## Background

Finding a parking spot may seem like a superficial problem or an individual problem, but is, in fact, one that causes significant issues in the day to day life of people. In metropolitan areas worldwide, having a parking area is increasingly important, and a substantial amount of land and buildings are set aside to accommodate the growing demand for parking spaces. For instance, in a developed country like Australia, parking areas hold great importance especially in major cities Brisbane, Sydney, and Melbourne, where a vast number of parking spaces can range from 25,633 to 41,687 [8]. Furthermore, estimates show that around 30,000 square kilometers of land in Europe and 27,000 square kilometers in the US are devoted to parking spaces [9].

Moreover, parking can be very time consuming and costly. Hence, it comes as no surprise that “The U.S. economy bears the brunt of parking pain as 40% of drivers say they have avoided driving to shops due to parking challenges.” [10]. According to a study from the car service company Inrix, Americans spend an average of 17 hours per year searching for a vacant parking spot, which leads to a loss of $345 per driver in wasted time and fuel. Similarly, drivers in the UK spend an average of 44 hours a year looking for an empty parking spot, with an estimated total loss of $954.

Contrary to the popular belief that generous parking allocations benefit users, the opposite can happen when there is a surplus of parking allocations. An enormous parking lot can prolong transportation times and is a waste of useable land. These effects, along with recent land-use, and socioeconomic and technological trends, are prompting towns to begin asking some critical questions about how to solve the parking problem smartly and cost-effectively. For this reason, there have been many attempts to provide smart parking systems as a solution, with each having its different approach as shown in the Related Work section.

## Related work

Several studies conducted and presented in previous literature related to our project have assisted us in the development and implementation of our design.

**A Secure Parking Reservation System Using GSM Technology**

In [11], GSM technology was used to propose a secure and smart reservation system for parking areas. The system consists of two major modules which are the security reservation module and parking lot monitoring.

Secure Reservation Module

The main idea is that the user has to send an SMS message using GSM (Global System for Mobile communications) to a Visual Basic (VB) application in a laptop/PC. The VB then processes the requested data and checks the available spot. The authors of the study use open APIs in VB to handle SMS messages. If the requested parking spot is available, the user will receive a confirmed message that has the location of the spot and password. Otherwise, the user will receive a reject message.

Parking Lot Monitoring Module

The primary aim of this module is to show the status of the parking spots using Parking Layout Animation program through Visual Basic application in a laptop/PC. The program displays the status of each parking spot through colors, green for *empty*, yellow for *reserved*, and red for *occupied*. Initially, the state of all parking spots is *empty*. The status will change to *reserved* if the user successfully reserves a parking spot. When the user arrives at the entrance gate, he enters the password, which after will be verified by the controller. If the password is valid, the gate will open, and the Parking Layout module will update the spot status to *occupied*.

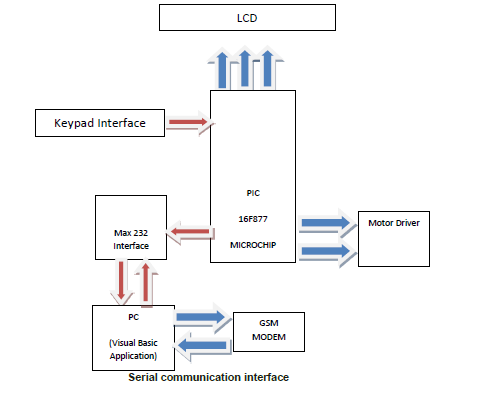
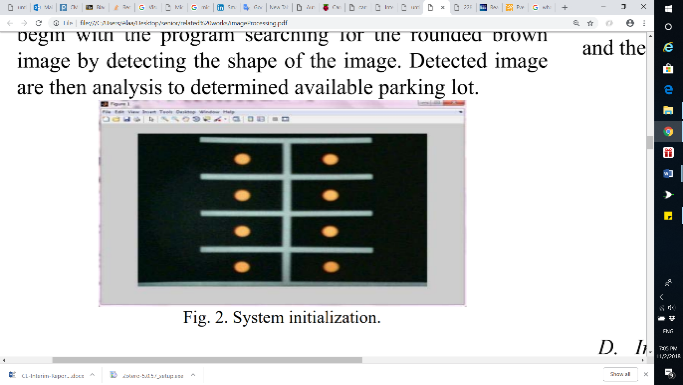
By using this approach, the user will be charged by the telecommunication company for sending the SMS messages. However, GSM is prone to be easily congested in crowded areas [12]. In this approach, all information is stored locally on one computer, which is inefficient as the computer memory will eventually be too full to save any more information from the system. Also, if the computer is somehow damaged, all the data will be lost.

