|  |  |
| --- | --- |
| **­** | **MINISTRY OF EDUCATION AND TRAINING** |

**FPT UNIVERSITY**

|  |  |
| --- | --- |
| **Report 4** | |
| **Parking Guidance System Solution** | |
| **Group 1** | |
| **Group members** | Trần Nguyễn Minh Trung – Team Leader – SE61496  Bùi Phú Hiệp – Team Member – SE61438  Nguyễn Đỗ Phương Huy – Team Member – SE61358 |
| **Supervisor** | Nguyễn Đức Lợi |
| **Ext. Supervisor** | N/A |
| **Capstone Project Code** | PGSS |

- Ho Chi Minh City, Jan, 2017

*This page is intentionally left blank*

# Table of Contents

[Table of Contents 1](#_Toc475972823)

[List of Tables 2](#_Toc475972824)

[List of Figures 2](#_Toc475972825)

[Definitions, Acronyms and Abbreviations 2](#_Toc475972826)

[D. Software – Hardware Design Description 3](#_Toc475972827)

[1. Design Overview 3](#_Toc475972828)

[2. System Architectural Design 3](#_Toc475972829)

[2.1. Hardware Program Architecture Description 3](#_Toc475972830)

[2.2. Desktop Application Architecture Description 3](#_Toc475972831)

[3. Component Diagram 3](#_Toc475972832)

[4. Detailed Description 3](#_Toc475972833)

[4.1. Class Diagram 3](#_Toc475972834)

[4.1.1. Mobile App Class Diagram 3](#_Toc475972835)

[4.1.2. Web API Class Diagram 3](#_Toc475972836)

[4.1.3. MCU Class Diagram 3](#_Toc475972837)

[4.1.4. Parking Lot Class Diagram 3](#_Toc475972838)

[4.1.5. Information LED Display Class Diagram 3](#_Toc475972839)

[4.2. Class Diagram Explanation 3](#_Toc475972840)

[4.3. Interaction Diagram 3](#_Toc475972841)

[4.3.1. Flowchart Diagram (cho phần cứng) 3](#_Toc475972842)

[4.3.2. Activity Diagram (mobile app) 3](#_Toc475972843)

[4.3.3. Sequence Diagram (web api) 3](#_Toc475972844)

[5. Interface 3](#_Toc475972845)

[5.1. Component Interface 3](#_Toc475972846)

[5.2. User Interface Design 3](#_Toc475972847)

[6. Database Design 3](#_Toc475972848)

[6.1. Logical Diagram 3](#_Toc475972849)

[6.2. Data Dictionary 3](#_Toc475972850)

[7. Algorithms 3](#_Toc475972851)

# List of Tables

[Table 1: Definitions, Acronyms and Abbreviations 3](#_Toc475466552)

[Table 2: Raspberry Pi 3 - Specification 6](#_Toc475466553)

[Table 3: Arduino Nano - Specification 8](#_Toc475466554)

[Table 4: RF module nRF24L01+ - Specification 10](#_Toc475466555)

# List of Figures

[Figure 1: PGSS Block Diagram 5](#_Toc475466537)

[Figure 2: Raspberry Pi 3 6](#_Toc475466538)

[Figure 3: Arduino Nano 7](#_Toc475466539)

[Figure 4: Compass Module 3-Axis HMC5883L 8](#_Toc475466540)

[Figure 5: RF module nRF24L01+ 9](#_Toc475466541)

[Figure 6: RF module nRF24L01+ - Specification 10](#_Toc475466542)

[Figure 7: 7-segment LED Display 11](file:///C:\Users\TrungTNM\Documents\-FPT-CAPSTONE-PGSS\Common\Reports\Report%203.docx#_Toc475466543)

[Figure 8: TPIC6B595 Power Logic 8-Bit Shift Register 12](#_Toc475466544)

[Figure 9: TPIC6B595 Pin outs 13](#_Toc475466545)

