down of a car. Display section is done by interfacing with 8255(PPI) of 8051.Here 3 ports of 8255 are connected to three 7-segment display. Block diagram of this section is shown. 3

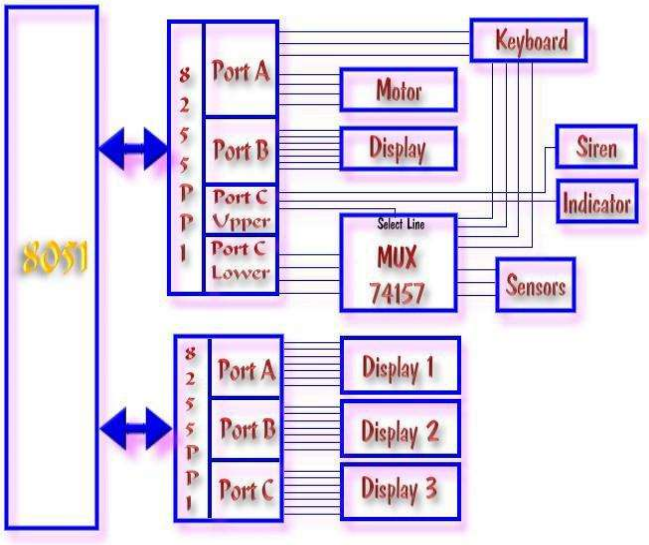

Figure 2.2.1 - Car Parking System using 8051 Microcontroller

If the entered password is wrong, beeper starts beeping signifying the incorrectness of the password entered. The indicator section contains 2 LED's, RED & GREEN which are present in all the floors. RED LED signifies that the lift is presently busy and shall not entertain any car to enter but if GREEN LED glows, it suggests that the lift is ready and the car can enter the particular floor. Beeper and LED's are connected to port C upper of 8255. One more advantage of beeper is that; when a person tries to enter the lift irrespective of finding the display section to be FFF (means the floors are already filled), program sends a signal to Beeper section and it starts beeping indicating that he is not supposed to enter the lift since all the floors are already filled. In lift and motor section, there is a light beam and LDR to know whether a car has entered the lift or not. When the GREEN LED of indicator section glows, that means the lift is ready for the car to enter. When the car enters the lift, the light beam falls on LDR present in the lift gets cut and it gives a signal that a car has entered the lift. Then program decides which floor lift has to go and gives a signal to motor section. The motor section is a mechanical part of the model which is used for taking the lift up/down. When the lift has to go up, program gives the signal and the motor rotates clockwise and if it has to go down, it rotates anticlockwise.

First 4 pins port A is connected to motor. Power transistors must be connected to drive the motor. Circuit diagram of this section is shown bellow. Sensor section contains LDR's .These LDR's are connected to each floor to give information if any car has to come down. When a person needs to come down from a particular floor to ground floor, he is expected to focus the headlight the car onto the LDR placed in that floor falls on LDR its resistance decreases. Hence IC 555 triggers and gives a signal. Program identifies that signal and gives a signal to motor section.

CHAPTER 3

EXISTING SYSTEM

In legacy parking management systems, only the administrator has information about the parking spaces occupied by vehicles. Since the existing parking system cannot use the active information exchanging, it did not provide useful parking information for drivers. To solve this problem, smart sensors and the middleware for handing them are needed.

The vehicle parking location service has been proposed on the using of RFID devices. In this service, the drivers have to receive an RFID tag on the entrance of parking lot. The tag provides the vehicle location service for drivers through the RFID reader of parking space. However, this approach is inconvenient because the driver must receive the RFID tag in the entrance. In addition, the cost for RFID tag is needed.

In the year 2006, 458,293 new registered vehicles were reported compared to the year 1999 where there were only 296,716 new registered vehicles, which makes it a rough estimate of 54.5% increase in a span of 7 years (Malaysian Ministry of Transportation, 2007). Referring to the aforesaid statistics provided by the Malaysian Ministry of Transportation, the current transportation infrastructure and car park facilities are deemed insufficient in sustaining the influx of vehicles on the road.

Enable the driver to collect ticket upon entrance: car Parking system should be able to allow the driver to get his ticket after he press the button of the gate barrier.

The system should record the entire cars that pass through the entrance.

The system should allow the gate to open whenever a driver has press the button and take his ticket.

Allow the drivers to make payment: if it's of commercial use, the system should enable the drivers to make payment of their charges before exiting.

Allow the driver to exit: if the driver has paid his charges and require exiting, the system should open the gate to allow him exiting.

CHAPTER 4

SMART PARKING SYSTEM

COMPONENTS REQUIREMENT FOR THE PROJECT

Name of equipment	Configuration
Sensor Controller	• Atmega 2560, Dual Serial,Microcontroller ATmega2560 • Operating Voltage 5V • Digital I/O Pins 54 (of which 15 provide PWM output) • Flash Memory 256 KB of which 8 KB used by bootloader • Clock Speed 16 MHz
Raspberry Pi V3.0	• Quad Core 1.2GHz Broadcom BCM2837 64bit CPU, 1GB RAM • BCM43143 WiFi and Bluetooth Low Energy (BLE) on board - 40-pin Extended GPIO, 4x USB 2 ports
RFID Card Reader	Mifare RC522 Card Read Antenna Rf Module RFID Reader IC Card Proximity ModuleRfid writer.
RFID Tags	10pcs/Lot NFC Sticker 13.56MHz ISO 14443A, NTag 213 RFID Tag for all NFC enabled phones.
Servo Motor	10 pcs/lot TowerPro Servo Motor 9g Torque 1.8Kg for RC Aero Modelling parts
Light Dependent Resistor (LDR)	LDR 5mm photo resistor
GSM Basic Module	SIM800C Quad-band wireless GPRS GSM Module for Raspberri Pi.
LCD (Liquid Crystal Display) Module	Nextion 2.4'' TFT Resistive Touch Screen UART HMI Smart Raspberry Pi LCD module display for Arduino TFT English

Table 4.1 – Requirements

CHAPTER 5

PROPOSED SYSTEM

5.1 Entry Side:

When the user enters the parking lot, the welcoming message is displayed. Then the RFID of the user's car is scanned and promted user to select his desired location at the time of his exit. The slot number is alloted according the the size of the car and the entry time is recorded. If the slot according to the car's size is not available, a random slot is alloted and asked with the user to confirm the slot. These details are then updated in the database and are sent to user's registered mobile number which is available at the time of his car's registration on successful confirmation from the user. On failure the system prompts with the appropriate failure message and terminate the session and if required the user can try after some-time. Guidance to the slot is also provided by glowing LED at the slot.

5.2 Exit Side:

At the time of user's exit, the user's RFID is scanned and interval between the time of entry and the time of exit is calculated. The system then checks for the fine and prompts the user to pay the fine and confirm's the location where the user had provided at the time of entry. Here a choice is provided whether to confirm or to redefine the desired location. The exit gate opens immediately if the choice of confirmation is success else the exit gate number is sent to user's mobile and the above checks are made once again except with the rentry of the location. When the user enters to that exit gate and the corresponding exit gate opens thus making user much convinient and system much more automated and efficient.

CHAPTER 6

WORKING PRINCIPLE

The following figure is self-explanatory which describes the brief working principle of the proposed prototype.

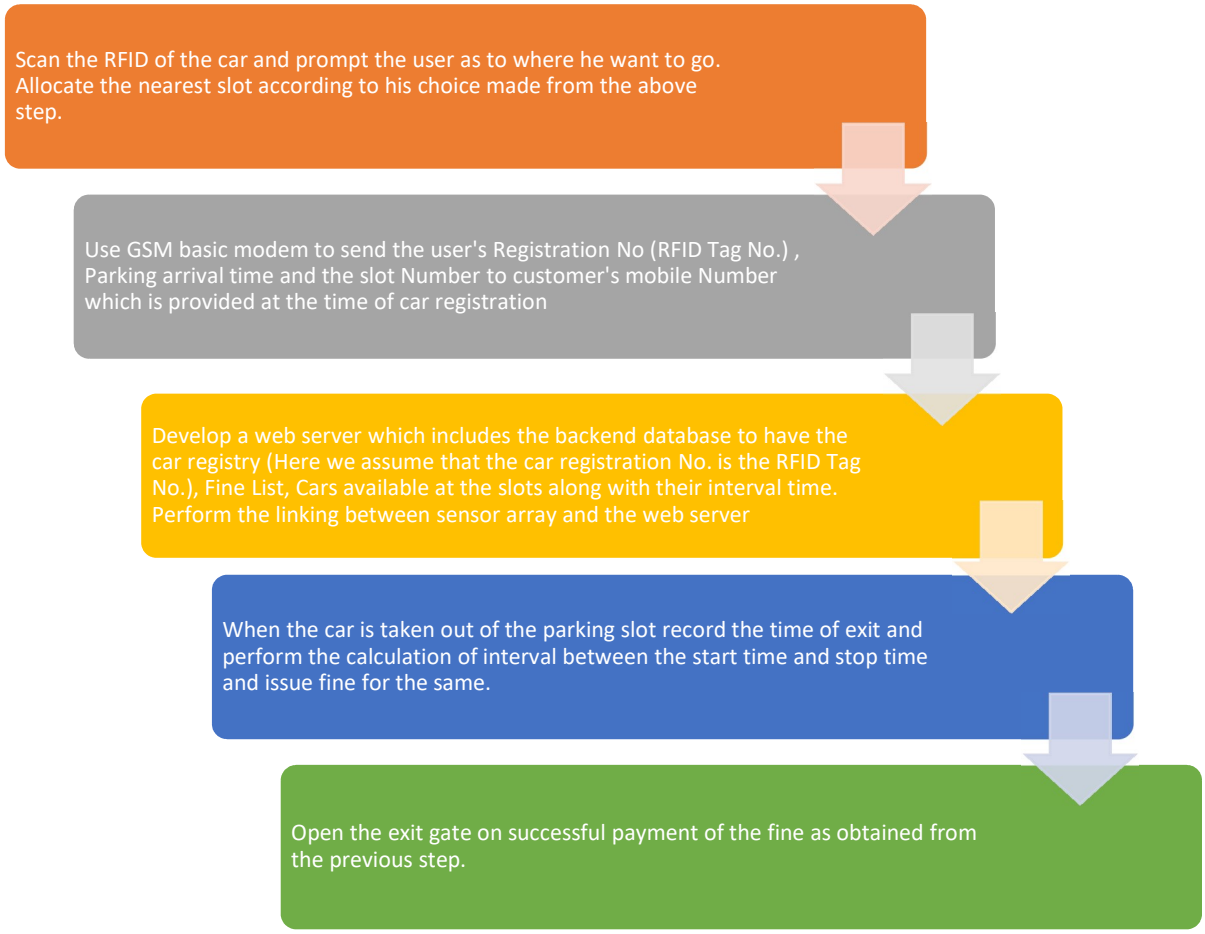

Figure-6.1: Block diagram of the working principle

In the above fig. it can be depicted that this follows the given scenario-

1. By deploying RFID technology and number plate technology a user can get access to nearest parking area as soon as possible without many efforts.