Figure .: A Secure Parking Reservation System Using GSM Technology – System architecture

**Intelligent Parking Space Detection System Based on Image Processing**

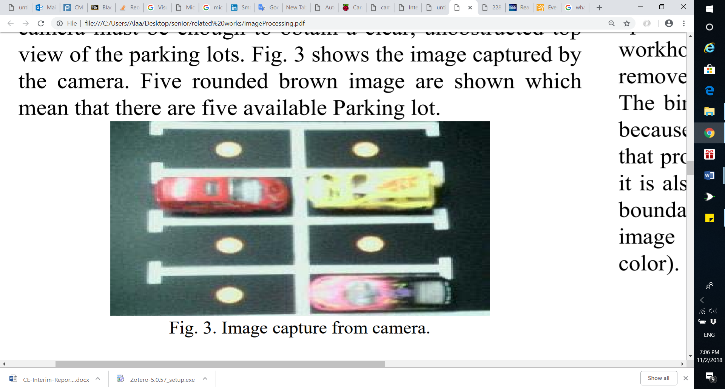
In [13], an image processing technique was used to detect whether the parking spot is empty. The proposed solution is carried out in 5 steps as follows:



1- System Initialization:

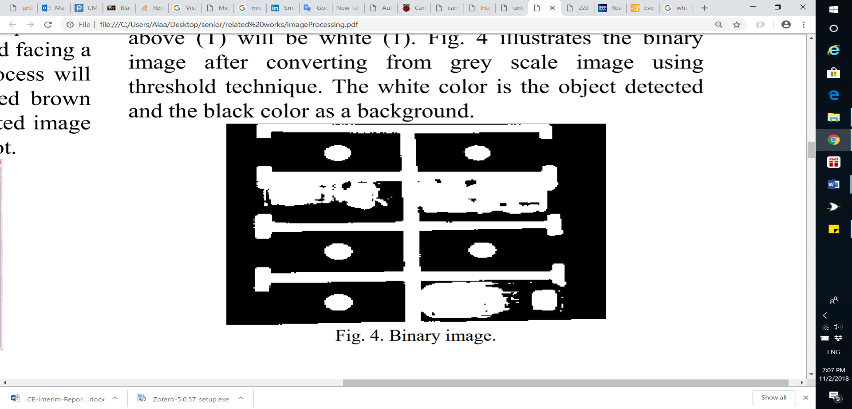
In the beginning, a rounded brown image is drawn at each parking spot manually to determine their positions.

Figure ‎2.2: System Initialization

2- Image Acquisition:

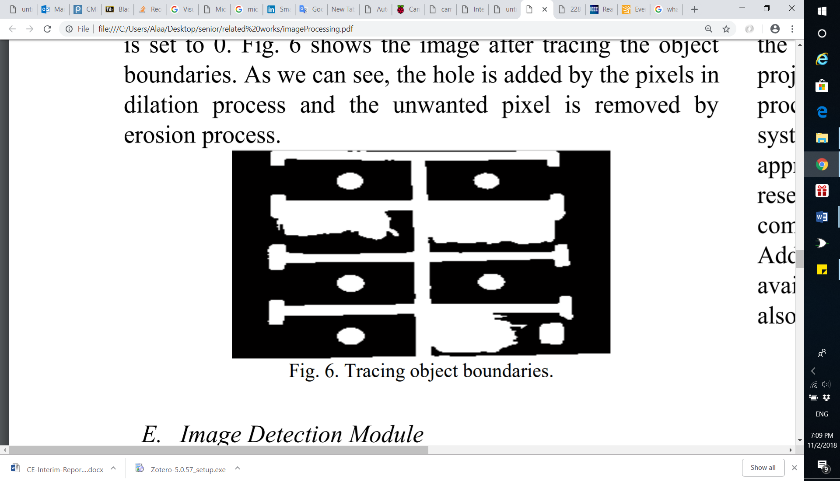
The camera captures an RGB image of the car park scene and sends it to the processing unit that runs in MATLAB.

