|  |
| --- |
| *Project Name* |
| **System Requirements Specification** |
| **SE 6387 Advanced Software Engineering Project**  **R.Z. Wenkstern**    ***Date*** |

|  |
| --- |
| **Group *X*** |
|  |
|  |
|  |

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Decription** | **Authors** |
| 1.0 |  | Completed initial draft |  |

Contents

[Revision History 2](#_Toc379403893)

[1. Introduction 1](#_Toc379403894)

[1.1. Purpose 1](#_Toc379403895)

[1.2. Scope 1](#_Toc379403896)

[1.3. Overview 1](#_Toc379403897)

[2. Overall Description 1](#_Toc379403898)

[2.1. Product Perspective 1](#_Toc379403899)

[2.2. Product Functions 1](#_Toc379403900)

[2.3. User Characteristics 1](#_Toc379403901)

[2.4. Constraints 1](#_Toc379403902)

[2.5. Assumptions and Dependencies 1](#_Toc379403903)

[3. Hardware Specification 1](#_Toc379403904)

[3.1. Hardware Component 1 1](#_Toc379403905)

[3.1.1. Functionality 1](#_Toc379403906)

[3.1.2. Operational Requirements 1](#_Toc379403907)

[3.1.3. QoS Requirements 1](#_Toc379403908)

[3.1.4. Parametric Requirements 1](#_Toc379403909)

[3.1.5. Design Requirements 1](#_Toc379403910)

[3.2. Hardware Component 2 2](#_Toc379403911)

[4. External Interface Requirements 2](#_Toc379403912)

[4. 1. User Interfaces 2](#_Toc379403913)

[4. 2. Hardware Interfaces 2](#_Toc379403914)

[4. 3. Software Interfaces 2](#_Toc379403915)

[4. 4. Communication Protocols and Interfaces 2](#_Toc379403916)

[5. System Features 2](#_Toc379403917)

[5.1. System Feature A 2](#_Toc379403918)

[5.1.1. Description 2](#_Toc379403919)

[5.1.2. Action/result 2](#_Toc379403920)

[5.1.3. Functional Requirements 2](#_Toc379403921)

[5.1.4. NFR 2](#_Toc379403922)

[5.2. System feature B 2](#_Toc379403923)

[6. Non Functional Requirements 2](#_Toc379403924)

[6.1. Product NFR 2](#_Toc379403925)

[6.2. Process NFR 2](#_Toc379403926)

[6.3. External NFR 2](#_Toc379403927)

[7. Other Requirements 3](#_Toc379403928)

[Appendix A: Glossary 4](#_Toc379403929)

[Appendix B: References 5](#_Toc379403930)

# 1. Introduction

## 1.1. Purpose

## 1.2. Scope

## 1.3. Overview

# 

# 2. Overall Description

## 2.1. Product Perspective

## 2.2. Product Functions

## 2.3. User Characteristics

## 2.4. Constraints

## 2.5. Assumptions and Dependencies

# 3. Hardware Specification

## 3.1. Hardware Component 1

### 3.1.1. Functionality

### 3.1.2. Operational Requirements

### 3.1.3. QoS Requirements

### 3.1.4. Parametric Requirements

### 3.1.5. Design Requirements

## 3.2. Hardware Component 2

# 4. External Interface Requirements

## 4. 1. User Interfaces

The CometPark system will be presented as a mobile application to the users and it will be available as a web interface for the administrator.

The layout of the screens of the mobile application and the web interface will be determined in the later stages as the project progresses.

## 4. 2. Hardware Interfaces

The CometPark system has a number of hardware components that will interact with each other to make the parking space information available to the user. The following are the hardware interfaces between the various components of the system.

1. XBee Explorer Dongle – To interface the XBee module with the Raspberry Controller

2. Wifi Adapter – For wireless network connection between the Controller and the Cloud server

3. Resistors to manage voltage differences between the sensors and power source.

## 4. 3. Software Interfaces

The user will be able to access the CometPark system via a mobile interface and the administrator will access the system through a web interface.

The system will use REST Web service to process the user request to get the vacant parking spot information.

## 4. 4. Communication Protocols and Interfaces

The communication between the sensors and the Raspberry Pi controller is through XBee module which uses the IEEE 802.15.4 networking protocol and operates on ISM 2.4 GHz.

The Raspberry Pi controller communicates with the cloud server through Wi-Fi. The Wi-Fi adapter provides this wireless capability to the controller. The Wi-Fi adapter uses the 802.11n standard and operates on 2.4GHz.

# 5. System Features

## 5.1. System Feature A

Locate the vacant parking spots that are nearest to the user.