2. By using GSM module various details like RFID no parking time and slot no of parking area is sent to user on his mobile once he has parked his car.

3. A web server keeps the tract of various details like time interval of car parking system, RFID of related car, slot no which useful for billing and data record.

4. If car is going out of parking area calculate the parking fees according to fixed rates and if there is any fine then it has to be imposed on the user's bill.

5. Open the exit gate on successful payment of the fine as obtained from the previous step.

This serves the purpose for public parking lot system.

It provides check-ins and check-outs of the vehicles at the parking place. The person who is entering the parking area, the vehicles information is read by the RFID reader, the RFID then checks the database to see whether is a registered person or not and the barrier gets opened by allotting a slot to park his vehicle. Check-in time of the vehicle is noted. When he is checking out of the place the check time of the vehicle is noted and the vehicle is authenticated with his ID card. Suppose an unauthorized user is taking the vehicle, then the barrier doesn`t get opened. No vehicle can be parked without registering with the organization. Only the persons who had registered their vehicles with the organization can only park. This serves one of the purposes of the "Smart Parking System" wherein the above explanation is suited for the apartments.

6.1 Flow chart for entry

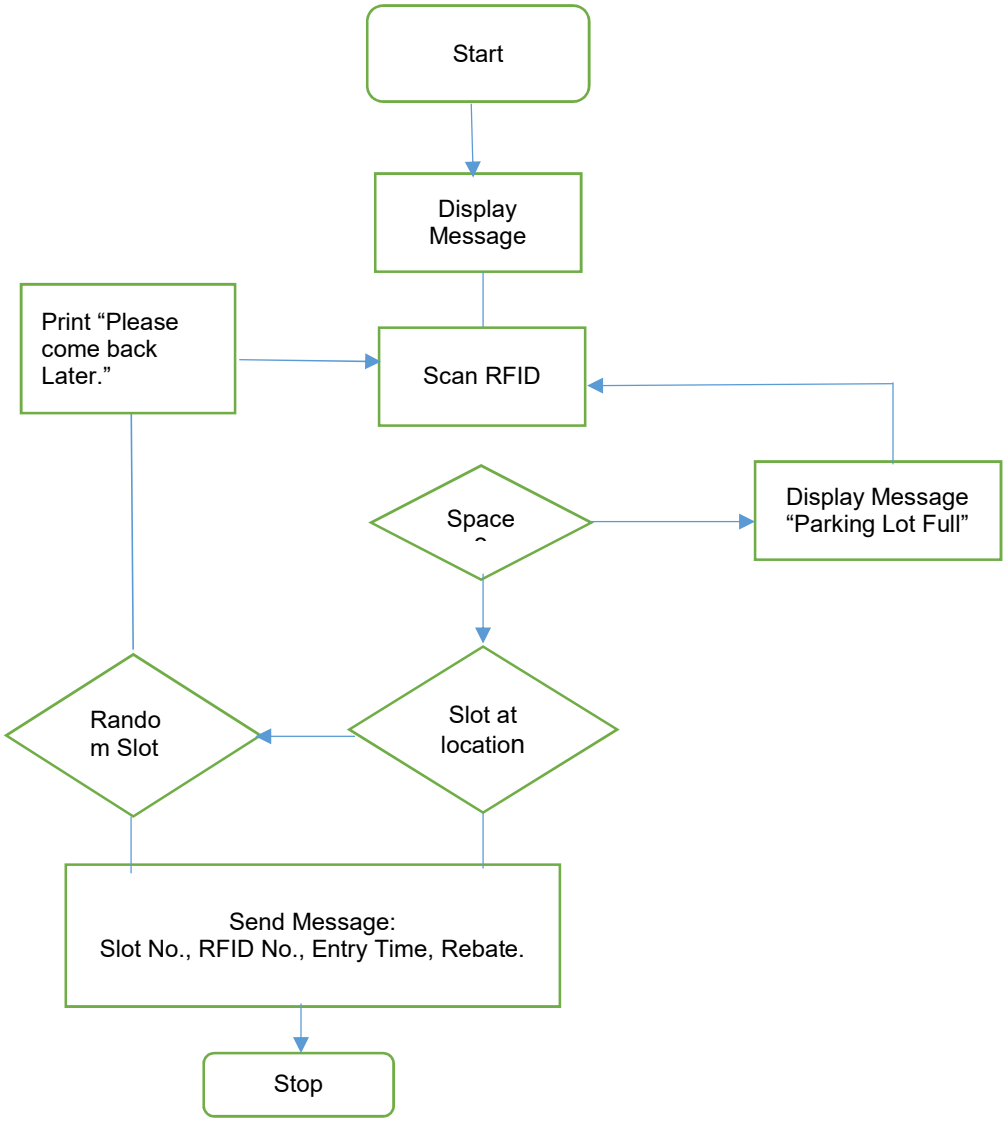

Figure-6.1.1:Flow chart of smart parking system

At the time of user's exit, the user's RFID is scanned and interval between the time of entry and the time of exit is calculated. The system then checks for the fine and prompts the user to pay the fine and confirm's the location where the user had provided at the time of entry. Here a choice is provided whether to confirm or to redefine the desired location. The exit gate opens immediately if the choice of confirmation is success else the exit gate number is sent to user's mobile and the above checks are made once again except with the rentry of the location. When the user enters to that exit gate and the corresponding exit gate opens thus making user much convenient and system much more automated and efficient.

6.2 Flowchart for Exit

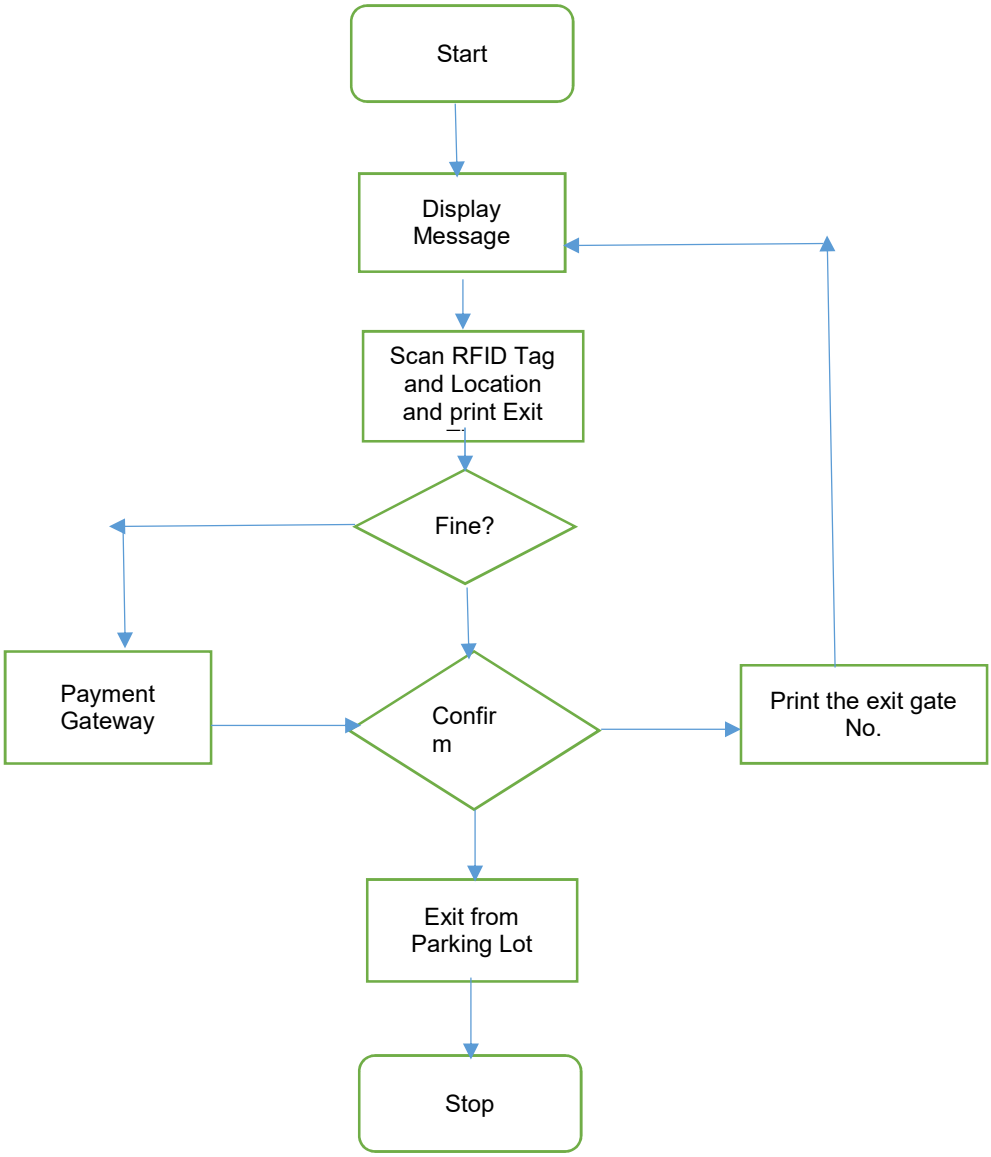

Figure-6.2.1 Exit System with automated payment system

At the time of user's exit, the user's RFID is scanned and interval between the time of entry and the time of exit is calculated. The system then checks for the fine and prompts the user to pay the fine and confirm's the location where the user had provided at the time of entry. Here a choice is provided whether to confirm or to redefine the desired location. The exit gate opens immediately if the choice of confirmation is success else the exit gate number is sent to user's mobile and the above checks are made once again except with the re-entry of the location. When the user enters to that exit gate and the corresponding exit gate opens thus making user much convenient and system much more automated and efficient.

CHAPTER 7

EXPECTED OUTCOME

1. Increase in accuracy
2. Better the time management.
3. Reduce parking and retrieval time.
4. Real time parking status updates.
5. Increases patron convenience
6. 24*7 Operation.
7. Enjoy wireless operation.
8. High security.
9. Traffic flow.
10. Pre-defined Access lists.
11. User friendly vehicle access.
12. Maintenance free, long-term use.
13. Easy to integrate / install.

CHAPTER 8
IMPLEMENTATION

- **Communication between multiple arduinos is successfully established.**

  We have used and programmed the popular I2C communication bus to receive and send data between the microcontrollers.

  I²C (**Inter-Integrated Circuit**): I-squared-C, is a multi-master, multi-slave, single-ended, serial bus invented by Philips Semi-conductor . It is typically used for attaching lower-speed peripheral ICs to processors and microcontrollers in short-distance, intra-board communication. Alternatively I²C is spelled **I2C** (pronounced I-two-C) or **IIC**(pronounced I-I-C). I²C uses only two bidirectional open-drain lines, Serial Data Line (SDA) and Serial Clock Line (SCL), pulled up with resistors. Typical voltages used are +5 V or +3.3 V although systems with other voltages are permitted. Common I²C bus speeds are the 100 kbit/s *standard mode* and the 10 kbit/s *low-speed mode*. Recent revisions of I²C can host more nodes and run at faster speeds (400 kbit/s *Fast mode*, 1 Mbit/s *Fast mode plus* and 3.4 Mbit/s *High Speed mode*). These speeds are more widely used on embedded systems than on PCs.