Figure ‎2.3: Image capture from camera

3- Image Segmentation:

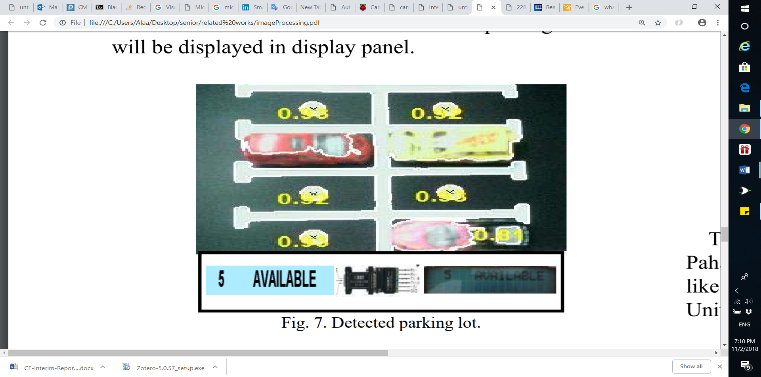
MATLAB converts the RGB Image to a grey scale image. Then thresholding technique on the grey scale image is applied to create a binary image. The technique helps in separating the objects from the background.

Figure ‎2.4: Binary image

4- Image Enhancement:

The morphology functions such as dilation, erosion, opening, and closing are used to remove noise from the binary image as well as to trace the exterior boundaries of the object.

Figure ‎2.5: Image after Tracing object boundaries

5- Image Detection Module:

The system detects if the parking spot is empty or not by identifying the rounded brown image at every spot.

Figure ‎2.6: Detected parking spot

This approach may not function well in extreme weather conditions, as the camera might be damaged. Also, placing the camera in a suitable location such that all parking areas are visible, and there are no objects that might obstruct the camera's vision is essential for the camera to work as planned.

**Monitoring Parking Space Availability via Zigbee Technology**

In [14], the system’s primary function is to allow its users to view the available parking spots through a screen at the gates of a parking area using Zigbee wireless technology for communication and digital infrared sensors for vehicle detection.

The system consists of two main modules:

Parking Lot Vacancy Monitoring Module

This module comprises of digital infrared sensors, Zigbee module, PIC microcontroller 18F4550, and an LCD. This module is used to detect the available parking spots. A digital infrared sensor is placed on top of each parking spot to detect the existence of a vehicle such that each sensor is used to monitor one parking spot. Infrared sensors work with the reflected light waves and consist of an IR transmitter and an IR receiver. The IR transmitter emits infrared light, which gets reflected when it meets a reflecting surface (e.g., white color). The IR receiver then detects the reflected light and calculates the distance between the sensor and the object. The infrared sensor interfaces with a PIC microcontroller, which also interfaces with a Zigbee module for wireless transmission. When the sensor senses a vehicle in the parking spot, it informs the microcontroller, and then the microcontroller notifies the master module through the Zigbee module.

Master Module

The master module consists of a GUI display shown through a PC or a laptop, and a Zigbee module. The GUI display, shown in Figure 2.7, allows the user to see precisely which parking spots are available. Whenever the Zigbee module receives data about a particular parking spot from the Parking Lot Vacancy Monitoring module, it informs the PC/laptop that it is interfaced with to update the status of a parking spot.

Nevertheless, this system has several disadvantages. The infrared sensor is entirely dependent on light as brighter surfaces are more accessible for detection than darker surfaces. Hence, changing light conditions could yield wrong outputs. Also, the system does not have a mobile application for a remote user interface and restrict its users to a GUI interface at the gates of the parking lot.

Figure 2.8 shows the hardware architecture of the system.

Figure .: GUI Display on PC or laptop

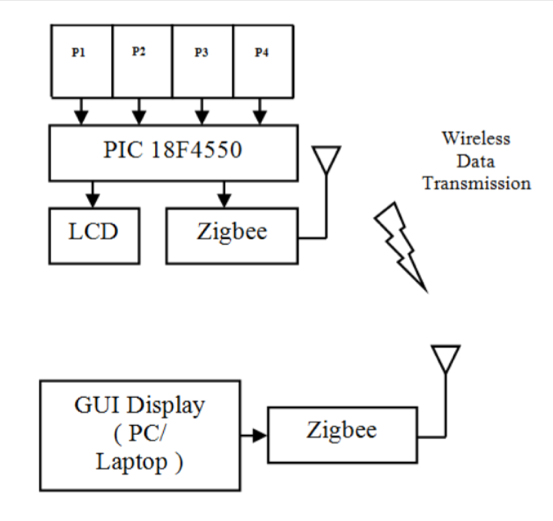


Figure .: Hardware architecture of the system

**Smart Parking System (SPS) Architecture Using Ultrasonic Detector**

[15] is a smart parking system for multilevel parking lots, which uses ultrasonic sensors to detect the occupancy of a parking space and shows it to the user at the parking area using display boards. The display boards show the number of available parking spots, and they are implemented indoors, at the entrance to each level and the end of each aisle of the parking lot, and outdoors, at the entry and exit of the parking lot. Furthermore, an ultrasonic sensor is installed on the ceiling of each parking spot and is used to detect the presence of a car in that spot only. LEDs with different colors accompany each ultrasonic sensor to inform the user of the parking spot status (reserved, occupied, vacant or handicapped). Moreover, the system includes a monitoring software and line detection system to detect improper parking. The line detection system operates by two additional ultrasonic sensors horizontally on the right and left of each parking spot such that they face the parking spot lines. Whenever a car goes over the detection line, an alarm will go off until the vehicle has moved out of the line.

