INDEX

CHAPTER 1 – INTRODUCTION

1.1 INTRODUCTION

India is a growing country that has seen substantial growth in the population and advancement in the ways things-are-done. It is also true that every city these days, aspires to become a smart city. Find a problem and use technology to solve it in a way that it makes our life easier. One such problem has been resolved through the below mentioned concept.

The proposed smart parking management system is an IoT module used to monitor the parking spaces in the city and signalize the availability of parking space. Consistent efforts are being made in the field of IoT to maximize the productivity and reliability of urban infrastructure. We present an IoT based cloud integrated smart parking system. The proposed Smart Parking system consists of an on-site deployment of an IoT module that is used to monitor and signalize the state of availability of each single parking space.

1.2 PROBLEM STATEMENT

Trouble in Finding Vacant Spaces, Quickly finding an empty space in a multilevel parking garage is troublesome if not unthinkable, particularly on ends of the week or open occasions. Discovering spaces amid ends of the week or open occasions can take over 10 minutes for around 66% of guests. Stadiums or shopping center are swarmed at pinnacle periods, and trouble in finding empty openings at these spots is a noteworthy issue for clients. Inadequate car parking space \ prompt activity blockage and driver disappointment.

1.3 NEED FOR PROPER SYSTEM

Since the past couple of years large advancements have been made in making smart cities a reality. The growth of Internet of Things and Cloud technologies have given rise to new possibilities in terms of smart cities. Smart parking facilities and traffic management systems have always been at the core of constructing smart cities. The system that we propose provides real time information regarding availability of parking slots in a parking area. The proposed smart parking system is indented to improve the parking facilities of a city and thereby aiming to enhance the quality of life of its people.

1.4 OBJECTIVES

a) Drivers find accessible parking spots close to them.
b) Less number of drivers seeking to stop, in this way decreases the movement clog.
c) Avoids air contamination and a dangerous atmospheric deviation.
d) Scalable, hearty, and solid.
e) Reduces the driver stretch and enhances the urban range.
f) Provides devices to streamline the parking spot administration accurately discover the vehicle inhabitancies progressively.

1.5 SCOPE

In future works, this framework can be enhanced by including different applications, For Example, internet booking by utilizing GSM. The driver or client can book their parking area at home or while in transit to the shopping center. This can diminish the season of the client to seeking the empty parking area. As a further review, distinctive sensor frameworks can be added to enhance this framework to distinguish the question and guide the driver or clients speediest. We will attempt to decrease the mechanical structure and attempt to make it ecofriendly.

CHAPTER 2 – LITERATURE SURVEY

2.1 EXISTING SYSTEM

The existing parking management system is a manual system. If we compare the proposed system to the existing system, we come to realize that the proposed system is extremely efficient and that the existing system posses lots of cons. The existing system involves cash transactions, manual entries, and parking personnel for managing the parking areas. There is also no certainty of the

2.2 PROPOSED SYSTEM

The presented smart parking management system does not involve any manual intervention. Integration of sensors and cameras ease the complexity. A free area of land meant for parking purpose have 80 parking slots, 60 out of which are for cars and 20 for two wheelers. Each slot is allocated an unique number. The parking area has two separate gates for entry and exit of vehicles. An infrared sensor is connected to each gate which controls the activity of turnstile gate. When a vehicle comes near the gate to enter, the IR sensor at entry gate detects the vehicle. It then checks the system for free space in the parking area. If found, it allots the car with the slot number and marks the spot as occupied. It also opens the gate and lets the vehicle in.

During the exit of the vehicle, IR sensors detect the vehicle and then the servo motor is used to turn the turnstile gate for the vehicle to move out. At the same time the website is updated and slot is made vacant.

2.3 FEASIBILITY STUDY

2.3.1 TECHNICAL FEASIBILITY

The technical feasibility in the proposed system deals with the technology used in the system. It deals with the hardware and software used in the system whether they are of the latest technology or not. It happens that after a system is prepared a new technology arises and the user wants the system based on that technology. This system use HTML, CSS as front end technology and MySQL server as backend technology. The SMART PARKING MANAGEMENT SYSTEM is technically feasible.

Technical feasibility centers on the existing computer system (hardware, software, etc.) and to what extent it can support the proposed addition. If the budget is a serious constraint, then the project is judged not feasible.

Technical feasibility speaks about the existing hardware and the software the we are using and the deviations that we have to make from the existing one, as we are developing the application using python there is no change in hardware that we are using. So, we can say that this application is technically feasible as there is no change in the configuration more over it is cost effective.

2.3.2 ECONOMICAL FEASIBILITY

Economic analysis is the most frequently used method for evaluating the effectiveness of a new system. More commonly known as cost/benefit analysis. MySQL database is easily available script server side in the internet.

This procedure is to determine the benefits and savings that are expected from a candidate system and compare them with costs. If benefits outweigh costs, then the decision is made to design and implement the system. Otherwise, further justification or alterations in the proposed system will have to be made if it is to have a chance if being approved . this is an ongoing effort that improves in accuracy at each phase of the system life cycle.

Economic feasibility talks about the benefits that which we get from this project. Here with the introduction of this online process we are not only reducing the time taken for the registration of the entrants. We even reduce the burden on the administrator. As this project is not only reducing the tome but also the work burden of the user we say that this project is economically feasible.

2.3.3 OPERATIONAL FEASIBILITY

The project has been developed in such a way that it becomes very easy even for a person with little computer knowledge to operate it. People are inherently resistant to change and computers have been known to facilitate change. It is understandable that the introduction of a candidate system requires special effort to educate, sell and train the staff on new ways of conducting business.

As this project is a user friendly version there is not much training for the people to use. This product is not only making the task of the administrator easy but it is reducing the time that is taken otherwise. So we say that this product is operationally feasible.

CHAPTER 3 – REQUIREMENT ANALYSIS

3.1 METHOD USED FOR REQUIREMENT ANALYSIS

Different techniques have been used to perform requirement analysis for the proposed model. It included recognizing the customer needs and devising a system/model that would resolve the issues to meet the needs of the customer. The next step is to evaluate the feasibility of the proposed system in terms of technical, economical and operational feasibility. It finally involves looking at the business aspects of the model.

3.2 FUNCTIONAL REQUIREMENTS

Our proposed system comprises of web application for the parking area admin, parking operators and users.

a) Web application for parking admins
   - Define new parking areas, define number of parking slots and the parking slots per minute/hour and other details.
   - Modify data of existing parking slots
   - View data of all registered parking areas.

b) Web application for parking operators
   - Send slot number and reservation password to the central server.
   - Issue bills to the users on checkout

c) Web application for the users
   - Register for service and enter personal and vehicle details.
   - Find a parking area from the list of areas, registered by parking admins.
   - View the details of a selected parking area such as the name, price per minute, number of total available lots.
   - Reserve an available parking lot and specify duration of reservation.

3.3 NON-FUNCTIONAL REQUIREMENTS

a) The central repository should be platform independent so that it can be accessible and store application data via the web application and the mobile app.
b) The server should be able to handle concurrent requests from different users.
c) The system should provide confidentiality for user data using database encryption and local encryption to protect data in the event of device theft (laptop/handheld device)

3.4 SYSTEM SPECIFICATIONS

3.4.1 HARDWARE SPECIFICATIONS

a) Arduino UNO: It is the machine that is embedded in the system. The microcontroller used is ATmega328P. Arduino UNO is an open source technology for developers and has good community support. This can be programmed using Arduino IDE which has an inbuilt function that brings down the effort of the user. Python is used as the go to programming language.
b) Infrared(IR) Sensors: IR sensors is an electronic device that detects infrared radiations. When an object comes close to this sensor, the IR rays from transmitter reflect off the object and detected by receiver. Here, the IR sensors are used to detect the vehicle passing through the turnstile gate.
c) Servo Motor: It is used to turn the turnstile gate up and down. This can be accessed by importing the SERVO library from pyFirmata.
d) Display: We use this LCD screen to display the slot number where the vehicle needs to be parked down.

e) Jumper wires: These are used for making connections between items on the breadboard and Arduino's header pins. Used to wire up the whole circuit.
f) BreadBoard: It is used to construct electronic circuits without involving the use of soldering iron. Components are pushed into the sockets on the breadboard and then connections are made using the jumper wires.
g) USB A to B cable

3.4.2 SOFTWARE SPECIFICATIONS

Arduino sketches are generally written in languages relatable to C++ which are then compiled and stored in the flash memory. It can also use other high level languages to program the Arduino. Here, we use Python as a main source code language because it is robust and provides easy usage of code lines. It is also open source, has availability of vast libraries, simple syntax, the most accessible software, maintenance can be handled in a great way and debugging can be done easily.