- **Suitable Backend system to receive the sensor's data is successfully created and tested.**

  We are using the popular PHP 5.6 and Maria-db as our preferred database for the same.

  Pre-processor hypertext is a server-side scripting language designed primarily for web development also used as a general-purpose programming language. Originally created by Rasmus Lerdorf , the PHP reference implementation is now produced by The PHP Development Team. PHP originally stood for *Personal Home Page*, but it now stands for the recursive acronym *PHP: Hypertext Preprocessor*. PHP code may be embedded into HTML or HTML5 code, or it can be used in combination with various web template systems.

- **Suitable frontend for the same is designed.**

  We are using the latest web view technology from Bootstrap to make it device independent compatibility. The webpage is highly responsive. However, we are yet to program the backend system with the same.

  Bootstrap is a front end free open source framework for, designing websites and web applications. It contains HTML and CSS-based design templates for typography, forms, buttons, navigation and other interface components, as well as optional JavaScript extensions. The style sheets are generally compiled into a bundle and included in web pages, but individual components can be included or removed. Bootstrap provides a number of configuration variables that control things such as color and padding of various components.

- **The raspberry pi which acts as gateway is successfully programmed to receive and store the sensor's data from the microcontroller.**

  We are using the latest version of raspberry pi with built Bluetooth and wifi technology to make the system completely wireless and accessibility of portal anywhere and anytime.

  The **Raspberry Pi** is a series of small single-board computers developed in the United Kingdom by the Raspberry Pi Foundation to promote the teaching of basic computer science in schools and in developing countries. The original model became far more popular than anticipated, selling outside of its target market for uses such as robotics. Peripherals (including keyboards, mice and cases) are not included with the Raspberry Pi.

8.1 Detailed Library Implementation

a) TCPDF: It is a free and open source software for generating PDF documents. TCPDF is the only PHP-based library. TCPDF is a PHP class for generating PDF files on-the-fly without requiring external extensions.
TCPDF library also includes classes to extract data from existing PDF documents and classes to generate 1D and 2D barcodes in various formats.

TCPDF is also one of the most used PHP libraries, Since it has been already included most of Applications: Drupal, Moodle, phpMyAdmin, TCExam, Xoops, Elxis CMS, Impress CMS, Jelix and many others.

Salient Features of TCPDF:

*There is no requirement for external libraries for performing the basic functions.
* All the formats like standard page formats, custom page formats etc are considered.
* Sub setting of fonts.
* Several methods to publish some JavaScript and Forms.
* Several images, graphic and transformation methods are involved.
* It supports JPEG, PNG images natively and all images are supported via ImagMagick.
* Automatic page header and footer management.
* Secure document encryption up to 256 bit and digital signature certifications;
* Also supports transactions UNDO commands;
* PDF annotations, including links, text and file attachments;[18]

b) Servo.h: It allows an Arduino board to control servo motors. Servos have integrated gears and a shaft that can be accurately controlled. Standard servos allow the shaft to be positioned at various angles, usually between 0 and 180 degrees.

The Servo.h library supports up to 12 motors on most Arduino boards and 48 on the Arduino Mega. On boards other than the Mega, use of the library disables AnalogWrite() (PWM) functionality on pins 9 and 10, whether or not there is a Servo on those pins. It supports functions like:

- attach()
- write()
- writeMicroseconds()
- read()
- attached()
- attach()

Circuit: Servo motors have three wires: power, ground, and signal. The power wire is typically red, and should be connected to the 5V pin on the Arduino board. The ground wire is either black or brown and should be connected to a ground pin on the Arduino board. The signal pin is typically yellow, orange or white and should be connected to a digital pin on the Arduino board. Note that servos draw considerable amount of power, so if you need to drive more than one or two, you'll probably need to power them from a separate supply (i.e. not the +5V pin on your Arduino). Be sure to connect the grounds of the Arduino and external power supply together. [17] [16]

c) Adafruit_GFX: It provides a common syntax and set of graphics functions for all of our LCD displays. This allows Arduino sketches to easily be adapted between display types, performance improvements and bug fixes will immediately apply across our complete offering of color displays.

The Adafruit_GFX library always works together with a second library Adafruit_ST7735 which is provided for each specific display type . The following libraries work as:

- RGBmatrixPanel (http://adafru.it/aHj), for our 16x32 (http://adafru.it/420) and 32x32 (http://adafru.it/607) RGB LED matrix panels.
- Adafruit_TFTLCD (http://adafru.it/aHk), for our 2.8" TFT LCD touchscreen breakout and TFT Touch Shield for Arduino(http://adafru.it/376).

- Adafruit_HX8340B (http://adafru.it/aHl), for our 2.2" TFT Display with microSD (http://adafru.it/797).
- Adafruit_ST7735 (http://adafru.it/aHm), for our 1.8" TFT Display with microSD (http://adafru.it/358).
- Adafruit_PCD8544 (http://adafru.it/aHn), for the Nokia 5110/3310 monochrome LCD (http://adafru.it/338).
- Adafruit-Graphic-VFD-Display-Library (http://adafru.it/aHo), for our 128x64 Graphic VFD (http://adafru.it/773).
- Adafruit-SSD1331-OLED-Driver-Library-for-Arduino (http://adafru.it/aHp) for the 0.96" 16- bit Color OLED w/microSD Holder (http://adafru.it/684).
- Adafruit_SSD1306 (http://adafru.it/aHq) for the Monochrome 128x64 (http://adafru.it/326) and 128x32 (http://adafru.it/661) OLEDs.

Using GFX Fonts in Arduino Sketches: After including the Adafruit_GFX and display-specific libraries, include the font files you plan to use in your sketch. For Example:

#include // Core graphics library

#include // Hardware-specific library

Each font takes up a bit of program space; larger fonts typically require more room. This is a finite resource (about 32K max on an Arduino Uno for font data and all of your sketch code), so choose carefully. Sometimes the code is too big and the code will refuse to compile (or in some edge cases, may compile but then won't upload to the board). If this happens, use fewer or smaller fonts, or use the standard built-in font. Inside these .h files are several data structures, including one main font structure which will usually have the same name as the font file (minus .h). To select a font for subsequent graphics operations, use the setFont() function, passing the address of this structure, such as: tft.setFont(&FreeMonoBoldOblique13pt8b); Following calls to tft.print() will now use this font. Most other attributes that previously worked with the built-in font i.e. color, size, etc. work similarly here. [17]

d) TFTLCD.h: The library works for the Adafruit_2.8" TFT display. The library works with the Adafruit_2.8" TFT Breakout with SD card as well as Adafruit_TFT_Touch_Shield. The displays use 8-bit parallel to communicate, 12 or 13 pins are required to interface (RST is optional). In the TFTLCD Library folder, we will need to edit TFTLCD.h. [17]

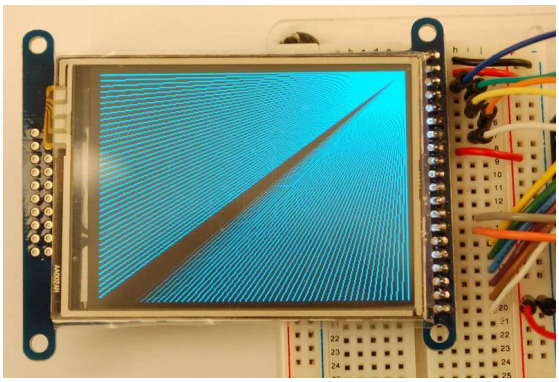

Figure 8.1.1 TFTLCD

e) SPI LIBRARY: It helps to communicate with SPI devices, with the Arduino as the master device. Serial Peripheral Interface is a synchronous serial data protocol used by microcontrollers for communicating with one or more peripheral devices quickly over short distances. It can be used for communication between two microcontrollers. [17]

SPI devices communicate in full duplex mode using a master-slave architecture with a single master. The master device originates the data frame for reading and writing. Sometimes SPI is called a *four-wire* serial bus, contrasting with three-wire, two-wire, and one-wire serial buses. The SPI can be described as a synchronous serial interface, but it is different from the Synchronous Serial Interface (SSI) protocol, which is also a four-wire synchronous serial communication protocol. But SSI Protocol employs differential signaling and provides only a single simplex communication channel. [17]

With an SPI connection there is one master device (usually a microcontroller) which controls the peripheral devices. Typically there are three lines common to all the devices:

- **MISO** - Master In Slave Out - Slave line for sending data to the master.
- **MOSI** - Master Out Slave In - Master line for sending data to the slaves.
- **SCK** - Serial Clock - The clock pulses synchronize data transmission.
- **SS** – Slave or Chip Select - the pin on each device that the master can use to enable and disable specific devices.

When a device's Slave Select pin is low, it communicates with the master. When it's high, it ignores the master. This allows us to have multiple SPI devices sharing the same MISO, MOSI, and CLK lines. For Example Multi RFID. [17]

Mode	Clock Polarity (CPOL)	Clock Phase (CPHA)	Output Edge	Data Capture
SPI_MODE0	0	0	Falling	Rising
SPI_MODE1	0	1	Rising	Falling
SPI_MODE2	1	0	Rising	Falling
SPI_MODE3	1	1	Falling	Rising

f) MFRC522: There are RFID modules that can read and write Mifare's tags. RFID-RC522 (MF-RC522), the microcontroller and card reader uses SPI for communication. The card reader and the tags communicate using a 13.56MHz electromagnetic field or radio frequency to be more precise. [17]

g) The Wire: This library allows you to communicate with I2C devices. On the Arduino boards with the R3 layout 1.0 pinout, SDA (the data line) and SCL (the clock line) are on the pin headers close to the AREF pin. The Arduino Due has two I2C / TWI interfaces SDA1 and SCL1 are near to the AREF pin and the additional one is on pins 20 and 21. The following reference table shows the pin numbers of TWI pins that are located on various Arduino boards.

Board	I2C / TWI pins
Uno, Ethernet	A4 (SDA), A5 (SCL)
Mega2560	20 (SDA), 21 (SCL)
Leonardo	2 (SDA), 3 (SCL)
Due	20 (SDA), 21 (SCL), SDA1, SCL1 [17]

As of Arduino 1.0, the library inherits from the Stream functions, making it consistent with other read/write libraries. Because of this, send() and receive() have been replaced with read() and write().

Hardware

There are three components involved:

1. **Micro Controller**:

- An Arduino is used as microcontroller.
- Also Raspberry pi, is also used which is a microcontroller as well as microprocessor with Broadcom chipset.

2. **Proximity Coupling Device (PCD)**:

- The PCD is the actual RFID **Reader** based on NXP MFRC522 Contactless Reader Integrated Circuit).

3. **Proximity Integrated Circuit Card (PICC)**:

- The PICC is the RFID **Card** or **Tag** using the ISO/IEC 14443A interface, for.