[Figure 10: RGB LED common anode 14](#_Toc475466546)

[Figure 11: RGB LED common anode pin-out 15](#_Toc475466547)

[Figure 12: TIP122 Transistor 16](#_Toc475466548)

[Figure 13: Overview use case diagram 17](#_Toc475466549)

[Figure 14: Manager Use case diagram 18](#_Toc475466550)

[Figure 15: Conceptual Diagram 27](#_Toc475466551)

# Definitions, Acronyms and Abbreviations

|  |  |
| --- | --- |
| **Name** | **Definition** |
| PGS | Parking Guidance System |
| Parking area | An area set aside for parking vehicles, aircraft, etc. |
| Parking lot | A place inside parking area that provide space for one vehicle |
| IoT | Internet of Things |
| CCU | Central Control Unit |

Table : Definitions, Acronyms and Abbreviations

# D. Software – Hardware Design Description

## Introduction

### Overview

This section has all necessary information about test plan, test case and its result, the environment for testing and test pass/fail criteria

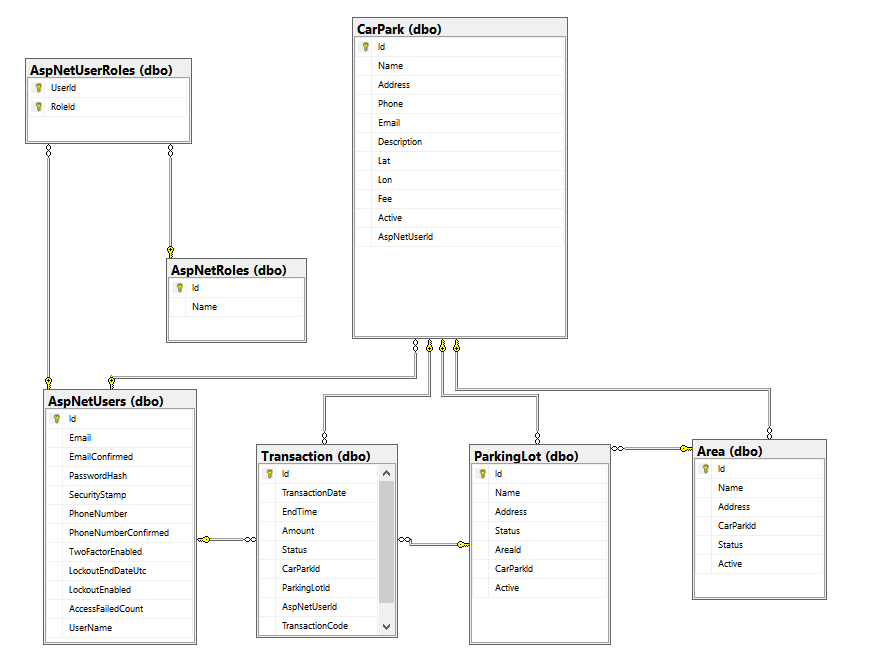
### Test Approach

White-box: Developers self-test on code in which function they developed

In this section, Black-box testing will be used to test whether the whole system meets the following objectives