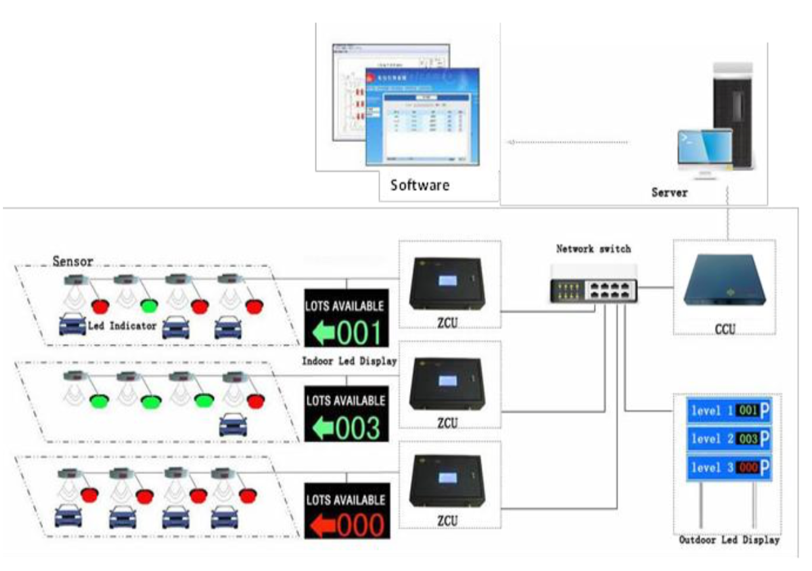


Figure ‎2.9: Hardware architecture of the system

Figure 2.9 shows the overall hardware architecture of the system. The system consists of ultrasonic sensors, LED indicators, indoor display boards, outdoor display boards, zone control units (ZCU), a central control unit (CCU), network switch, telephone cable, and management software. When the ultrasonic sensor detects a change in the status of a parking area, it transmits the status to the zone control unit (ZCU) through a telephone cable. The ZCU then forwards the information to the central control unit (CCU) with Cat5 cables and sends commands to the indoor display boards and the LEDs on the parking sports to change status accordingly. Both the indoor display boards and the LEDs connect to the ZCU through RS-485 ports. The ZCU connects to the central control unit (CCU) through network switches and LAN connections. The CCU upon receiving new information from any of the ZCUs processes the data with the whole parking lot data, then transmits commands to the outdoor display boards to update the parking area information.

The system is able to efficiently manage multilevel parking areas. However, the system contains many hardware components and connects its components through a great number of cables. Any extension to the system would require more cables and more complications on the system.

**Intelligent Parking System**

In [16], the system uses image processing techniques which capture, and process circles or rounded images drawn from the parking spot, which are then displayed in an android application. This rounded image indicates empty parking spaces to users.

A camera is used with a sensor to take photos to show the occupancy of car parks. The application displays whether a parking space is available or not. If the camera detects the vacancy of a parking area, a green circle appears to the user. If there is a car in a spot, it becomes unavailable, and no green circle is shown at that spot in the application.

The system contains three modules: A Monitoring module, Control module, and a Displaying unit. The monitoring module has ultrasonic sensors which identify the free parking spaces and transmit information to the control unit which processes and sends data to the administrative system. Besides that, there is a centralized system that supervises the received parking information from the controller. This information is then sent to the user’s phone.

Below (Figure 2.10) illustrate the working process of the system.