Protocols

1. The micro controller and the reader use SPI for communication.

- The protocol is described in the NXP datasheet.

2. The reader and the tags communicate using a 13.56 MHz electromagnetic field.

- The protocol is defined in ISO/IEC 14443-3:2011 Part 2 Type A.
    - The reader does not support ISO/IEC 14443-2 Type B.

Security: This library only supports crypto1-encrypted communication. Crypto1 cryptographic protocol has been known as broken from a few years, so it does NOT offer ANY security, it is virtually unencrypted communication. [17]

8.2 Algorithmic Analysis and pseudo-code of Smart Parking System

8.2.1 Embedded System Arduino programming

- *Request Data from the other Arduino*

```
void requestOtherArduino() {
   //If there Exist any Slot Allocation for the vehicles from other Arduino.
This means a car in dest2    //has arrived at dest1 or viceversa.
   Accept rfid and flag;
   Allocate a slot;
   print the details on lcd;
   send the slot via serial communication;
}
```

- *Allocate the Slots and Print the details on the LCD*

```
void printAlloc() {
   //Print the Scanned RFID and request the user for destination.
   if(destination!=other arduino) {
          Allocate a slot;
          print the details on lcd;
          send the slot via serial communication;
   }
   else {
          Establish I2C communication with the other arduino;
          Send it to the other Arduino;
   }
}
```

- *Main of Arduino*

```
   void setup() {
          Initialize all the sensors and the components;
           Initialize SPI communication;
          Initialize I2C communication;
          Initialize Serial communication;
   }
   void loop() {
          Scan RFID;
          Print the Entry Message on LCD;
   }
```

8.2.2 Server Sided PHP programming

- *Database Connection.*

```
<?php
      mysql_connect(‘Host_Ip',‘User_name',‘Password')
      or    die(mysql_error());
      mysql_select_db(‘Database_name') or die(mysql_error());
?>
```

- *Forms Creation.*

```
<form method=“post/get”  name=“ form _name” action=“next_action_form”>
     // Code for Creating textboxes/buttons etc.
</form>
```

- *Session Creation.*

```
<?php
      ob_start();         // object for session is created.
      session_start();      // session starts.
      require ‘databse_connection_file_name';
      if (isset($_SESSION['user']))  { header("Location:
session_filename"); exit;}
?>
```

- *Report Generation*

```
function fetch_data()
   {
      $output = '';
      while(Fetch all mysql table rows)
      {
           $output .= '<tr>
      Tabular rows
       </tr>';
     }
       return $output;
   }
```

8.2.3 Front End JavaScript Bootstrap validation

```
// Algorithm for form validation in jquery and js

function validateform2()
{
/* Function to be executed on successful validation */
}

}
$(function() {
  // Initialize form validation on the registration form.
  // It has the name attribute "registration"
  $("form[name="form name"]").validate({
    // Specify validation rules
    rules: {
    },
    // Specify validation error messages
    messages: {
    },
    // in the "action" attribute of the form when valid
    submitHandler: function(form) {
      validateform2();
    }
  });
});
```

8.2.4 Algorithmic Efficiency Analysis

- *Embedded Arduino Programming.*

Best case: $O(1)$
Average case: $O(1)$
Even though there exists many for loops, the number of comparisons are constant. The following reasons suffice the same.

(a) Number of slots in a parking system is assumed to be constant and not varying depending the size of the vehicle.
(b) Even though there are unlimited number of cars that are capable of entering the parking slots, there are finite slots. Hence there is a limit to the number of cars entering and exiting in a parking slot.
Mathematically,
$\exists\, f(x): x \to y$

Where x is the number of cars (A 1-D space of ∞ elements), y is the number of slots (A 1-D space of finite elements)
$\therefore$ There exists constant number of comparisons.

Worst case: $O(1)$
The same explanation can be observed.

As the efficiency is observed to be constant for all the three cases as explained, it can thus be concluded asymptotically that,

$$O(1) = \theta(1) = \Omega(1)$$

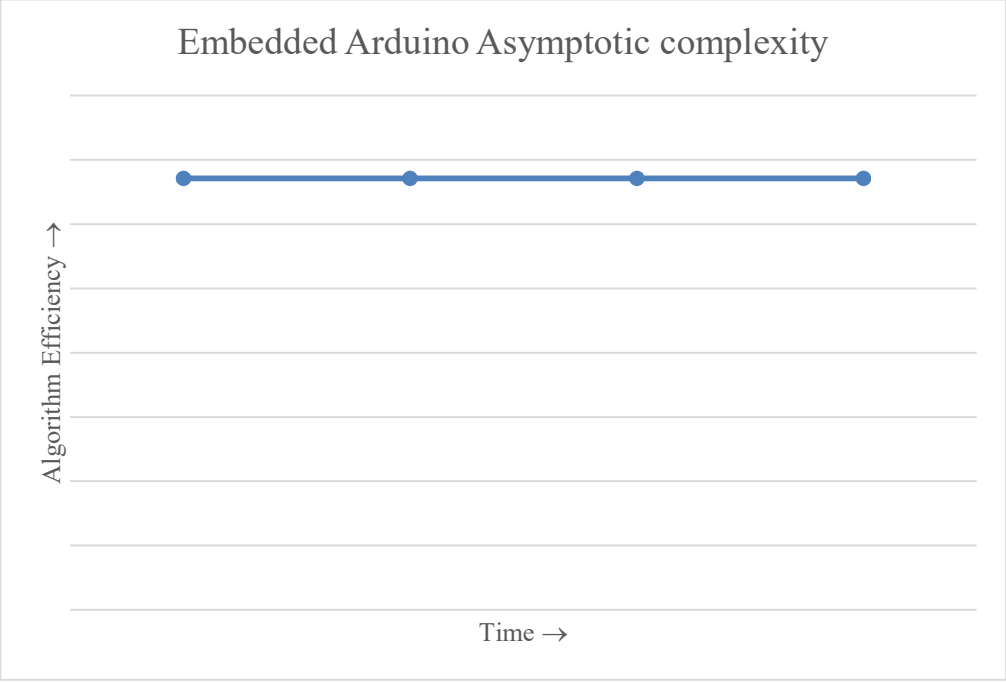

Fig 8.2.1 Embedded Arduino Asymptotic complexity

- *Server Side PHP Scripting and MySQL record Searching.*

  Best case: $O(1)$
  Average case: $O(\log n)$
  The following reasons will suffice the same:
  (1) As per the Oracle documentation MySQL uses indexing for searching. As far as this is concerned, it uses B+ tree for the same.
  (2) B+ tree takes $O(\log n)$ asymptotic complexity for searching the elements. Therefore the entire search takes the complexity as specified above.
  Worst case: $O(n)$
  It takes worst complexity only when user avails the following services:
  (1) Report Generation using PDF
  (2) Check the current availability of the parking slots in the mall.

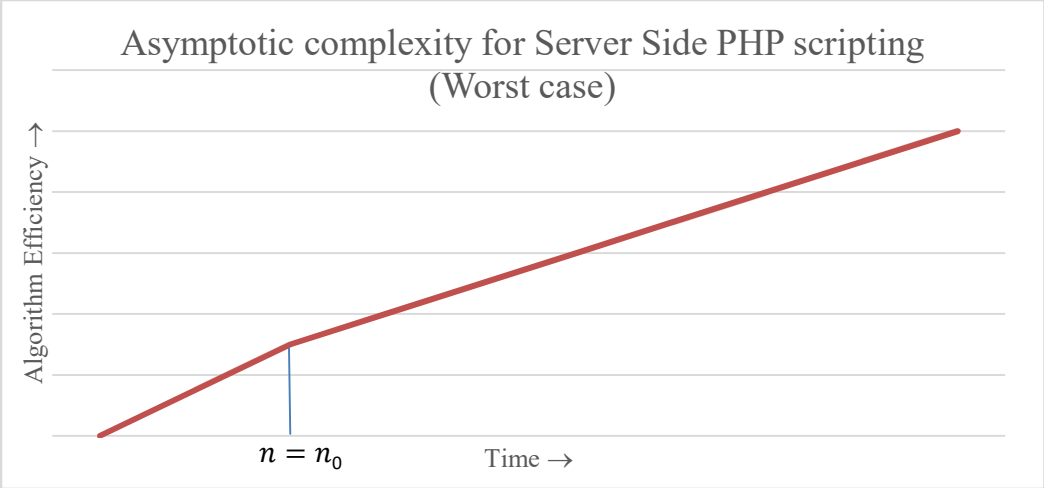

Fig 8.2.2 Asymptotic complexity for Server Side PHP scripting (Worst case)

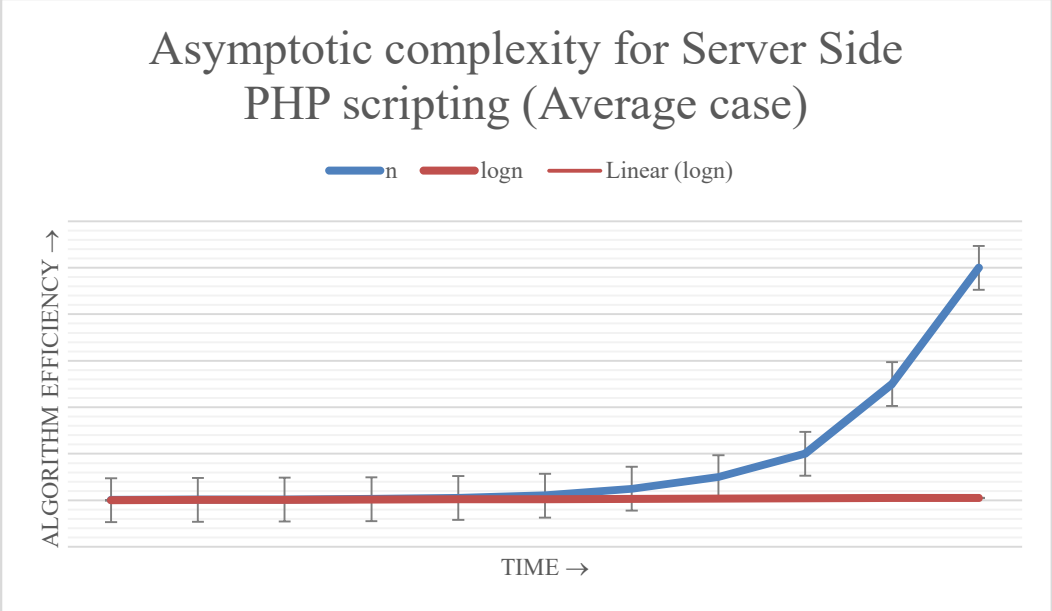

Fig 8.2.3 asymptotic complexity for Server Side PHP Scripting (Average Case) $O(\log n)$