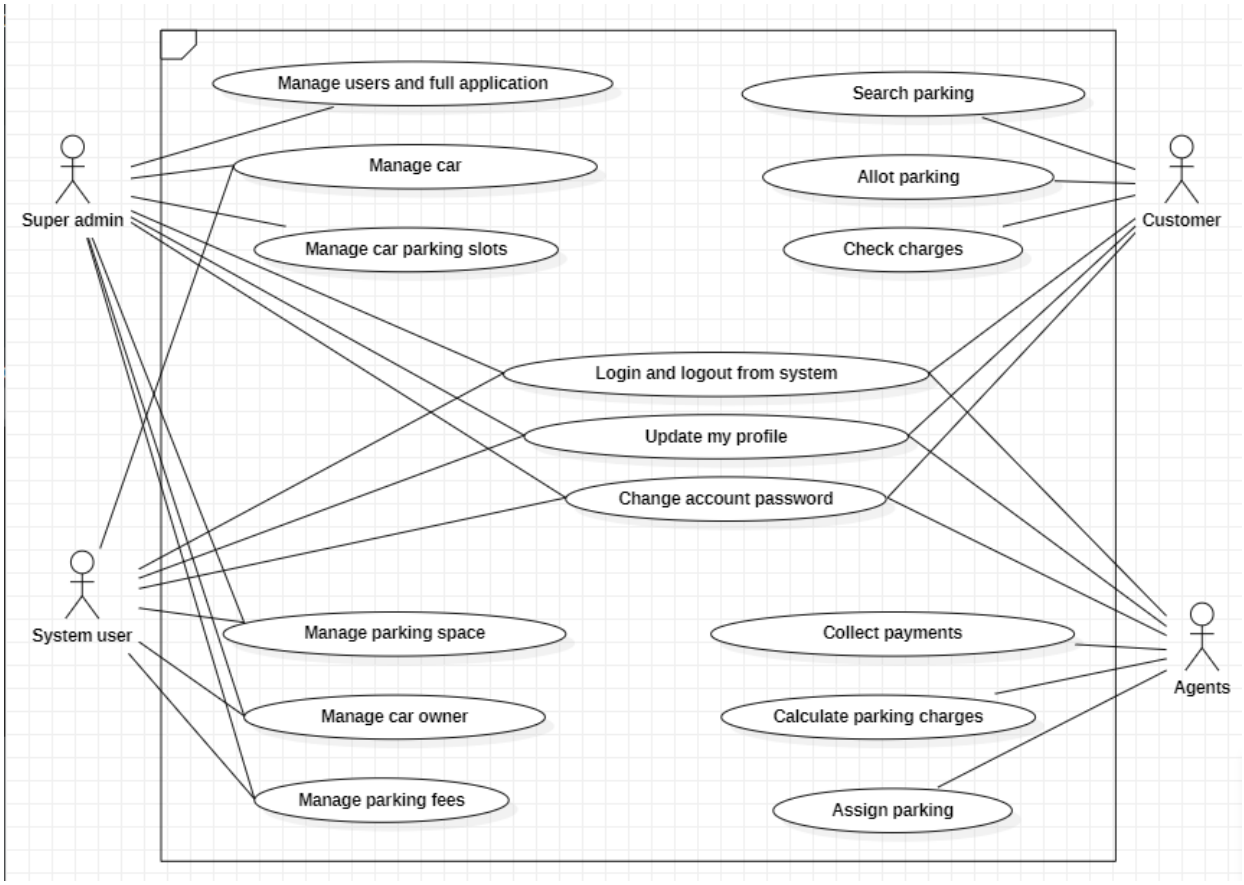
One way to use python and Arduino is by using the Firmata protocol. This protocol completely transfers the control of Arduino to the external software.

How does this happen? Arduino program is developed that is controlled using a serial port. The fermata protocol solved the complication of uploading the Arduino sketch everytime there a minor change. Once burnt successfully Arduino is ready to accept connection from other programs or software like python that supports serial communication.

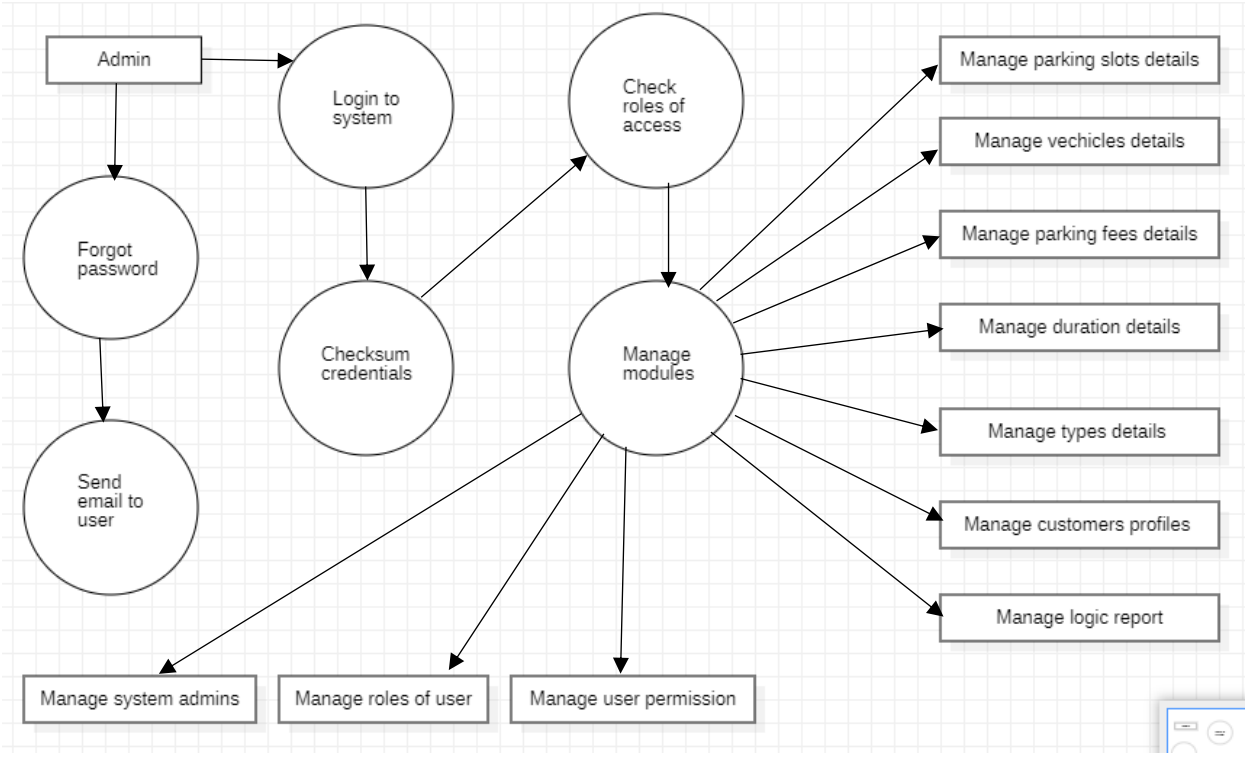
Here, we use Python as a substitute for the Arduino IDE sketch by using the pyserial library as it supports serial communication. For python and Arduino to work together we also need a library that will support the Firmata protocol. For this, pyFirmata library is imported. This library is built on the pyserial library.