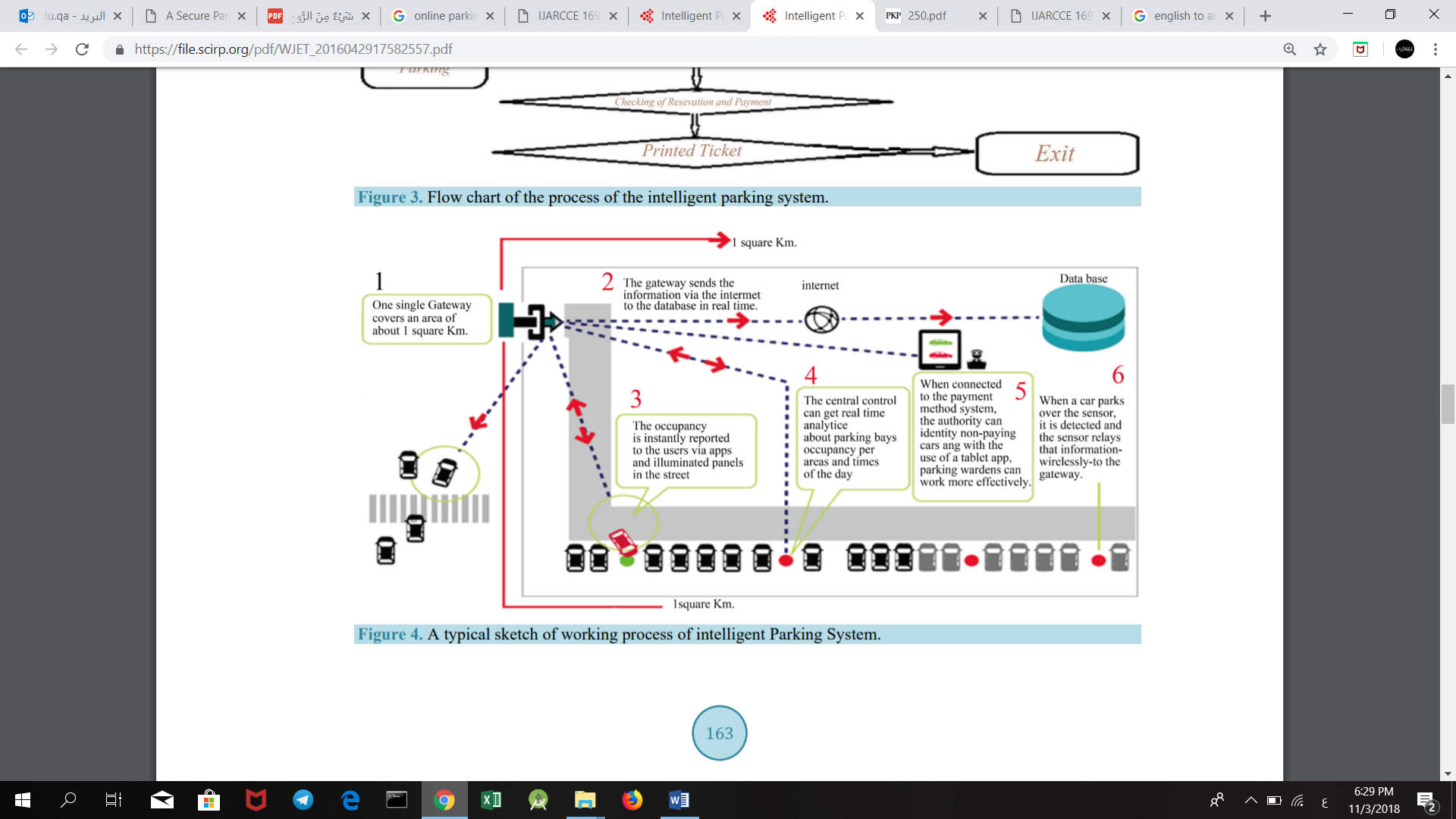


Figure .10: Working process of the system

Related work comparison

Table ‎2.1: Related work comparison

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Paper** | **Hardware Used** | | | **Service** | | | **User Interface Through** | **Connection** |
| **Microcontroller** | **Sensors** | **Others** | **Reservation** | **Availability checking** | **Payment** |
| **[11]** | Pic Microcontroller |  | LCD,  Motor driver |  |  |  | Visual Basic Application | GSM |
| **[13]** |  |  | Camera |  |  |  |  |  |
| **[14]** | Pic Microcontroller | Infrared Sensor | Zigbee- module  LCD |  |  |  | GUI displayed on PC or laptop | Wireless |
| **[15]** |  | Ultrasonic Sensor | Display  Boards,  LEDs,  Network Switch |  |  |  |  | Telephone cables, LAN connection |
| **[16]** |  | Ultrasonic sensors | Camera |  |  |  | Android Mobile Application |  |
| **Our Method** | Arduino Uno | Ultrasonic  RFID | Servo  motor,  LEDs |  |  |  | Mobile- Application  Website | Wireless |

Table 2.1 compares the existing solutions to ParQU. All the related works implement one of the following services: Checking Availability, Reservations, Payment except [16]. [16] gives all the above functions that we aim to provide in our system. However, [16] only implements an Android application while we will provide a website in addition to that. Furthermore, the above approaches use ultrasonic, infrared and camera for car detection only. Our method uses an ultrasonic sensor for car detection and RFID (Radio Frequency Identification) for car identification and detection.