CHAPTER 9

RESULTS

9.1 Hardware and Embedded Screenshots

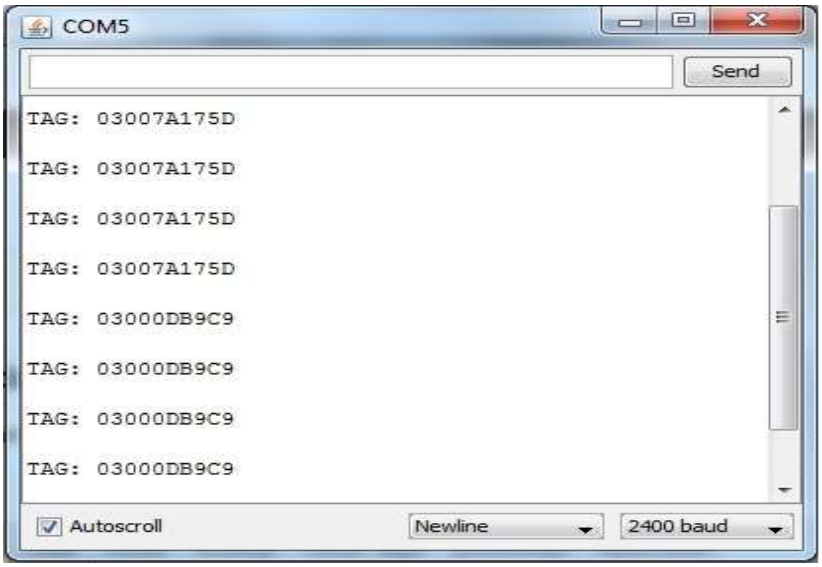

Figure-9.1.1 RFID Tag Scanning

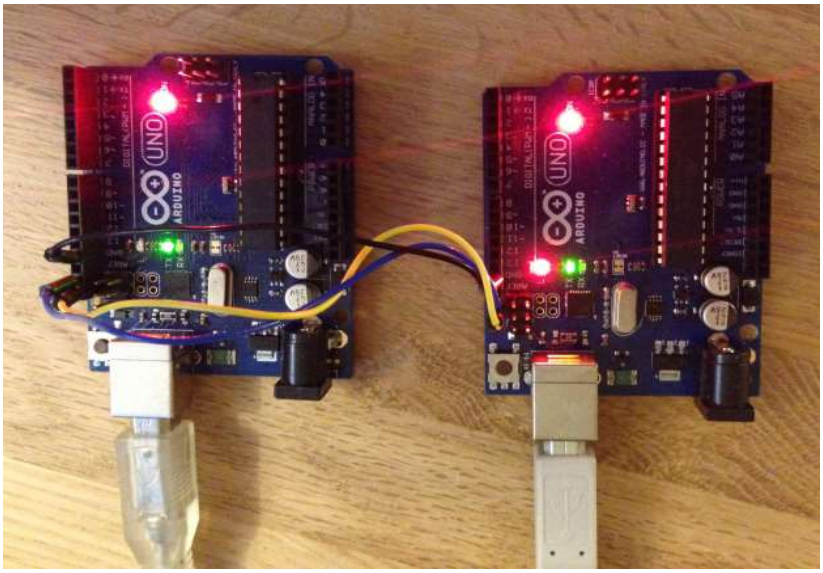

Fig 9.1.2: I2C communication between arduinos

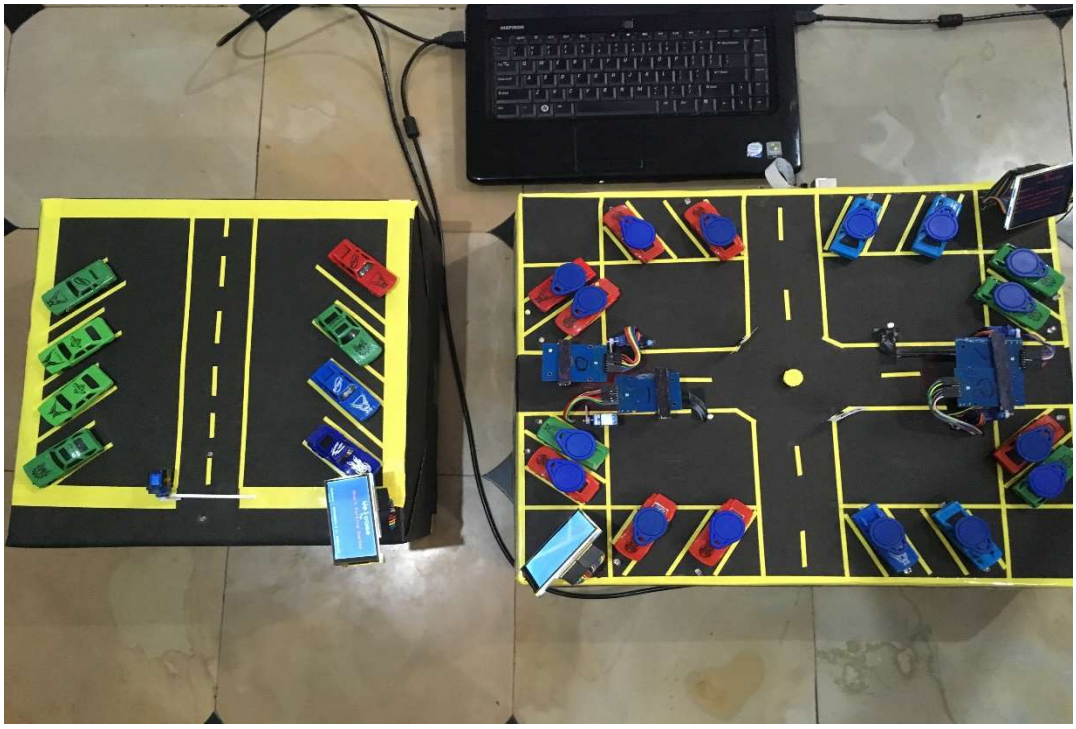

Fig 9.1.3 Overview of Smart Parking System

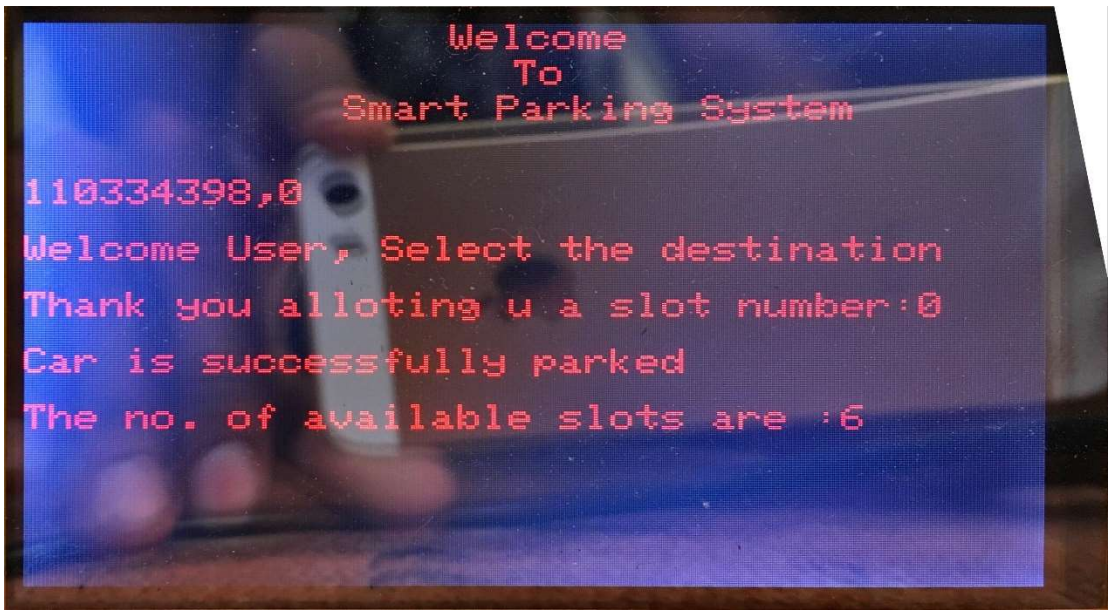

Fig 9.1.4 3.5'' TFT LCD Displaying slot allocation information

9.2 PHP Backend

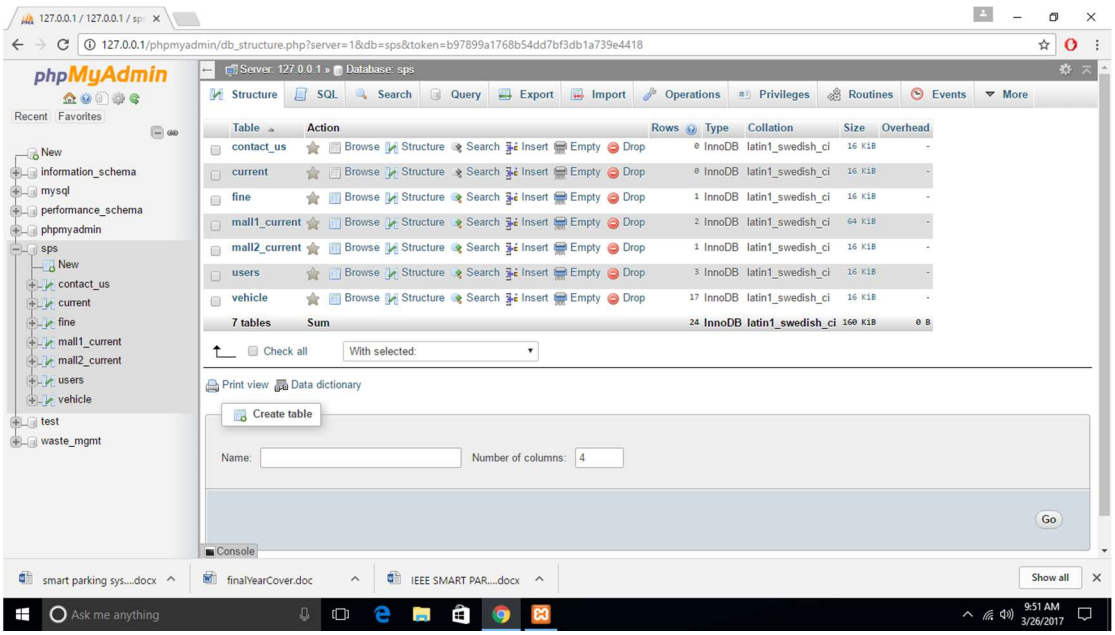

Fig 9.2.1: Database sps

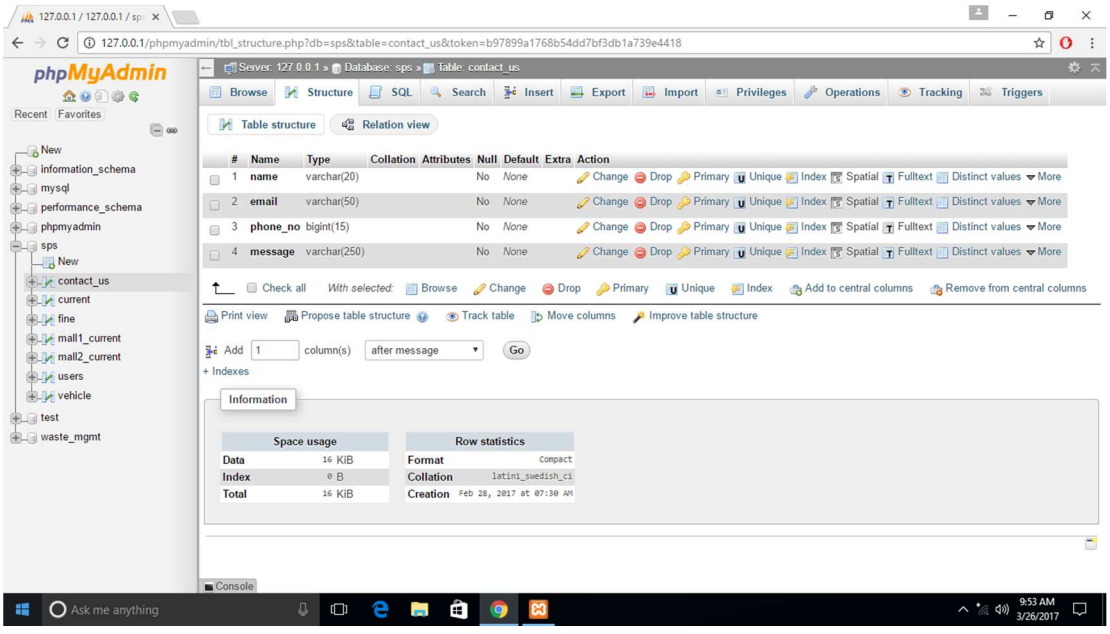

Fig 9.2.2 Database sps: Table: users structure

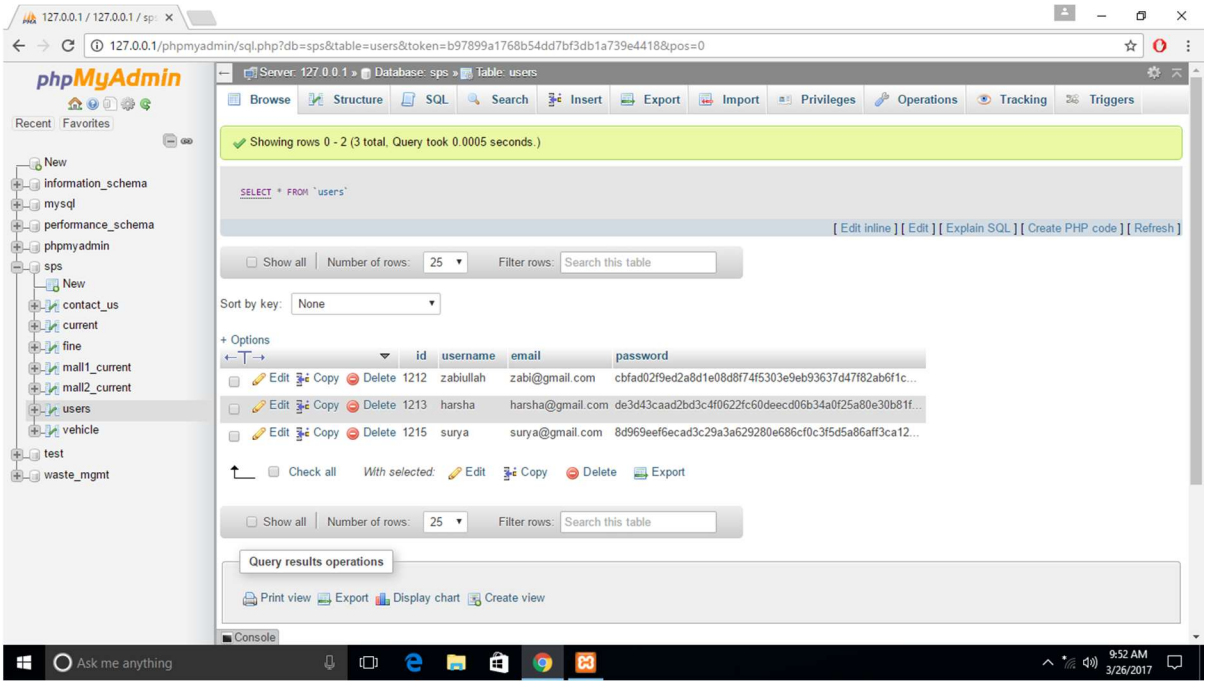

Fig 9.2.3 Database sps: Table: users tuples

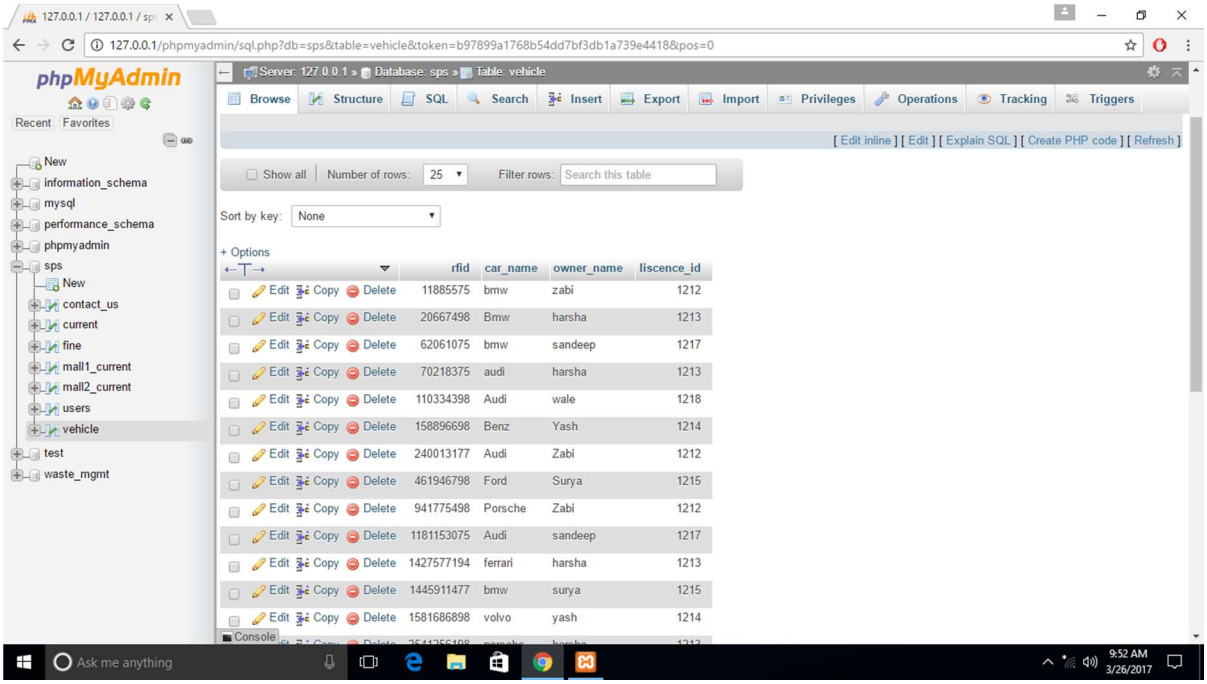

Fig 9.2.4 Database sps: Table: vehicle tuples

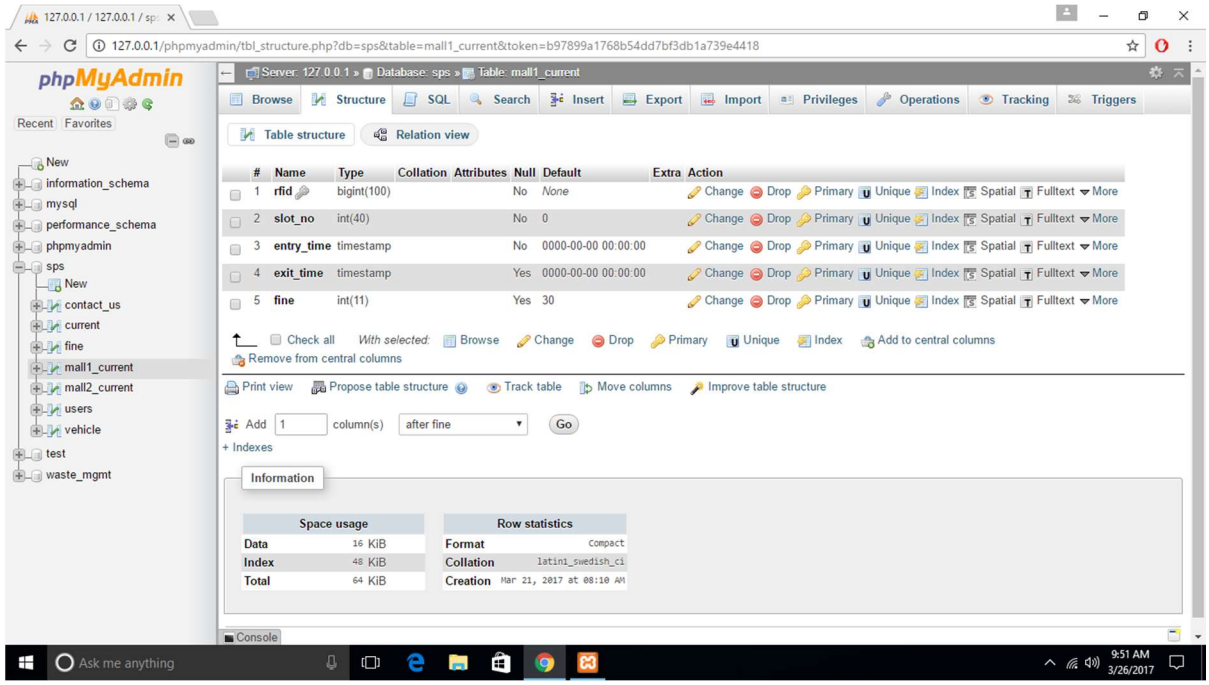

Fig 9.2.5: Database sps: Table: mall1_current Structure

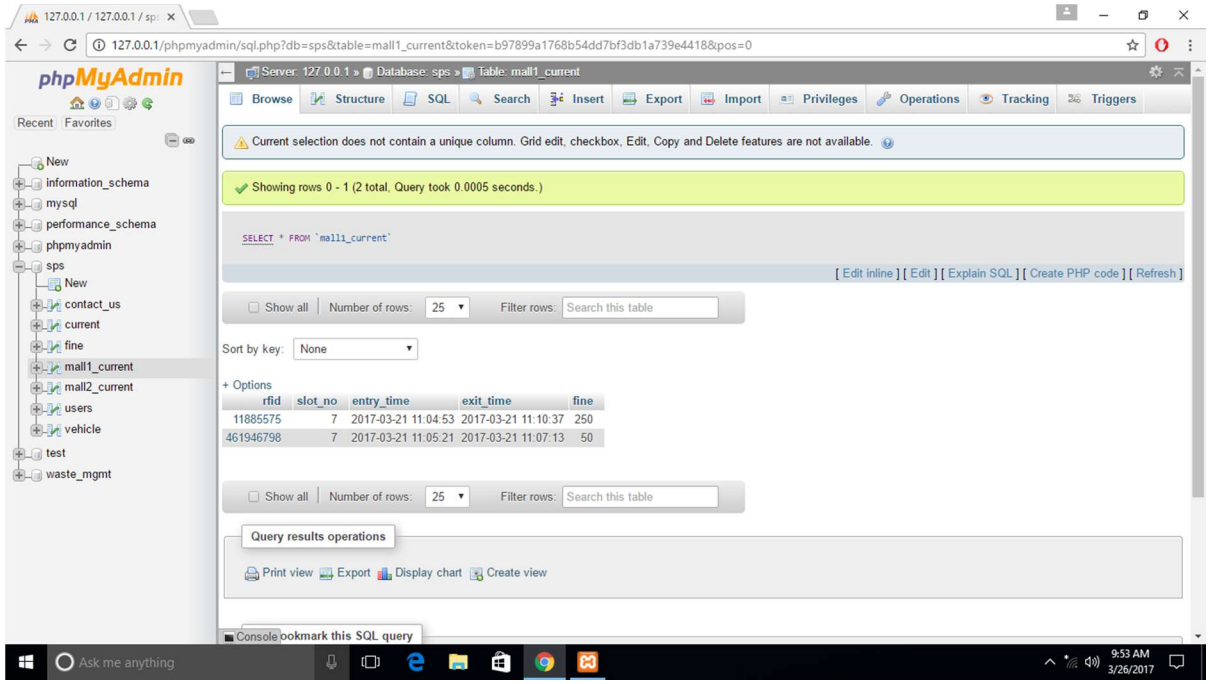

Fig 9.2.6 Database sps: Table: mall1_current Tuples

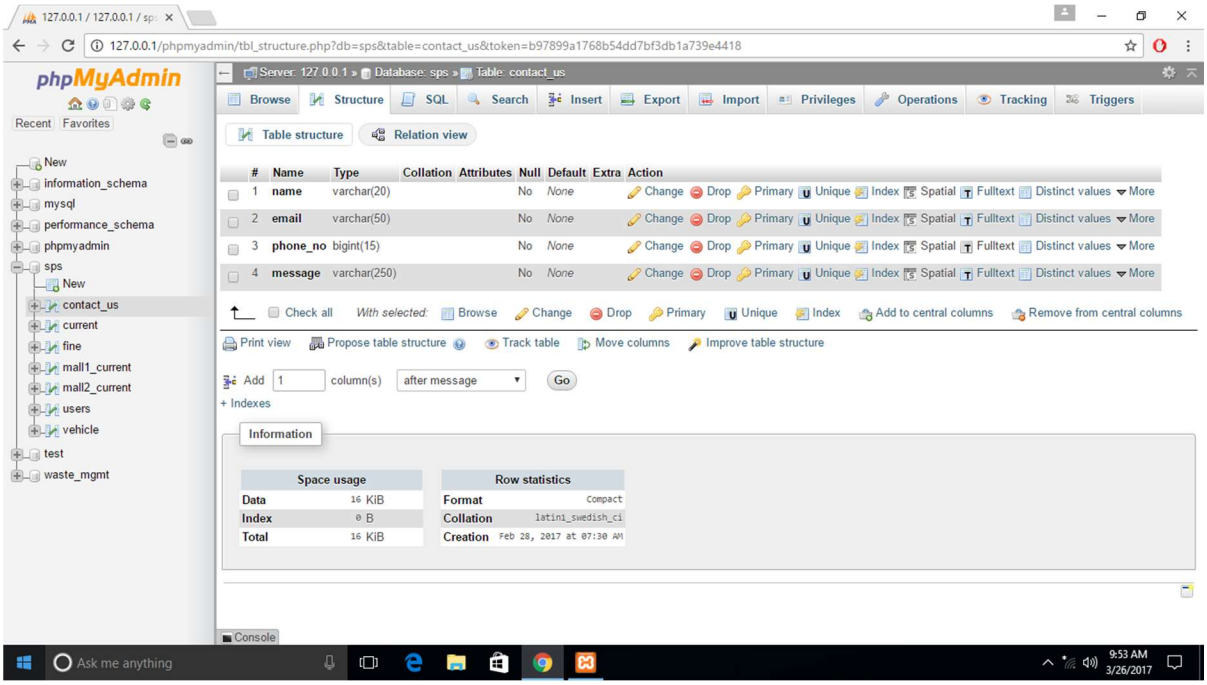

Fig 9.2.7 Database sps: Table: contact_us

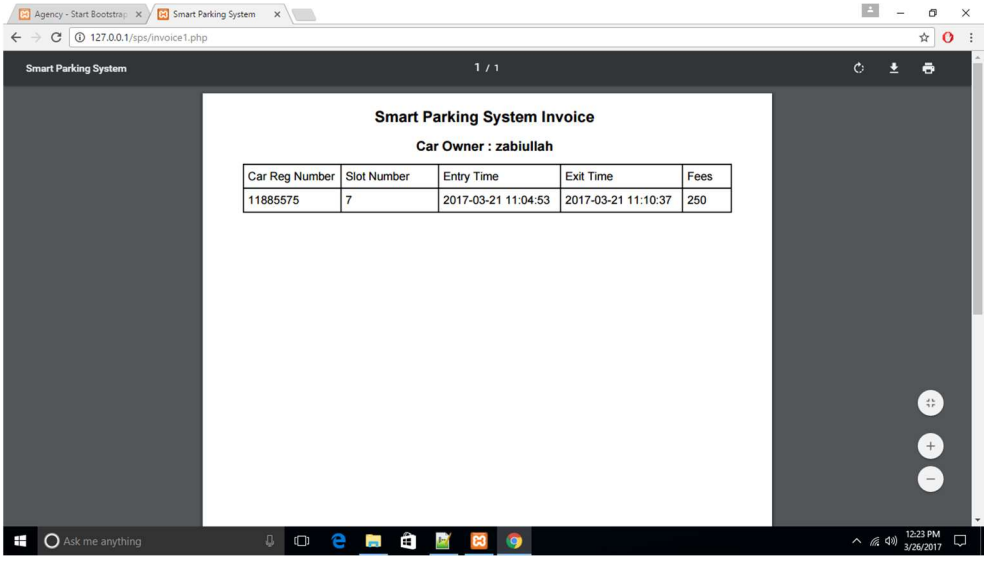

Fig 9.2.8: Report Generation in Pdf

9.3 Front-end

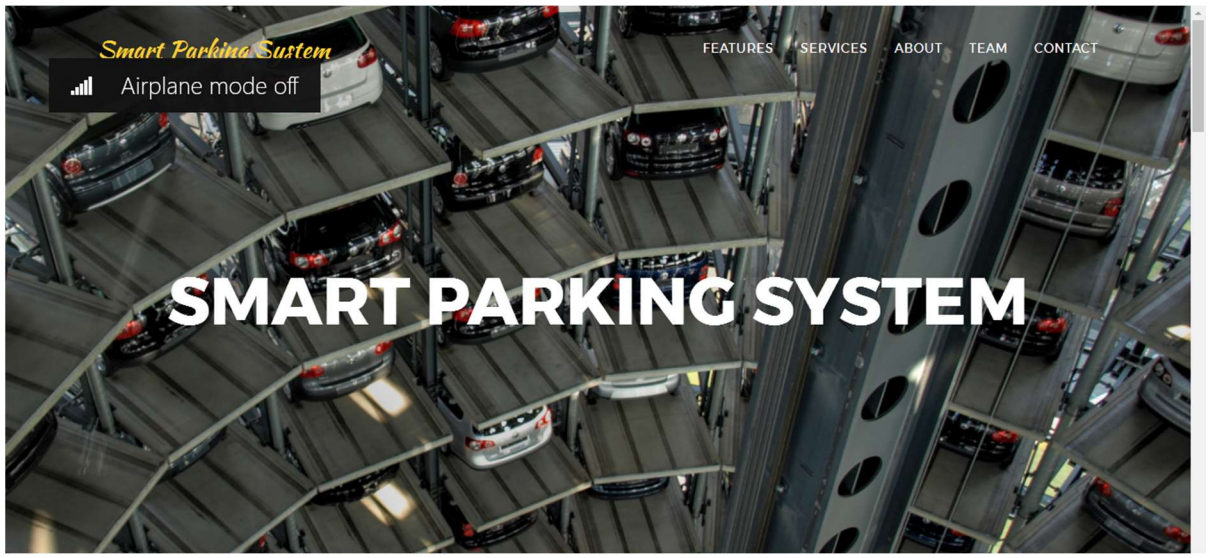

Fig 9.3.1: Front end – Introduction page

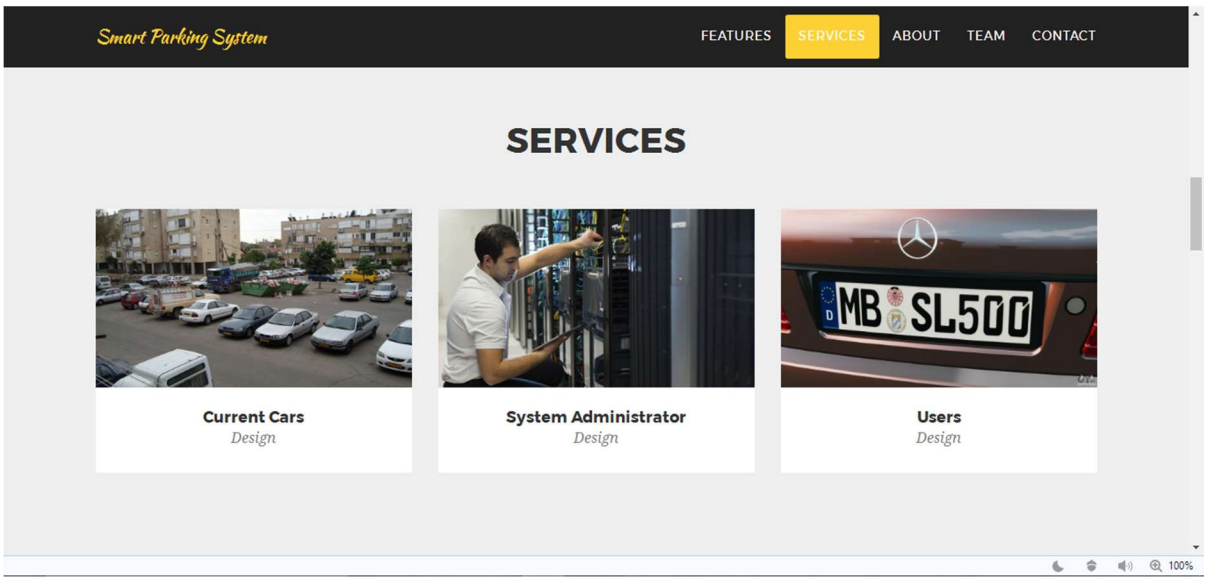

Fig 9.3.2: Front end – Services provided

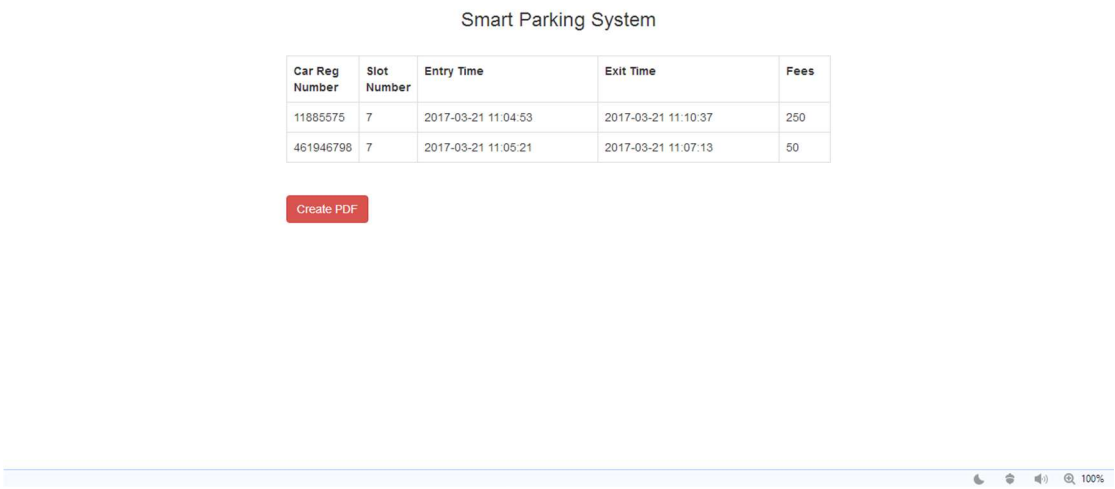

Fig 9.3.3 Administrator Smart Parking System Log Record

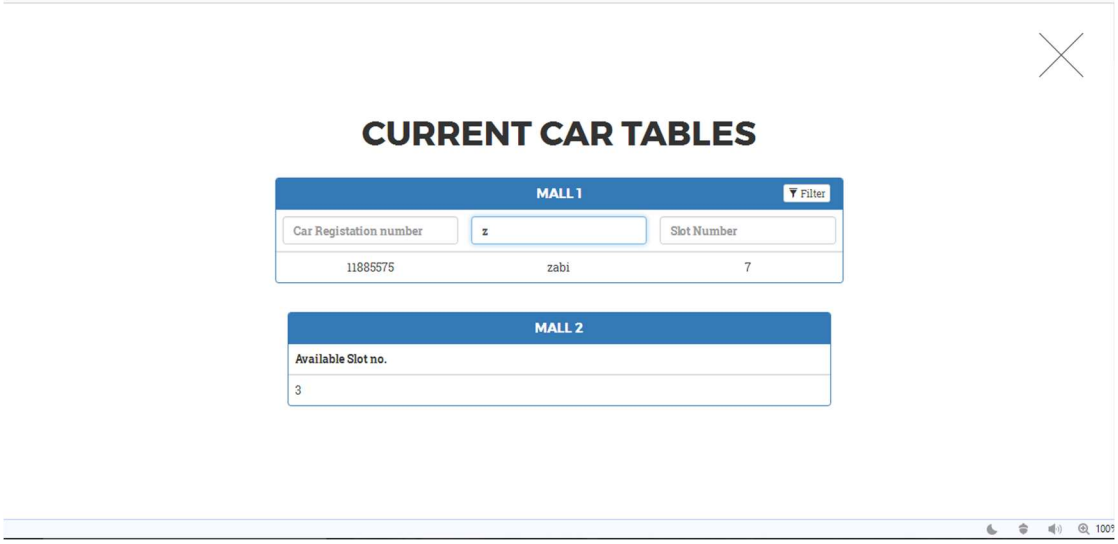

Fig 9.3.4 Mall1 and Mall2 Parking details with filtering enabled.

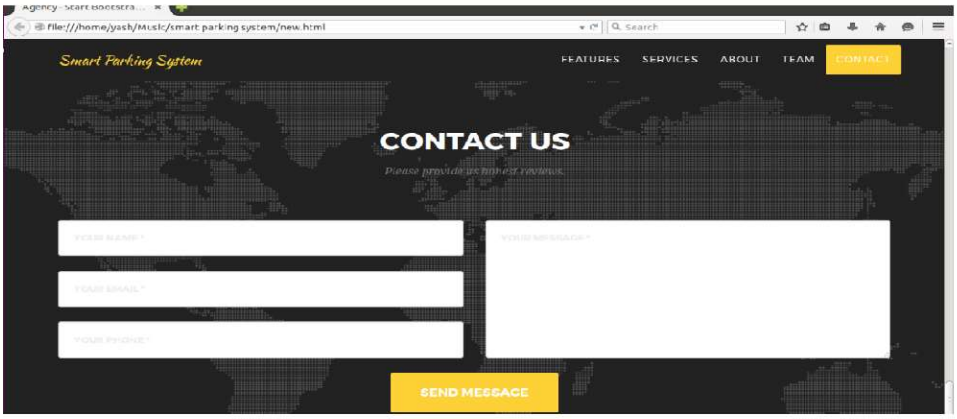