CHAPTER 4 – DESIGN

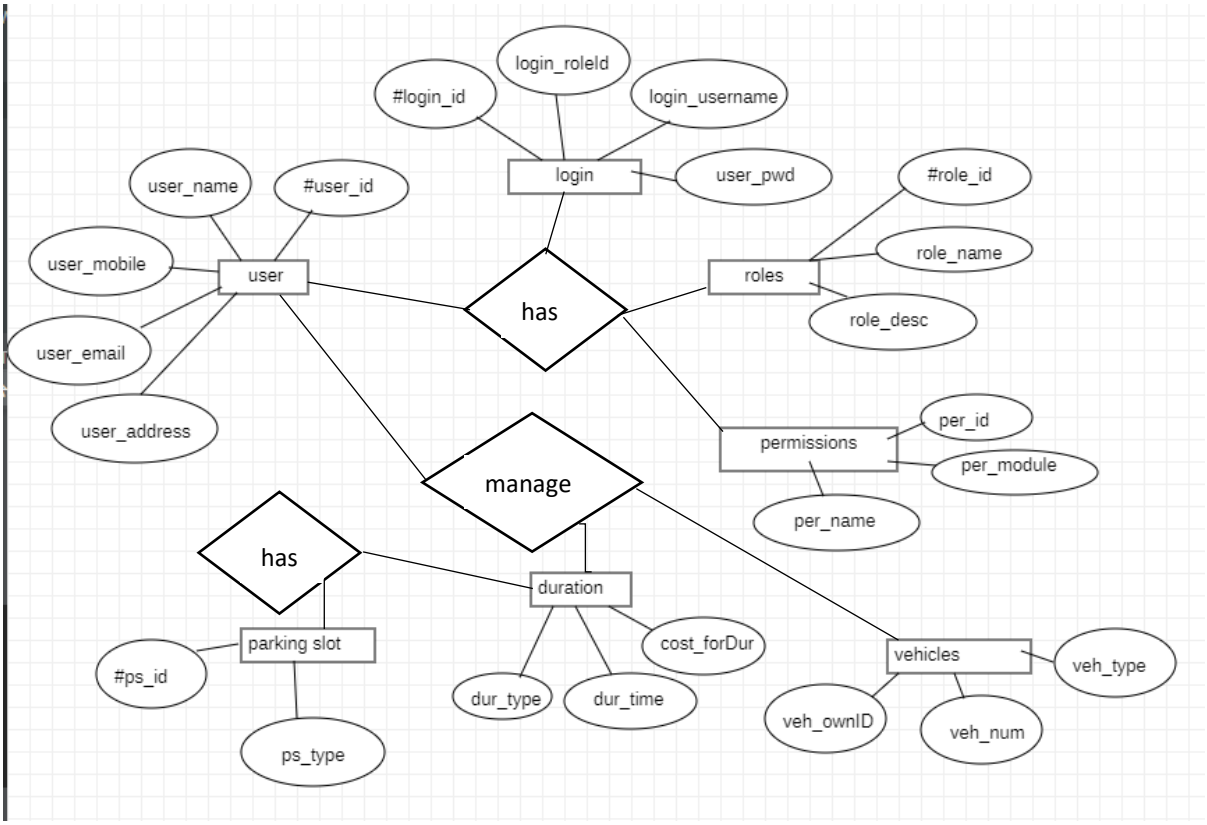
4.1 USE CASE DIAGRAM



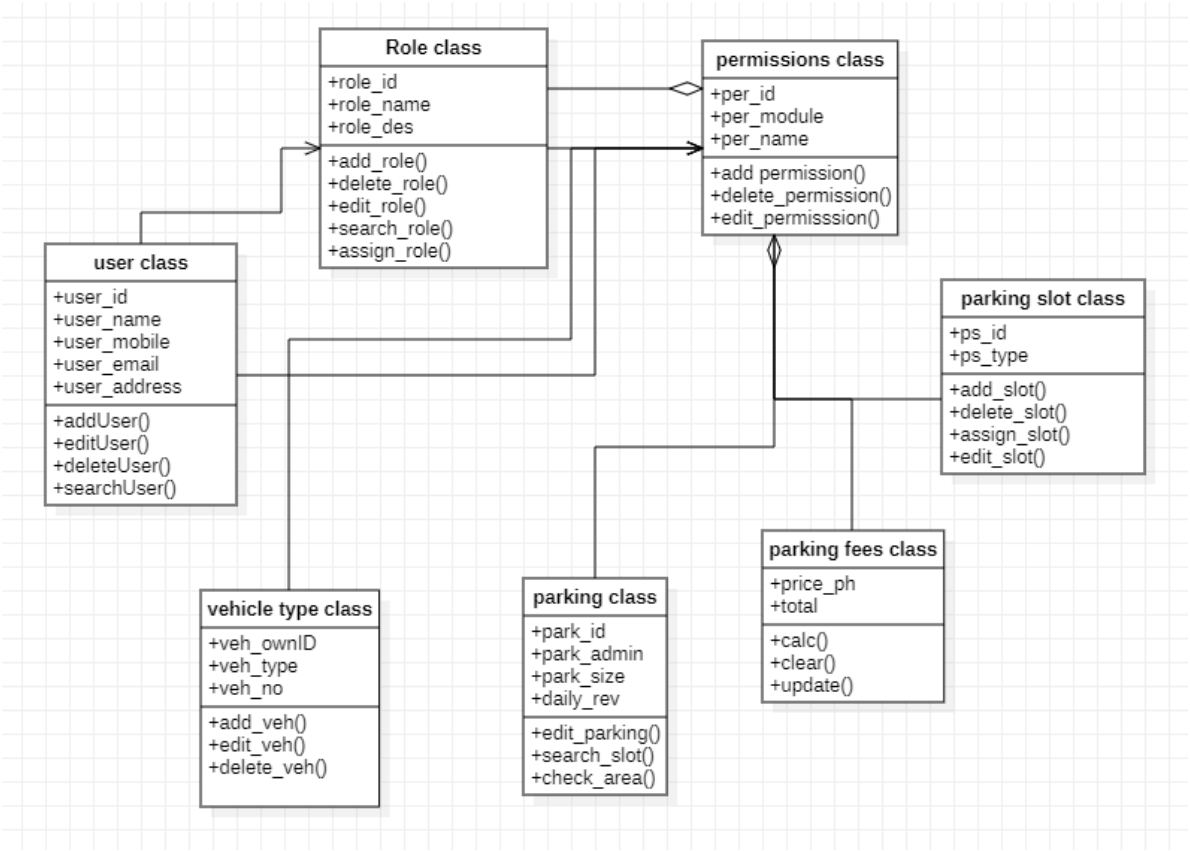
4.2 DATAFLOW DIAGRAM



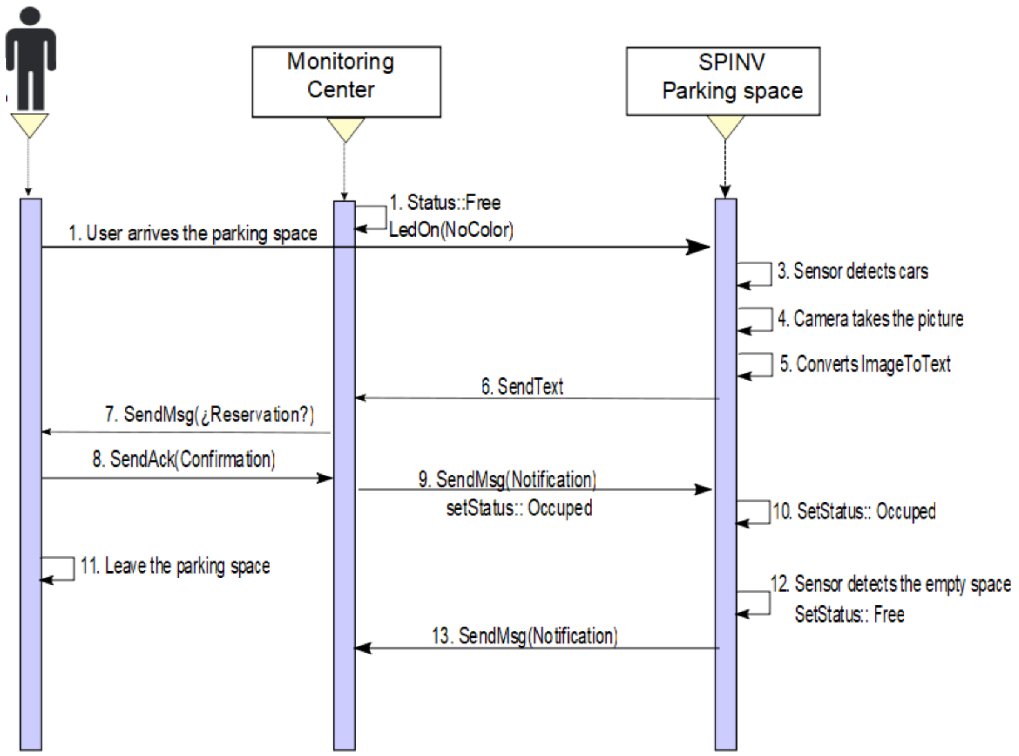
4.3 DATABASE DESIGN (ER DIAGRAM)