## Database Relationship Diagram

### Physical Diagram



### Data Dictionary

|  |  |
| --- | --- |
| **Table Name** | **Description** |
| Area | Contain the area information of the car park |
| ParkingLot | Contain the information of each lot in the car park |
| Transaction | Contain the transaction information |
| CarPark | Contain the information of each car park |
| AspNetRoles | Contain the user’s role name |
| AspNetUserRoles | The mapping between 2 tables: AspNetRoles and AspNetUsers |
| AspNetUsers | Contain the user’s information |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table Name** | **Column Name** | **Description** | **Is Nullable** | **Data Type** |
| **Area** | Id | Unique Id for each area | NO | int |
| Name | Name of area | NO | nvarchar |
| Address | Bit address of a area | YES | int |
| CarParkId | Id of car park which hold area | NO | int |
| Status | The number indicate the status of area | NO | int |
| Active | Value show that the area is deleted or not | NO | bit |
| **ParkingLot** | Id | Unique Id for each parking lot | NO | int |
| Name | Name of parking lot | NO | nvarchar |
| Address | Bit address of a parking lot | NO | int |
| Status | The number indicate the status of parking lot | NO | int |
| AreaId | Value show that the parking lot is deleted or not | YES | int |
| CarParkId | Id of area which hold parking lot | NO | int |
| Active | Id of car park which hold parking lot | NO | bit |
| **Transaction** | Id | Unique Id for each transaction | NO | int |
| TransactionDate | The time customer book the parking lot | NO | datetime |
| EndTime | End of the booking time | YES | datetime |
| Amount | The total fee of transaction | NO | money |
| Status | The number indicate status of the transaction | NO | int |
| CarParkId | The code for the user to enter in the car park to unlock the booked parking lot | NO | int |
| ParkingLotId | Id of the lot user reserved | NO | int |
| AspNetUserId | Id of the user who reserved | NO | nvarchar |
| TransactionCode | The code for the user to enter in the car park to unlock the booked parking lot | YES | nvarchar |
| **CarPark** | Id | Unique Id for each car park | NO | int |
| Name | Name of car park | NO | nvarchar |
| Address | Address of carpark | NO | nvarchar |
| Phone | Phone number of car park | YES | nvarchar |
| Email | Email of car park’s owner | YES | nvarchar |
| Description | Information of car park | YES | nvarchar |
| Lat | Latitude of a car park | NO | nvarchar |
| Lon | Longitude of a car park | NO | nvarchar |
| Fee | Amount of money each hour | YES | money |
| Active | Value show that the car park is deleted or not | NO | bit |
| AspNetUserId | Id of the owner | YES | nvarchar |
| **AspNetRoles** | Id | Unique Id of each role | NO | nvarchar |
| Name | Role name | NO | nvarchar |
| **AspNetUserRoles** | UserId | Id of the AspNetUsers | NO | nvarchar |
| RoleId | Id of the AspNetRoles | NO | nvarchar |
| **AspNetUsers** | Id | Unique Id for each user | NO | nvarchar |
| Email | Email of the user | YES | nvarchar |
| EmailConfirmed | Check if the user confirm the email | NO | bit |
| PasswordHash | Hash of the user password, don’t save exact password | YES | nvarchar |
| SecurityStamp |  | YES | nvarchar |
| PhoneNumber | The number of the user | YES | nvarchar |
| PhoneNumberConfirmed | Check if the number is confirmed | NO | bit |
| TwoFactorEnabled |  | NO | bit |
| LockoutEndDateUtc |  | YES | datetime |
| LockoutEnabled |  | NO | bit |
| AccessFailedCount | The number of time access failed | NO | int |
| UserName | Username of the user | NO | nvarchar |

## Performance Measures

## Test Plan

The purpose of the test is to verify the functionality of the system.

Functions need to be tested. Functions to ensure technical requirements and system requirements of the user. Error will not happen after the trial

The next content will describe which function will be tested, which will not and plan for them.

### Features to be tested

We separate two part for testing: hardware and software.

#### Hardware

After assembling hardware devices to system, we test all of them to ensure that all hardware devices working well.