Fig 9.3.5: Contact form

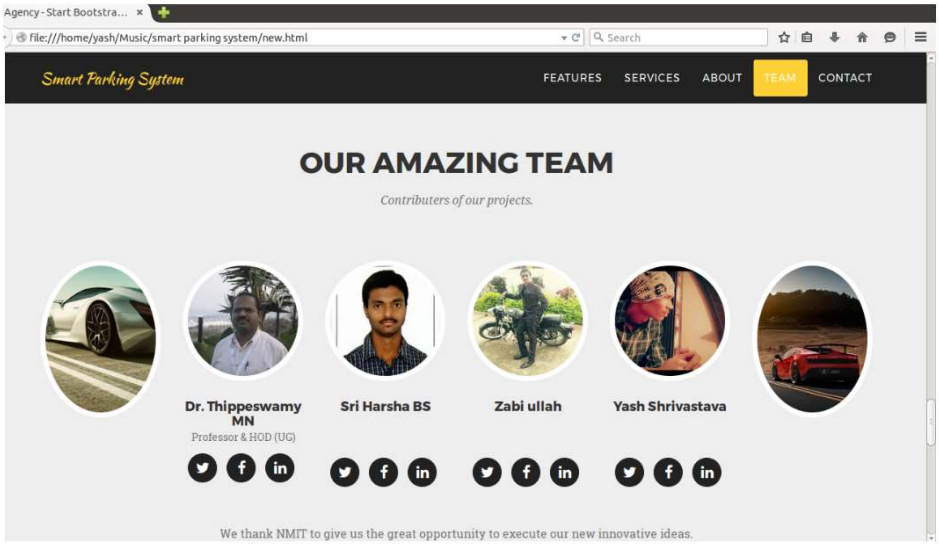

Fig 9.3.6: Team page

CHAPTER 10

CONCLUSION

In this project, it has been proved that by utilizing RFID readers and RFID labels with a centralized database system, all the parking-lots in a city could be operated in an economical, dynamic and fast way. The given prototype will provide an atomized operation and controlling ability for all of the parking-lots in a city by using standardized devices. At the end of each month, the total fees for each of the parking-lot member drivers can be calculated. The additional facility is also available that these fees could be drawn from their bank accounts and transferred to the parking-lots' accounts automatically, issuing an invoice for each transaction.

Via such a system, personnel costs will be cut off because of less direct human intervention. As is in the case of ATM machines, realization of unmanned, completely atomized parking-lots will be possible in the near future. Without having to stop vehicles, check-ins and check-outs will be possible. In that way, there won't be any traffic jam problems and with limited infrastructure traffic can be managed dynamically.

According to the traditional parking lot systems, drivers will no longer be bound to take and process parking tickets during check-ins and check-outs. Moreover there won't be any ticket-jamming problems either. Vehicle owners will have the facility where they will not have to make payments upon every check-out. This will minimize congestion to greatest possible extent.