4.5 DETAILED CLASS DIAGRAM

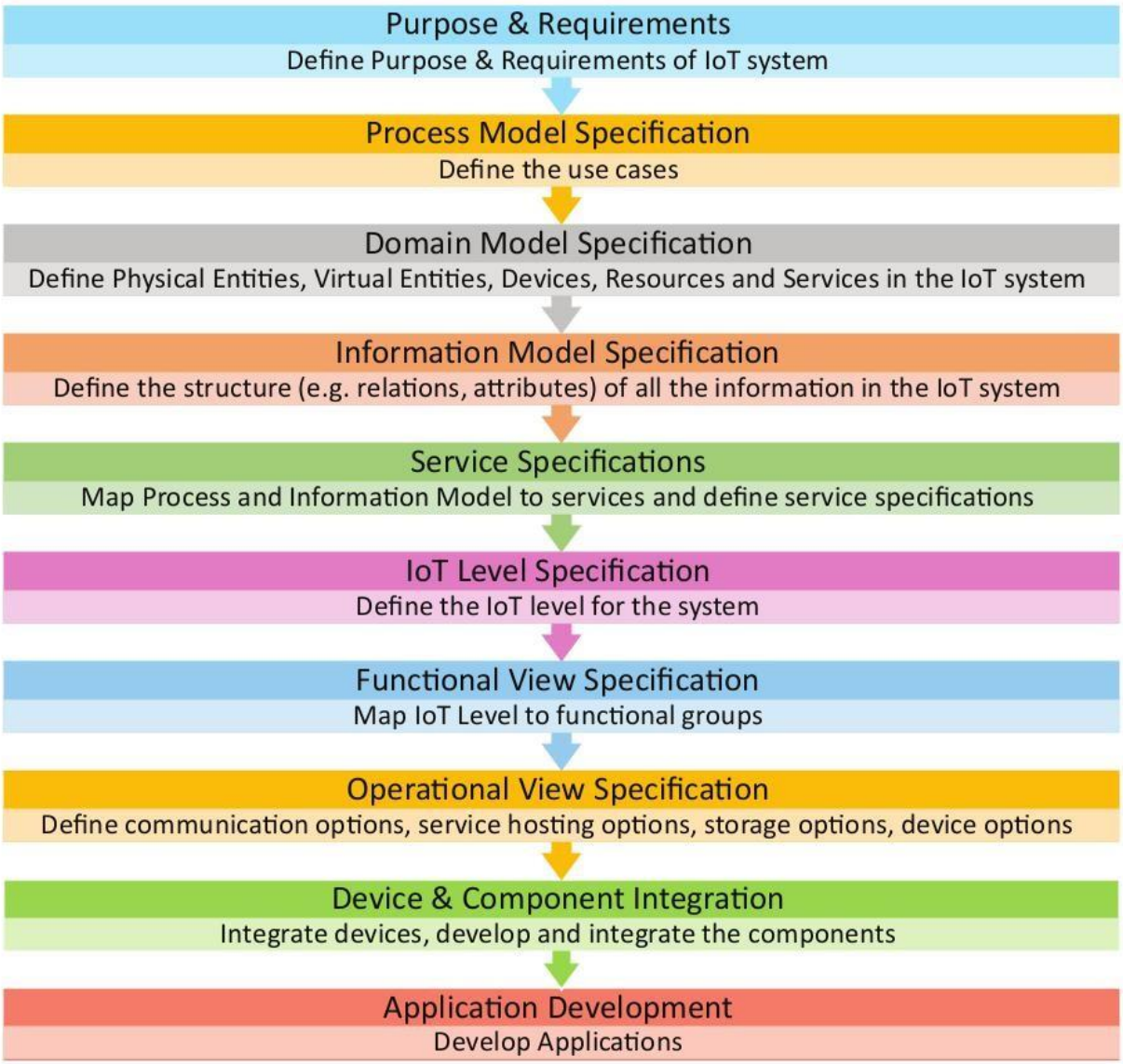


4.6 SEQUENCE DIAGRAM



CHAPTER 5 – IoT METHODOLOGIES

IoT Design Methodology – Steps



Step 1: Purpose & Requirements Specification

The first step in IoT system design methodology is to define the purpose and requirements of the system. In this step, the system purpose, behavior and requirements (such as data collection requirements, data analysis requirements, system management requirements, data privacy and security requirements, user interface requirements, ...) are captured.

Step 2: Process Specification

The second step in the IoT design methodology is to define the process specification. In this step, the use cases of the IoT system are formally described based on and derived from the purpose and requirement specifications.

Step 3: Domain Model Specification

The third step in the IoT design methodology is to define the Domain Model. The domain model describes the main concepts, entities and objects in the domain of IoT system to be designed. Domain model defines the attributes of the objects and relationships between objects. Domain model provides an abstract representation of the concepts, objects and entities in the IoT domain, independent of any specific technology or platform. With the domain model, the IoT system designers can get an understanding of the IoT domain for which the system is to be designed.

Step 4: Information Model Specification

The fourth step in the IoT design methodology is to define the Information Model. Information Model defines the structure of all the information in the IoT system, for example, attributes of Virtual Entities, relations, etc. Information model does not describe the specifics of how the information is represented or stored. To define the information model, we first list the Virtual Entities defined in the Domain Model. Information model adds more details to the Virtual Entities by defining their attributes and relations.

Step 5: Service Specifications

The fifth step in the IoT design methodology is to define the service specifications. Service specifications define the services in the IoT system, service types, service inputs/output, service endpoints, service schedules, service preconditions and service effects.

Step 6: IoT Level Specification

The sixth step in the IoT design methodology is to define the IoT level for the system.

Step 7: Functional View Specification

The seventh step in the IoT design methodology is to define the Functional View. The Functional View (FV) defines the functions of the IoT systems grouped into various Functional Groups (FGs). Each Functional Group either provides functionalities for

interacting with instances of concepts defined in the Domain Model or provides information related to these concepts.

Step 8: Operational View Specification

The eighth step in the IoT design methodology is to define the Operational View Specifications. In this step, various options pertaining to the IoT system deployment and operation are defined, such as, service hosting options, storage options, device options, application hosting options, etc.

Step 9: Device & Component Integration

The ninth step in the IoT design methodology is the integration of the devices and components.

Step 10: Application Development

The final step in the IoT design methodology is to develop the IoT application.