# Requirements analysis

In this section, the software development process used in this project is defined. The functional and nonfunctional requirements of the system are addressed.

## Software development process

The software development process in our project is Scrum. Scrum is an iterative and incremental agile software development process that focuses on adapting to rapid changes. Scrum was chosen for our project because it does not require us to know precisely how and what to do from the beginning of the development process. The Scrum process is responsive to changes and improvements with repeated reviews and work inspection [17]. Figure 3.1 shows the Scrum process overview.

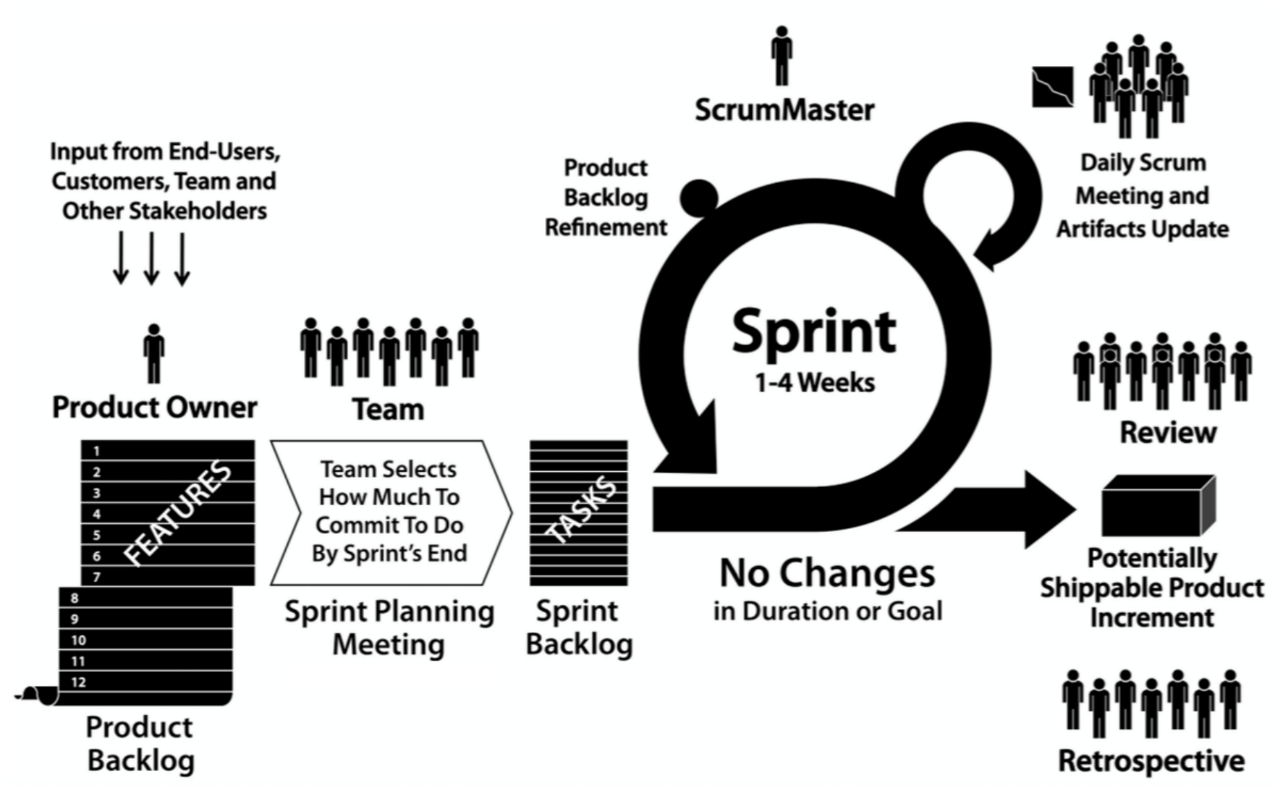


Figure 3.: Scrum process overview

The central part of the Scrum cycle is the Sprint. Each Sprint is time restricted to one week and has four steps as shown in Figure 3.2.