The following conclusions will lead to clear understanding of the project “Smart Parking System Using Internet of Things (IoT)

- The proposed system will overcome the current challenges thus avoiding the congestion with in the parking lot itself.
- This parking system can be either used in Public places or at apartments thus making it area independent.
- Minimizing Man power thus maximizing throughput.
- Exact calculation of the time interval between the Entry time and Exit time and automated report/bill generation or mechanism.

- This system will be developed for the very first time in the world with these many unique plugins.
- A Win-Win situation for both Users as well as the authority members.
- As per our assumption that RFID itself is the car's registration number, the proposed system can thus be made reality on the following enhancements.
  i) Create One-One Mapping. (Recommended) between the RFID and the CAR's Registration Number
  ii) Image and Video processing. (Secondary Approach). : Use a high quality camera to record the image and with appropriate image processing algorithms derive the car's registration number and continue with the process.

CHAPTER 11

REFERENCES

[1] Smart Parking System for Internet of Things, 2016 IEEE conference on Consumer Electronics, Chungsan Lee, Youngtak Han, Soobin Jeon, Dongmahn Seo*, Inbum Jung

[2] https://www.ukessays.com/essays/information-technology/literature-review-on-car-parking-system-information-technology-essay.php

[3]. http://www.mdpi.com/1424-8220/14/12/22372/pdf