SMART PARKING MANAGEMENT SYSTEM

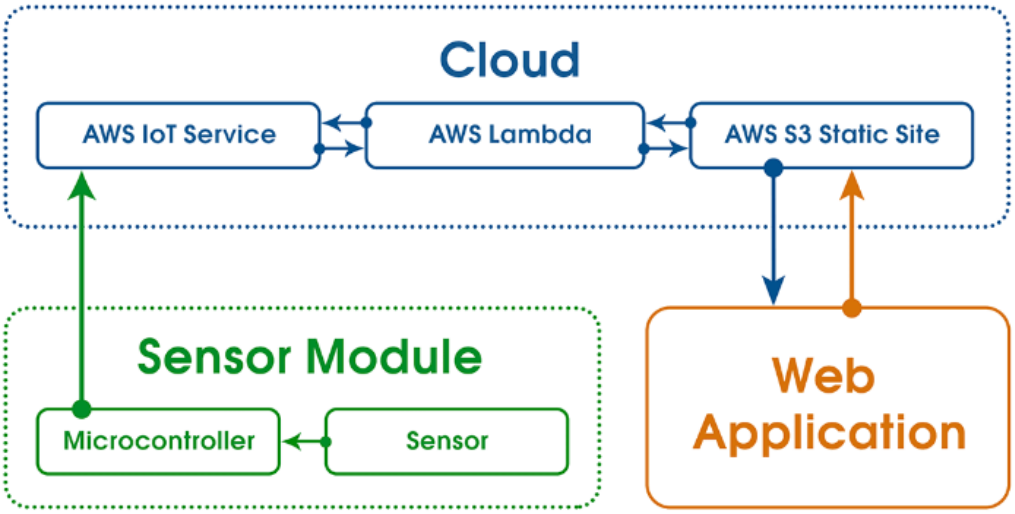
1. PURPOSE AND REQUIREMENT

Quickly finding an empty space in a multilevel parking garage is troublesome if not unthinkable, particularly on ends of the week or open occasions. Discovering spaces amid ends of the week or open occasions can take over 10 minutes for around 66% of guests. Stadiums or shopping center are swarmed at pinnacle periods, and trouble in finding empty openings at these spots is a noteworthy issue for clients. Inadequate car parking space \ prompt activity blockage and driver disappointment.

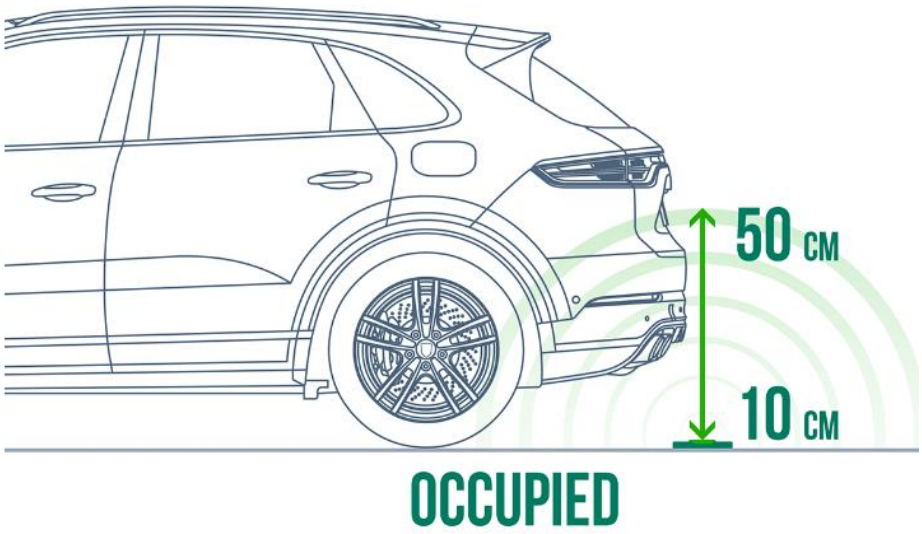
The growth of Internet of Things and Cloud technologies have given rise to new possibilities in terms of smart cities. Smart parking facilities and traffic management systems have always been at the core of constructing smart cities. The system that we propose provides real time information regarding availability of parking slots in a parking area. The proposed smart parking system is indented to improve the parking facilities of a city and thereby aiming to enhance the quality of life of its people.

2. PROCESS SPECIFICATION

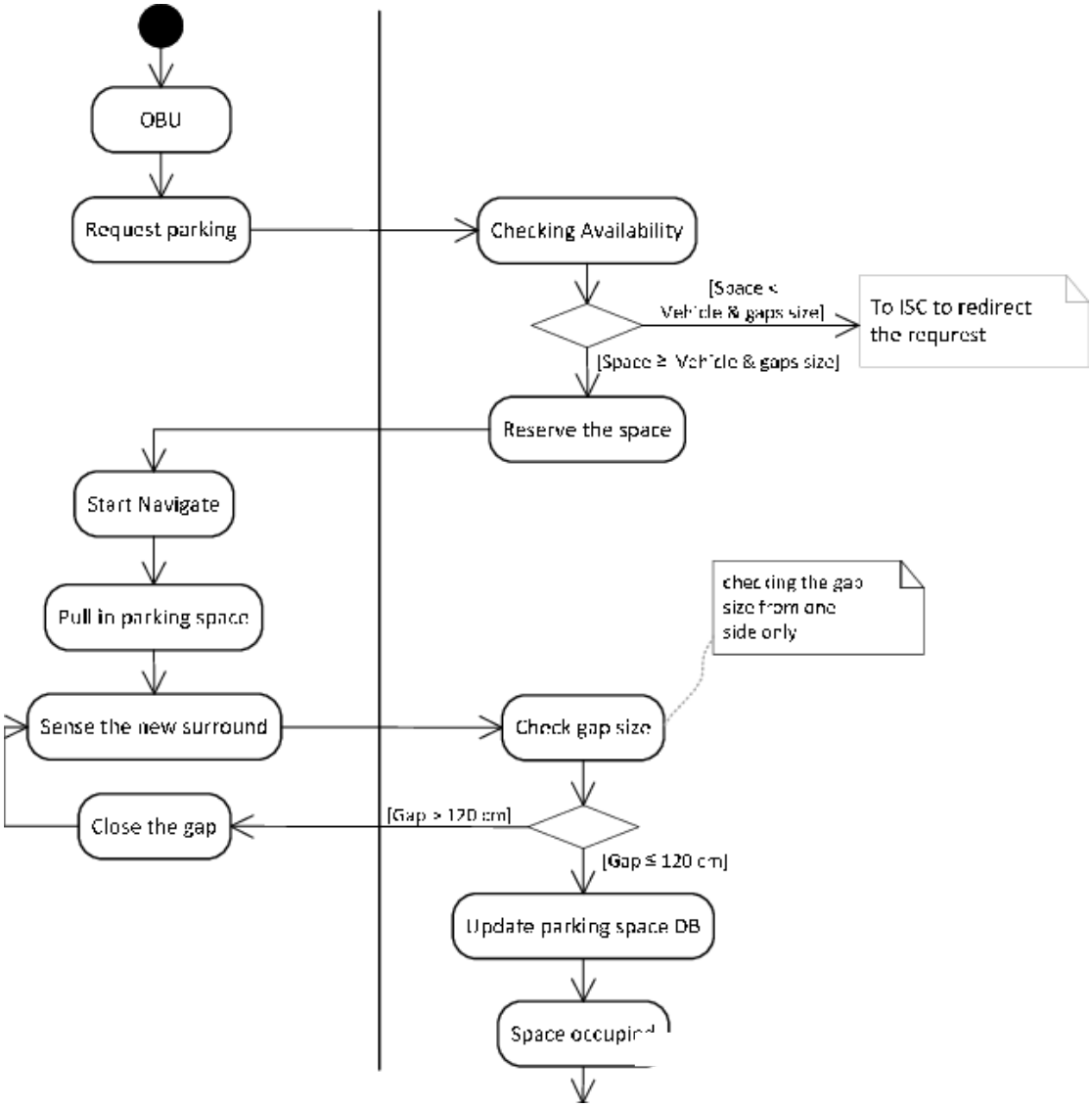
The presented smart parking management system does not involve any manual intervention. Integration of sensors and cameras ease the complexity. A free area of land meant for parking purpose have 80 parking slots, 60 out of which are for cars and 20 for two wheelers. Each slot is allocated an unique number. The parking area has two separate gates for entry and exit of vehicles. An infrared sensor is connected to each gate which controls the activity of turnstile gate. When a vehicle comes near the gate to enter, the IR sensor at entry gate detects the vehicle. It then checks the system for free space in the parking area. If found, it allots the car with the slot number and marks the spot as occupied. It also opens the gate and lets the vehicle in. During the exit of the vehicle, IR sensors detect the vehicle and then the servo motor is used to turn the turnstile gate for the vehicle to move out. At the same time the website is updated and slot is made vacant.