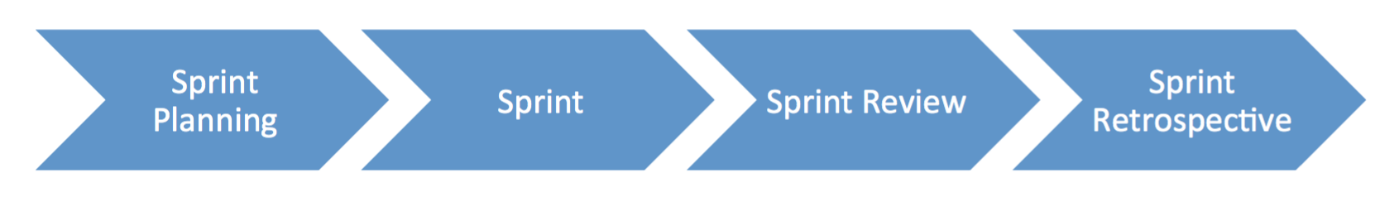


Figure 3.: Scrum cycle

**1. Sprint Planning**

At the beginning of the project, we collected the requirements into a Product Backlog from the users by researching the parking problem in Qatar University and the users’ needs. Before starting a Sprint, we held up a meeting to decide on the top priority requirements from the Product Backlog and created a Sprint Backlog. The Sprint Backlog contained the tasks that we could complete during the Sprint which in our case was one week.

**2. Sprint**

In this stage, we started working on the tasks as specified in the Sprint Backlog. Further, a Daily Scrum Meeting was held to discuss the tasks that were done by each member so far, and the tasks for the next meeting. Also, if any member has faced any difficulties during the task, we tried to find a solution for it.

**3. Sprint Review**

At the end of a Sprint, a meeting is held with the ScrumMaster (supervisor) to demonstrate our work. The ScrumMaster suggested solutions to the issues we faced during the Sprint.

**Sprint Retrospective**

After the Sprint Review, we refined our tasks to meet the comments and suggestions given by the ScrumMaster and decided on the improvements for the next Sprint.

## Functional requirements

* An RFID reader should be installed and kept operational at the gate of the parking facility.
* VIP users must have an RFID tag in their possession before entering the parking facility.
* Sensors should be hardwired to Arduino.
* Arduino must be connected to a power source.
* Arduino must be connected to a Wi-Fi module.
* Firebase must have a connection with the Arduinos, the mobile application and the website via Wi-Fi.
* The mobile must have internet connection.
* The user should be a University member or visitor.

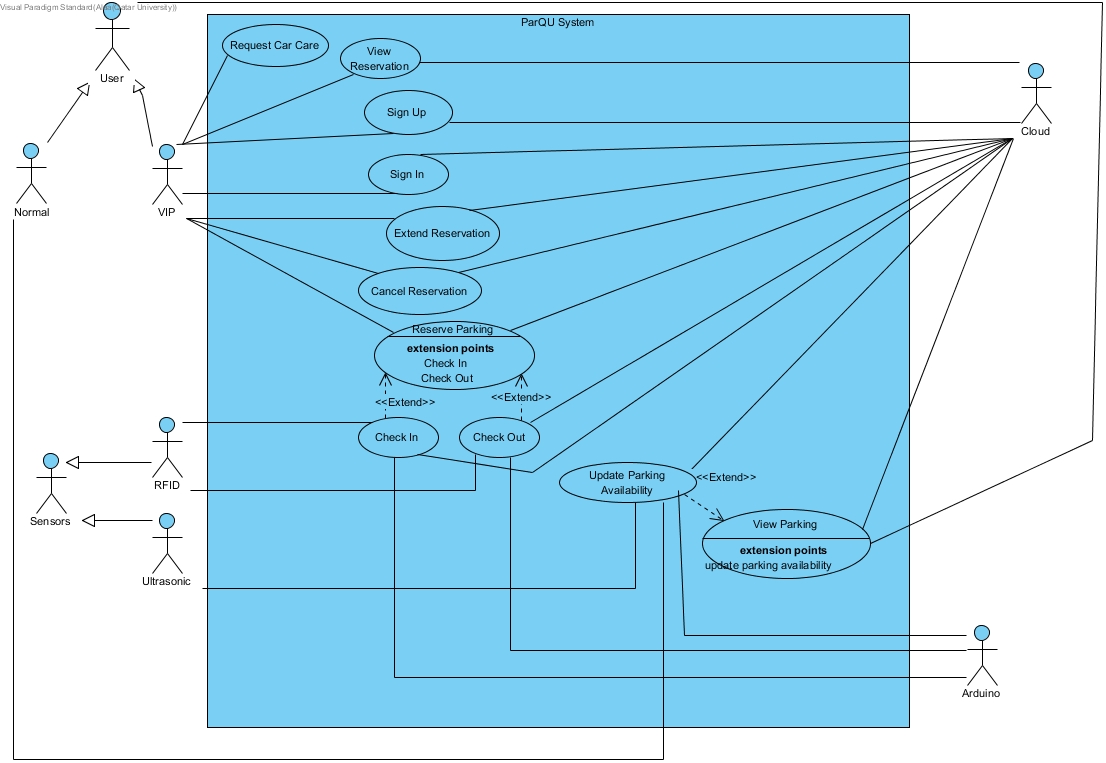


Figure 3.: Use case diagram

Table ‎3.1: Actor type and descriptions

|  |  |
| --- | --- |
| Actor types | Description |
| VIP User | Anyone subscribed to the VIP services such as reservation service and requests car care service. |
| Normal User | Anyone who is not subscribed to the VIP services. |