[4]https://www.computer.org/csdl/proceedings/wowmom/2013/5827/00/06583499.pd

[5].http://www.nominet.uk/researchblog/creating-a-smart-parking-system-using-our-iot-tools/

[6]. http://www.sciencedirect.com/science/article/pii/S014036641630072X

[7]. http://electronicsofthings.com/expert-opinion/smart-city-solutions-smart-parking-lots

[8]. ZigBee: Wireless control that simply works http://www.zigbee.org page (pp 1-15)

[9]. Zig Bee Standards Overview.

[10]. http://www.smartparking.com/technologies/rfid-solutions

[11].http://www.seminarsonly.com/Engineering-Projects/Computer/Automated-Parking-System.php

[12]. http://www.en.wikipedia.org/text/Smart_Parking_Syetm

[13]. www.googlescholars.com

[14]. International Journal of Scientific and Research Publications, Volume 2, Issue 10, October 2012 1 ISSN 2250-3153 www.ijsrp.org Comparative Implementation of Automatic Car Parking System with least distance parking space in Wireless Sensor Networks Mala Aggarwal, Simmi Aggarwal, R.S.Uppal.

[15]. International Journal of Machine Learning and Computing, Vol. 2, No. 2, April 2012 Automatic Parking Management System and Parking Fee Collection Based on Number Plate Recognition M. M. Rashid, A. Musa, M. Ataur Rahman, and N. Farahana, A. Farhana.

[16]. Password Protected Locking System Using Arduino Sriharsha B S[1] , Zabiullah[2] , Vishnu S B[3] and Sanju V[4] Copy Right © BIJIT –2016; January - June, 2016; Vol. 8 No. 1; ISSN 0973 –5658

[17]. https://www.arduino.cc/en/reference/libraries

[18]. https://tcpdf.org