Here is a list of hardware devices that we tested.

|  |  |
| --- | --- |
| **Functions** | **Description** |
| Magnetometer Sensor HMC5883L | We test detect metal indoor, outdoor, in real environment at some points of time in day.  After many test case, we will get the estimate value of the sensor |
| NRF24L01 | RF Communicate:  We tested packets to transfer and receive. We define some variables, which have different sizes, to put in packet for transfer and receive via RF from sub to main and otherwise. If NRF24L01 could transfer and receive the packet, test case is “Passes”, else if NRF24L01 could not transfer or receive, test case is “Failed”.  We also tested the distance of RF in different topographic in many regions such as obstructions of buildings, walls or metal. Besides that, we also tested distance on the straight line, how far they could work. The test case is “Passed” if sub and main still send and receive signals, otherwise is “Failed” |
| Arduino Nano | The controller in sub node, we tested the SPI and I2C connection between Nano and NRF24L01, HMC5883L. If it can receive and transfer data, the test case is “Passed”, otherwise is “Failed”. |
| Raspberry Pi 3 | We tested ability of SPI connection between Raspberry Pi and NRF24L01. If it can receive and transfer data, the test case is “Passed”, otherwise is “Failed” |
| Led Segment | We need to tested the ability to display number on the led from 0 to 9 |
| Servo SG90 Mini | We test the ability to rotate from 0 degree to 90 degree of the servo |

#### Software

The features of user and web services will be focused and list below:

|  |  |
| --- | --- |
| Functions | Description |
| API Web Service | Services provide for Raspberry Pi 3.   * Get the list of Area * Get the list of Parking Lot * Update status of Parking Lot   Service provide for Mobile Application   * Register * Login * Get the list of car park * Get car park information * Get the list of area * Update the area * Get the list of parking lot * Update the parking lot |
| Mobile Application | Black box testing on user interface |

### Features not to be tested:

N/A

### Test environment

Web server: Firefox 52.0.2(32-bit)

OS: Windows 10

Android application: Android 5.0

Hardware: Test on Raspberry Pi 3

## System Testing Test Case

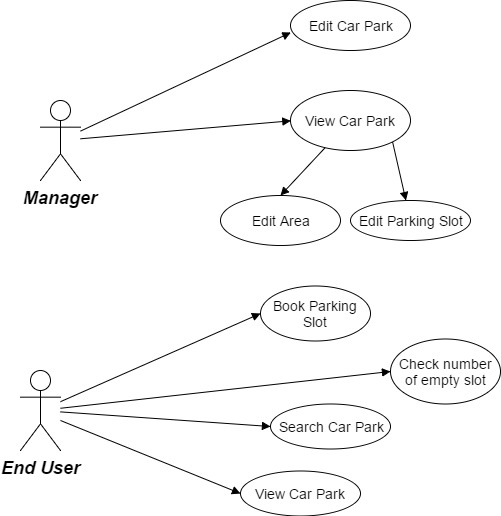


Figure : Manager, End User Core Flow

### Component Testing

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Id** | **Test Case Description** | **Test Case Procedure** | **Expected Output** | **Inter-test Case Dependence** | **Result** | **Test Date** | **Note** |
| C-01 | HMC5883L – Detect car park in range | Connect HMC5883L with Nano  Connect RGB Led with Nano  Add a car on the top of the board | The Led go red | N/A | Passed | 03/04/2017 |  |
| C-02 | HMC5883L – Detect car park in range | Connect HMC5883L with Nano  Connect RGB Led with Nano  remove a car on the top of the board | The Led go green | N/A | Passed | 03/04/2017 |  |
| C-03 | NRF24L01 + 7-Segment Led | Connect NRF24L01 with Nano  Connect 7Segment Led with Nano  Send a message with a number to Nano through RF | Display correct number send to Nano by 7Segment Led | N/A | Passed | 03/04/2017 |  |
| C-04 | Arduino Nano | Connect NRF24L01 with Nano  Connect HMC5883L with Nano  Put a car on top of the board | The status of lot is send to Pi and update on server | N/A | Passed | 03/04/2017 |  |
| C-05 | Raspberry Pi 3 | Connect NRF24L01 with Pi 3  Send a message from Pi to Nano through RF | Message receive in Nano | N/A | Passed | 03/04/2017 |  |
| C-06 | Servo SG90 Mini | Connect NRF24L01 with Nano  Connect Servo with Nano  User reserve a lot on mobile application | The servo rotate 90 degree to block a reserved lot | N/A | Passed | 03/04/2017 |  |