 # HOW THINGS WORK?



 # # VEHICLE DETECTION RANGE



 # ACTIVITY DIAGRAM FOR SMART PARKING SYSTEM

3. DOMAIN MODEL SPECIFICATION

Different entities present in this prototype are:

Vehicle : Looks for a parking space

Parking class: Group of parking spaces

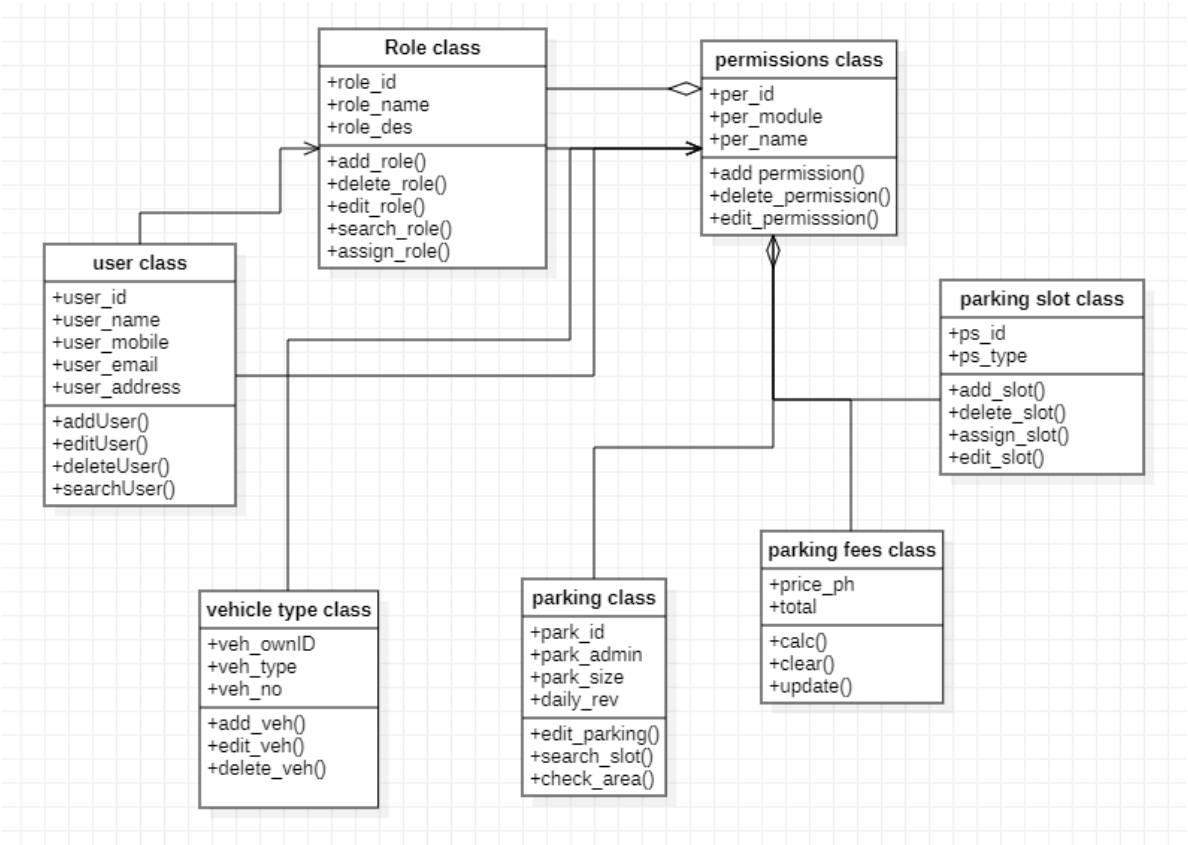
Parking slot: Single parking space

Database: Store information about parking space

Role: different users that interact with the software

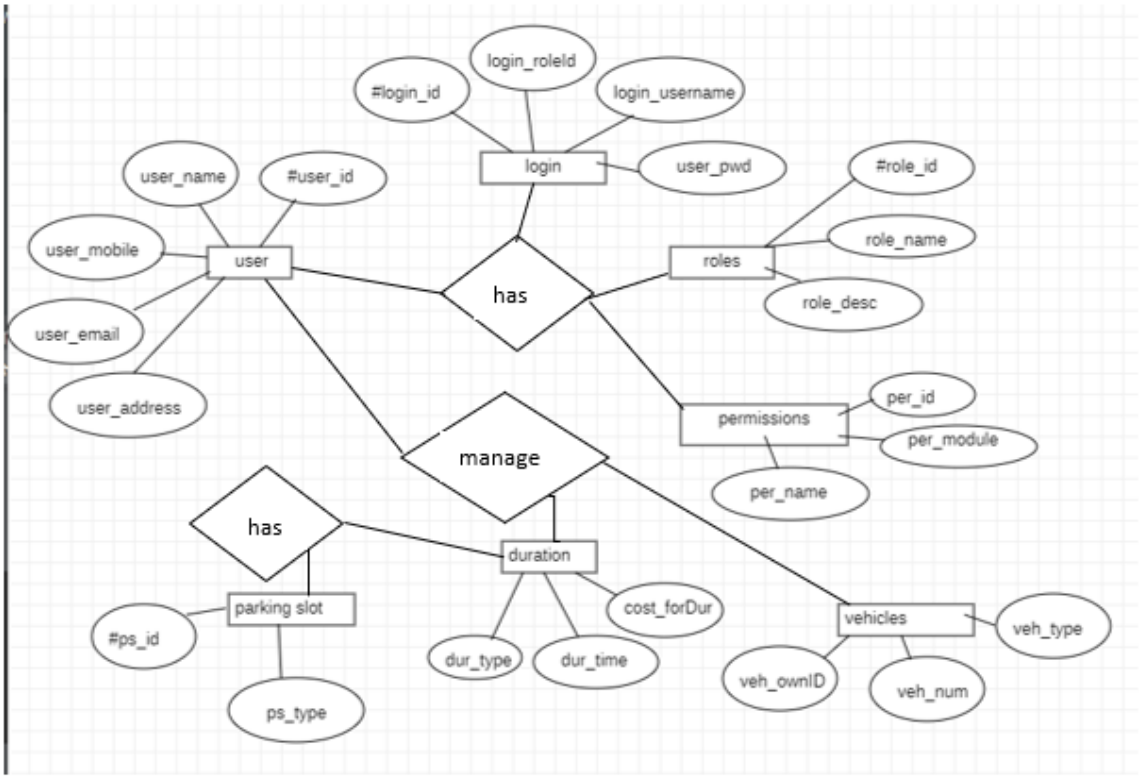
Display: A mobile application or the OLED display

- The car entity uses the parking lot to park itself in a vacant parking space(parking lot extends parking space).
- The database and parking lot has one to many relationship as one database is used to store multiple parking lots information.
- Finally, display entity shows the information from the database.

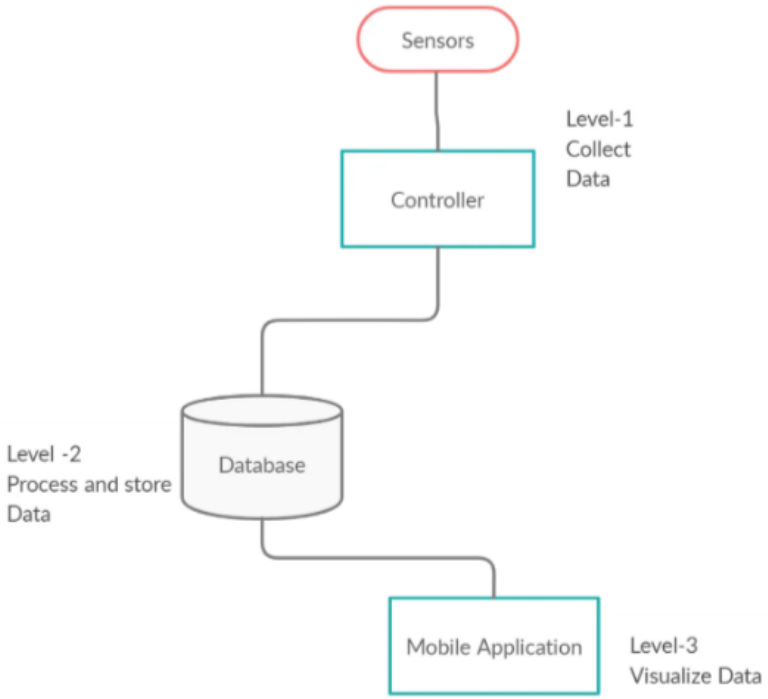


4. INFORMATION MODEL SPECIFICATION

The ER diagram for the database to be used is shown below:



The system uses a 3-level architecture, where the sensors collect data (level-1) and sends data to a cloud platform(level-2) to process and store the data. The processed data is displayed on a web application(level — 3).