Table ‎3.2: Use cases and descriptions

|  |  |
| --- | --- |
| Use case | Brief description |
| Sign Up | VIP user registers in the system. |
| Sign In | VIP user logs into the system. |
| Reserve Parking | VIP user reserves a parking spot for several hours. |
| View Reservation | VIP user views reservation details. |
| Extend Reservation | The VIP user extends a reservation. |
| Cancel Reservation | The VIP user cancels a reservation. |
| Check In | The system reads the RFID tag on the car, checks its validity, then allows reserved VIP users only to enter the parking lot. |
| Check Out | The system reads the RFID tag on the car, checks its validity, then update available parking spots. |
| View Parking | The user views a map with the current status of parking spots. |
| Update Parking Availability | The system receives updated data from sensors through the Arduino board. |
| Request Car Care | The VIP user requests car care service which will display contact numbers of car care company. |

## Non-functional requirements

**Design constraints**

Table ‎3.3: Technical design constraints

|  |  |
| --- | --- |
| Name | Description |
| Reliability | Data is saved at all times on the Firebase database, so no data will be lost. |
| Scalability | The system can support the addition of sensors and components as well as have more users and the cloud can be upgraded accordingly. |
| Connectivity | Firebase database needs to be connected to the internet to collect updated data from the sensors through the Arduino board. Also, the mobile application needs to be connected to the internet to get the data |
| Availability | The system should be always available. During using the application, if any data is changed in the firebase database, the application should automatically display the updates without needing to refresh the page. |
| Mobility | The system can be accessed from two different platforms: an Android application and a website |
| Power | Power source needed for:  Motor: 3-7V  Sensor: 5V  Arduino: 7-12V |

Table ‎3.4: Practical design constraints

|  |  |  |
| --- | --- | --- |
| Type | Name | Description |
| Economics | Cost | The prototype for availability module which consists of 4 parking spots should cost 270 QR on average.  The prototype for reservation module should cost 260 QR on average. |
| Social | Usability | A normal user with minimal software knowledge should be able to use the mobile application and the website with ease |
| Sustainability | Maintenance | The system’s component should be easy to replace, remove and implement. |
| Quality | Performance | The system should provide efficient information and accurate readings from the parking area. |

**Design standards**

Table ‎3.5: Standard types and descriptions

|  |  |
| --- | --- |
| Standard Types | Description |
| Networking Standard | HTTP (Hypertext transfer protocol) |
| Security Standard | OpenID Connect (used for delegated authentication) |
| Wireless Standard | IEEE 802.11 (Wi-Fi) |
| Serial Communication Standards | SPI (Serial Peripheral Interface) |

## Assumptions

Below are some assumptions that are beyond the scope of this project but necessary for the project to work properly:

* Arduino is always connected to the sensors.
* A power source is connected to the Arduino.
* Arduino always has a Wi-Fi connection.
* Firebase can always store data without limits.
* Application and website are always available.
* Mobile and PC that access our application and website is always connected to the internet.
* VIP user always has an RFID tag.

## Ethics

Table ‎3.6: IEEE and ACM code of ethics

|  |  |
| --- | --- |
| Code | Project Perspective |
| IEEE  To seek, accept, and offer honest criticism of technical work, to acknowledge and correct errors, and to credit properly the contributions of others.  ACM  - Honor property rights including copyrights and patent.  - Give proper credit for intellectual property. | Any project or work used is referenced in this paper. Feedback will be accepted by the supervisor and the examiners. |
| IEEE  To treat fairly all persons and to engage in acts of discrimination based on race, religion, gender, disability, age, national origin, sexual orientation, gender identify, or gender expression.  ACM  - Be fair and take action not to discriminate. | Project members have all worked together to produce this work which can be used by all types of people. |
| IEEE  To assist colleagues and co-workers in their professional development and to support them in following this code of ethics. | The project is collaborative; every team member will work on it and ensure the code of ethics is strictly followed. |
| ACM  - Contribute to society and human well-being.  - Manage personnel and resources to design and build information systems that enhance the quality of working life. | The design aims to achieve better standards and benefits. |

## 