### API Web Service Testing

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Id** | **Test Case Description** | **Test Case Procedure** | **Expected Output** | **Inter-test Case Dependence** | **Result** | **Test Date** | **Note** |
| API-01 | Get the list of Area | Raspberry connect to Internet  Call API through HTTP request | Show the list of Area in JSON format | N/A | Passed | 31/03/2017 |  |
| API-02 | Get the list of Parking Lot | Raspberry connect to Internet  Call API through HTTP request | Show the list of Parking Lot in JSON format | N/A | Passed | 31/03/2017 |  |
| API-03 | Update status of Parking Lot | Raspberry connect to Internet  Call API through HTTP request | The API return JSON with format  {  “success”:true  } | N/A | Passed | 31/03/2017 |  |
| API-04 | Register | Mobile send request to API Service through HTTP | If create success, return JSON  {  “success”:false  }  If create fail, return JSON  {  “success”:false  } | N/A | Passed | 31/03/2017 |  |
| API-05 | Login | Mobile send request to API Service through HTTP | If login success, return JSON  {  “success”:false  }  If login fail, return JSON  {  “success”:false  } | N/A | Passed | 31/03/2017 |  |
| API-06 | Get the list of car park | Mobile send request to API Service through HTTP | Return the list of Car Park in JSON | N/A | Passed | 31/03/2017 |  |
| API-07 | Get the car park information | Mobile send request to API Service through HTTP | Return the Car Park information in JSON | N/A | Passed | 31/03/2017 |  |
| API-08 | Update the area | Mobile send request to API Service through HTTP | If login success, return JSON  {  “success”:false  }  If login fail, return JSON  {  “success”:false  } | N/A | Passed | 31/03/2017 |  |
| API-09 | Update the parking lot | Mobile send request to API Service through HTTP | If login success, return JSON  {  “success”:false  }  If login fail, return JSON  {  “success”:false  } | N/A | Passed | 31/03/2017 |  |

### Mobile Testing

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Id** | **Test Case Description** | **Test Case Procedure** | **Expected Output** | **Inter-test Case Dependence** | **Result** | **Test Date** | **Note** |
| M-01 | View Car Park | Open Mobile Application  Login with manager account  Select a car park | Show the car park information on screen | N/A | Passed | 31/03/2017 |  |
| M-02 | Edit Car Park | Open Mobile Application  Login with manager account  Select a car park  Select Edit  Fill in update information and click Finish | Update the information to the server and reload page | N/A | Passed | 31/03/2017 |  |
| M-03 | Edit Area | Open Mobile Application  Login with manager account  Select a car park  Select edit next to an area on the list  Fill in update information and click Finish | Update the information to the server and reload page | N/A | Passed | 31/03/2017 |  |
| M-04 | Edit Parking Lot | Open Mobile Application  Login with manager account  Select a car park  Select an area  Select edit next to a parking lot on the list  Fill in update information and click Finish | Update the information to the server and reload page | N/A | Passed | 31/03/2017 |  |
| M-05 | Book Parking Lot | Open Mobile Application  Login with end user account  Select a car park  Select Reserve  Fill in Addition Information  Click Finish | Show the reserved lot information if has  Otherwise, show the message error information | N/A | Passed | 31/03/2017 |  |
| M-06 | Check number of empty lot | Open Mobile Application  Login with end user account  Select a car park | The car park information screen show the number of empty lot | N/A | Passed | 31/03/2017 |  |
| M-07 | Search Car park | Open Mobile Application  Login with end user account  Enter the address on the textbox of the map | The map will transfer to the input address and show nearest car park from the address | N/A | Passed | 31/03/2017 |  |
| M-08 | View Car park | Open Mobile Application  Login with end user account  Select a car park | Show the car park information | N/A | Passed | 31/03/2017 |  |