### 5.1.1. Description

The primary goal of this project is to help the users locate a vacant spot close to them to park their vehicle. When the user arrives at a location and refreshes the application, the application shall display the unoccupied spots in the parking lots close to him/her.

### 5.1.2. Action/result

When the user is near a building in the campus and wants to park his/her vehicle in a nearby parking lot, refreshing the mobile application will display the vacant parking spot that is nearest to him/her.

This feature will save a lot of time and fuel to the users who commute to the UTD campus via car. The application will prevent frustration among the students and staff in trying to find a vacant spot during peak hours.

### 5.1.3. Functional Requirements

FR 1.1 -The CometPark system shall find the vacant parking spots that are of shortest distance from the user’s location.

FR 1.2 - The system shall be available to the user as a mobile application.

FR 1.3– The system shall be able to determine the status of the individual parking spots.

FR 1.4 – The system will allow the user to choose the number of nearest vacant spots (between 1 and 5) to be displayed to him/her.

FR 1.5 – The system shall identify whether or not there are any vacant parking spots in the lot.

### 5.1.4. NFR

NFR 1.1 – Adaptability – The system shall allow the users to set the number of spots to be displayed according to his/her needs.

## 5.2. System feature B

Provide options to the user to find parking spots with the preferred color code.

### 5.2.1. Description

According to the current Parking system at UTD, the users purchase parking permit of a particular color and must park their vehicles only in the spots allocated for that color. To meet this requirement, the application will provide the user with the option to set their preferred color and view only the vacant spots associated with that color.

The following are the options that will be available for the user to choose from:

1. Green

2. Gold

3. Orange

4. Evening Orange

5. Purple

6. Handicap

According to the color code chosen, the available parking spaces in the lots will be determined according to the following table specified by the UTD Parking Department.

|  |  |
| --- | --- |
| **Permit Type** | **Permissible Parking Areas** |
| Green | Green and Extended parking areas |
| Gold | Gold, green and Extended parking areas |
| Evening Orange | Orange after 5.P.M ONLY- gold, green and extended parking areas anytime |
| Orange | Orange, Gold and Green and Extended parking areas |
| Purple | Purple, Orange, Gold, Green and Extended parking areas |
| Handicap | Areas designated for Handicap parking |

### 5.2.2. Action/result

The user can set the color code of their choice from a menu option that the application will provide. The application will locate the nearest available parking spaces that are allocated for the color set by the user.

This feature will allow the users to see only the vacant spots that correspond to the user’s parking permit.

### 5.2.3. Functional Requirements

FR 2.1-The application shall provide the user with the option to set the color code of their permit.

FR 2.2- The application shall display only the vacant parking spaces associated with the color chosen by the user if the user sets a color.

### 5.2.4. NFR

NFR 2.1 – Adaptability – The system shall allow the users to set their preferred color parking spaces that are to be searched for.

NFR 2.2 – Extensibility – The system shall allow the addition of new color codes for the parking spaces.

## 5.3. System feature C

Manage the closure of some of the parking lots in case of events and occasions.

### 5.3.1. Description

In the event of a special occasion, the Parking Management at UTD closes some of the parking lots to the staff and students. During such an event, the application must not display the spots in those lots even though they are unoccupied.

### 5.3.2. Action/result

The administrator of the application marks the parking lots that are closed to the staff and students . As a result of this feature, when the user searches for a parking spot, the application will not display the spaces in the lots that are marked as closed.

### 5.3.3. Functional Requirements

FR 3.1 – The system shall authenticate the administrator by validating the user credentials.

FR 3.2 – The system shall manage the partial closure of parking lots by not displaying the spaces in the lots indicated as closed by the administrator.

### 5.3.4. NFR

NFR 3.1 –Configurability– The system shall be configurable by allowing the administrator to handle the closure status of parking lots.

## 5.4. System feature D

Manage the closure of all parking lots in case of emergencies.

### 5.4.1. Description

In the event of an emergency, the application will display an alert message on its home page informing the closure of all the parking lots in the campus.

### 5.4.2. Action/result

The administrator of the application sets all the parking lots as closed and sets an emergency alert message to be displayed on the application’s homepage.

### 5.4.3. Functional Requirements

FR 5.1 –The application shall inform the users of the closure of all parking lots in the event of an emergency.

# 6. Non Functional Requirements

## 6.1. Product NFR

## 6.2. Process NFR

## 6.3. External NFR

# 7. Other Requirements

# Appendix A: Glossary

|  |  |
| --- | --- |
| **Term** | **Definition** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

# Appendix B: References