5. SERVICE SPECIFICATION

I/O

· IR sensors placed at each parking lot and at entry and exit gates

· The parking system gives information about the vacant parking slots

Endpoints: Sensors(IR), Arduino, Database, Mobile application

Schedules

· Every time a car enters or exits the parking lot

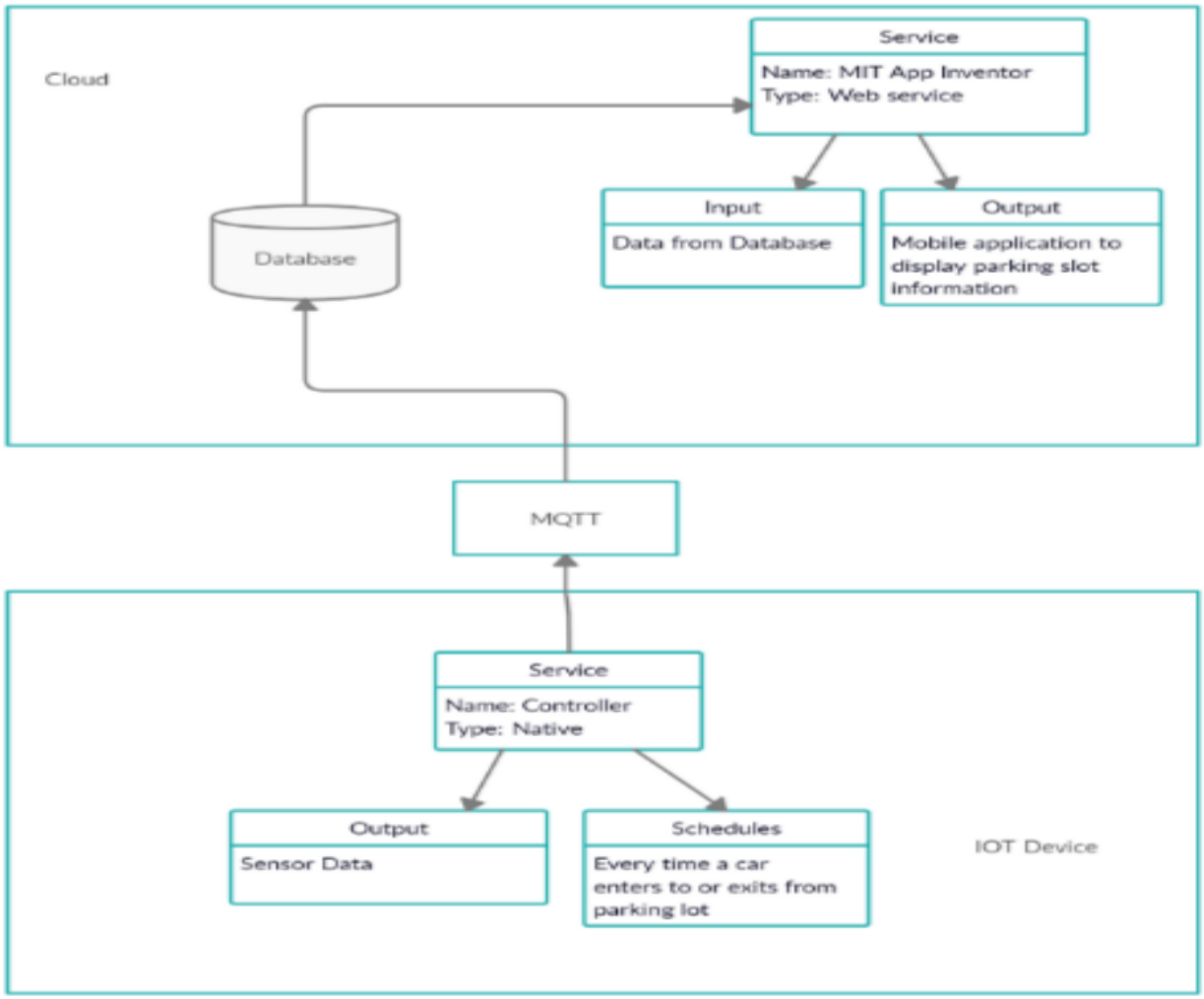
Preconditions

· User should be aware of the mobile application.

· User should have internet access to the mobile application.

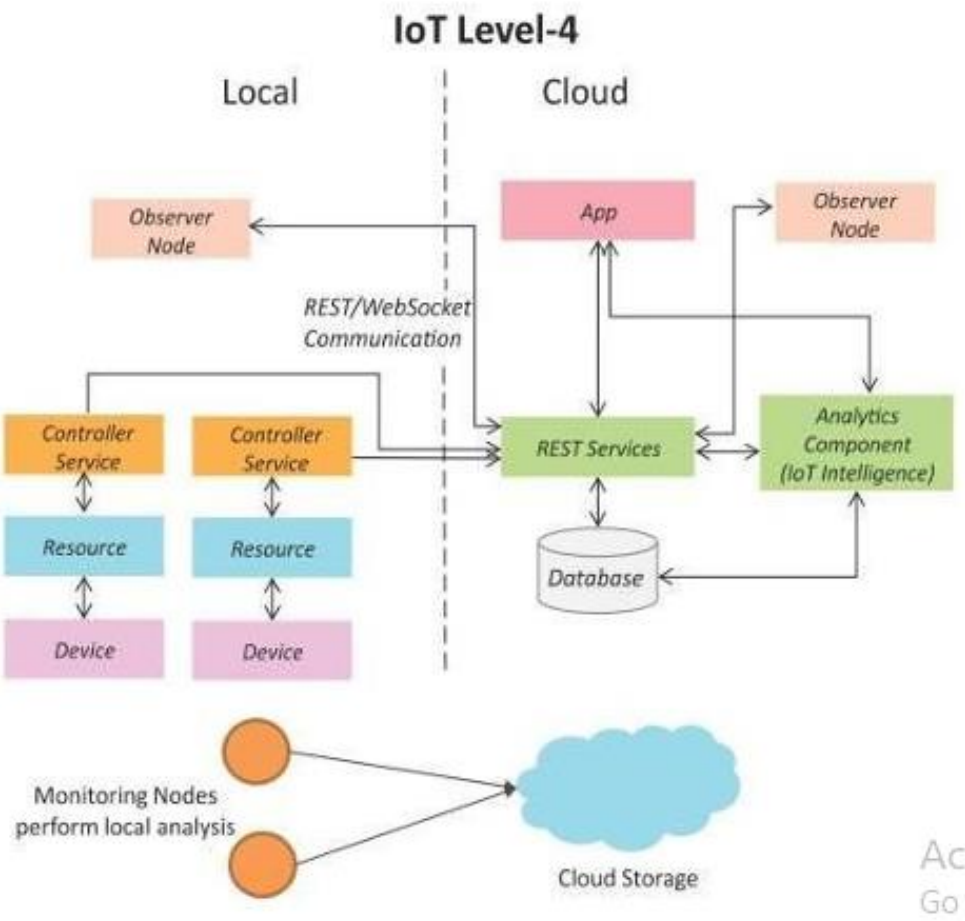
Effects

· User can view the vacant parking slots with slot number.



6. IOT LEVEL SPECIFICATIONS

Level 4 seems to be a good enough solution for applications such as this. IoT system-level-l is the example for modeling low complexity and low-cost solution where the analysis requirement is not comprehensive and the data involved is not big.



7. FUNCTIONAL VIEW SPECIFICATION

Here is a list of functional view specification of smart parking system

Devices: Actuator, Sensors, Computing Device

Communication:

· MQTT Broker — Send data from controller to the cloud

· Wi-Fi — To view data on a mobile application

Services: Native and web services

Management:

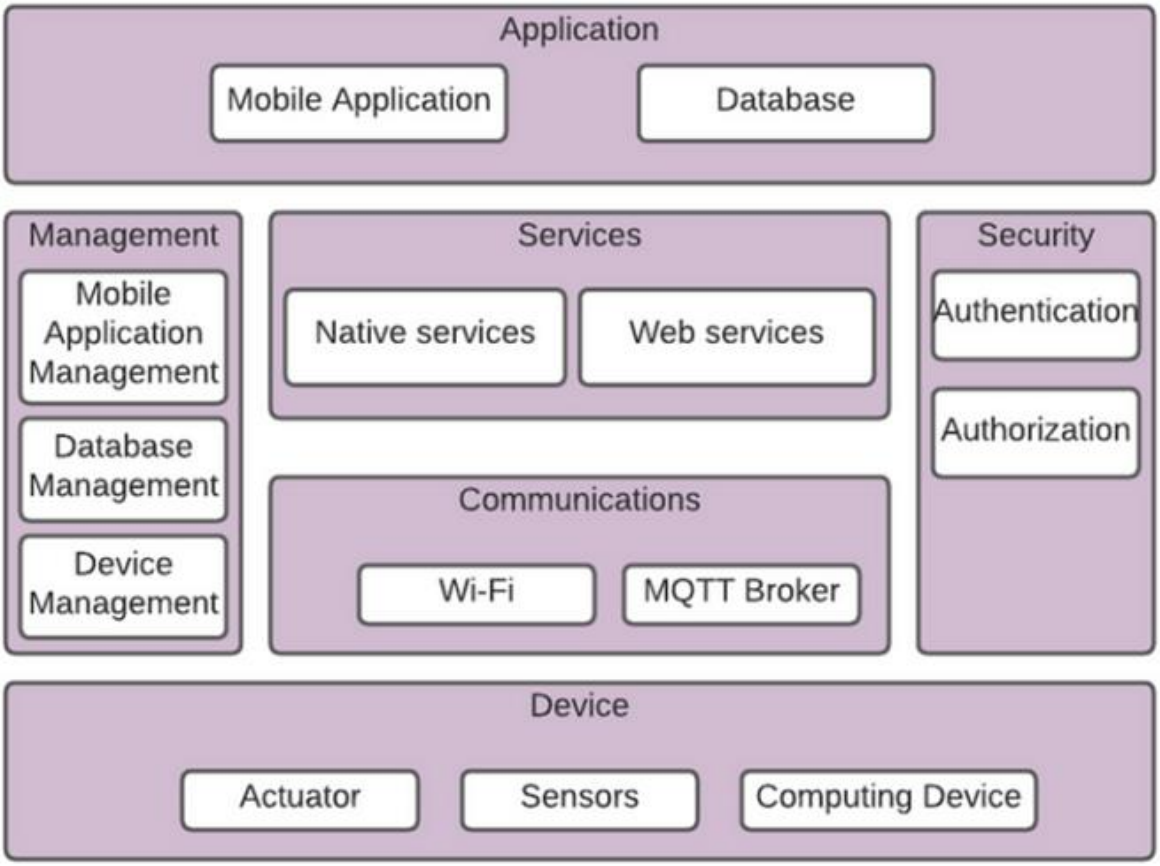
· Mobile App Management

· Database Management

· Device Management

Security: Authentication, Authorization

Application: Mobile Application and Database



8. OPERATIONAL VIEW SPECIFICATION

The given specifications are preferred choices for enhanced and efficient experience.

1. **Application Group:**

· Mobile Application: MIT Mobile App

· Database: Google Firebase

2. **Service Group:**

· Native Service: Controller Service

· Web Service: MIT App Inventor

3. **Management Group:**

· Application Management: MIT Management, Google play store, Appstore

· Database Management: Google

· Device Management: Arduino

4. **Security Group:**

· Authentication: Mobile App, Database

· Authorization: Mobile App, Database

5. **Communication Group:**

· Protocols: MQTT, Wi-Fi

6. **Device Group:**

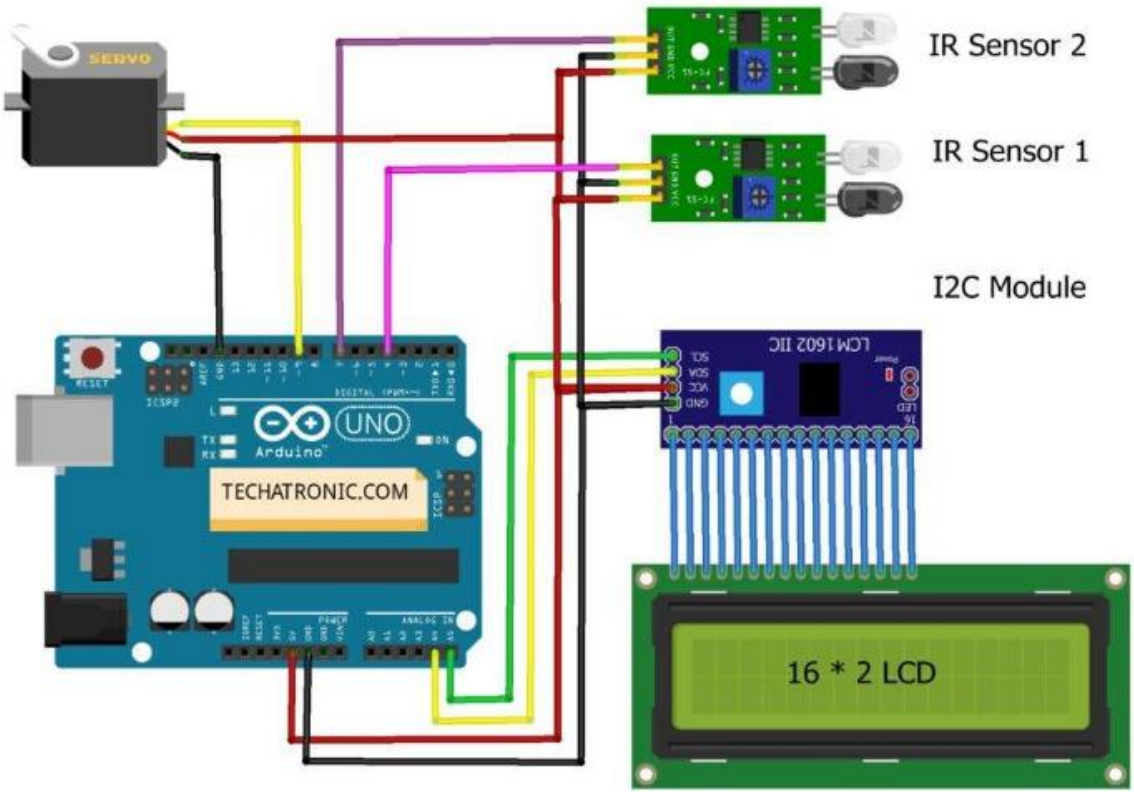
· Computing Device: NodeMCU

· Sensor: IR sensor

· Actuator: Servo motor

9. DESIGN AND COMPONENT INTEGRATION

A free area of land meant for parking purpose have 80 parking slots, 60 out of which are for cars and 20 for two wheelers. Each slot is allocated an unique number. The parking area has two separate gates for entry and exit of vehicles. An infrared sensor is connected to each gate which controls the activity of turnstile gate. When a vehicle comes near the gate to enter, the IR sensor at entry gate detects the vehicle. It then checks the system for free space in the parking area. If found, it allots the car with the slot number and marks the spot as occupied. It also opens the gate and lets the vehicle in. During the exit of the vehicle, IR sensors detect the vehicle and then the servo motor is used to turn the turnstile gate for the vehicle to move out. At the same time the website is updated and slot is made.



10. APPLICATION DEVELOPMENT

The Smart Parking System as a whole is built on three subsystems that run in parallel namely web applications, mobile applications, and embedded systems. The concept of IoT on the Smart Parking System is applied to the communication lane between the application and the embedded system which enables the embedded system to be controlled remotely through the application. Applications with different platforms require subsystems to communicate through networks with special protocols. The

lightweight communication protocol used in the Smart Parking System is MQTT. The web application has two execution environments namely API and the webserver.