Campus Parking

Software Requirements Specification

Version: 1.0.0

June 09, 2020

Team Members

Alekya Pochampally

Chetan Kudaravalli

Chinmayi Ambati

Manideep Chamala

Manisha Mengani

Xizi Chen

Submitted in partial fulfillment

Of the requirements of

CSIS 44-691 Graduate Directed Project 1

# **Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| **Date** | **Description** | **Author** | **Comments** |
| June 10, 2020 | Introduction to the project | Chetan Kudaravalli,  Manideep Chamala | Added Introduction chapter-1 |
| June 10, 2020 | General description of the project | Alekya Pochampally,  Chinmayi Ambati | Added general description of project chapter-2 |
| June 10,  2020 | Requirements of the project | Manisha Mengani,  Xizi Chen | Added requirements of the project chapter-3 |

# **Document Approval**

The following Software Requirements Specification has been accepted and approved by the following:

|  |  |  |  |
| --- | --- | --- | --- |
|  | Chetan Kudaravalli | Mr. | June 10, 2020 |
|  | Manisha Mengani | Ms. | June 10, 2020 |
|  | Manideep Chamala | Mr. | June 10, 2020 |
|  | Alekya Pochampally | Ms. | June 10, 2020 |
|  | Xizi Chen | Mr. | June 10, 2020 |
|  | Chinmayi Ambati | Ms. | June 10, 2020 |

**Table of Contents**

**Table of Contents Page Number**

1. Introduction
   1. Purpose
   2. Scope
   3. Definitions, Acronyms, and Abbreviations
   4. References
   5. Overview
2. General Description
   1. Product Perspective
   2. Product Functions
   3. User Characteristics
   4. General Constraints
   5. Assumptions and Dependencies
3. Specific Requirements
   1. External Interface Requirements
      1. User Interfaces
      2. Hardware Interfaces
      3. Software Interfaces
      4. Communications Interface
   2. Functional Requirements
   3. Use Cases
   4. Class/Objects
   5. Non-Functional Requirements

3.5.1. Performance

3.5.2. Reliability

* + 1. Availability
    2. Security
    3. Portability
  1. Inverse Requirements
  2. Design Constraints
  3. Logical Database Requirements
  4. Other Requirements
  5. Prototypes (for complete project)
  6. Use Case Diagrams

1. Design

4.1. ER diagram

4.2. GUI

1. Analysis Models

4.1. Data Flow Diagram

4.2. Sequence Diagram

## CHAPTER 1

**Introduction**

1.1 **Purpose**

It is basic that Students and faculty have to get to class on time and quite often we see a lot of delay in searching for free parking space. We are introducing the Campus Parking app which gives people the best option in saving time in searching for the free parking space on the campus. The app shows users how many open spaces are available in selected campus parking lots. Not only this, but the app also gives details about the positions of Faculty, Student, Handicapped, and also the Visitor parking in the Campus area.

1.2 **Scope**

Initially, we have a couple of options that the particular university has to execute for the best use of the app. First thing is that the university may issue a rule that every user must install the app to use the campus parking and as there are no censors coming into the play the user itself has to select the parking space in the app once it is occupied by that user, so that once the user selects the particular parking space that they occupied, that particular parking space turns into Red color and the remaining empty spaces will be as it is in White color making everyone know the availability of free parking spaces. This can be achieved by just using Notices round the Campus Parking Zone and on Campus.

The second thing which is an alternative to the first step and also which could be a better idea is that the University may allot separate parking lots for this app so that only users who have this app installed with them can use this space without having conflicts with the users who don't have this app installed.

**1.3 Definitions, Acronyms, and Abbreviations**

The following terms, acronyms, and abbreviations are used throughout this document.

**Campus Parking App:** We use this term many times and this is the actual app name that we are going to develop in the project which is Campus Parking. This helps users to know the availability of parking spaces in the Campus.

**Faculty, Student, and Visitor:** All these three terms refer to almost the same thing that these are the USERs who are going to use the parking space in the Campus and who has the app installed.

**Reserved:** This refers to the parking space that is already filled by other users.

**Vacant:** This refers to the parking space that is empty and any user can use the appropriate space by notifying the other users. This includes both the cases where the user forgets to update that he vacated the parking slot and also if the user reserved a slot and didn’t update it to the app.

**Token:** Token refers to the change that’s imposed on the user if he fails to update the usage of the parking lot.

**Parking Lots:** This refers to a cleared area that is intended for parking vehicles. Usually, the term refers to a dedicated area that has been provided with a durable or semi-durable surface.

**1.4**  **References**

• <https://fluttercrashcourse.com/>

• <https://www.tutorialspoint.com/dart_programming/index.htm>

• <https://flutter.dev/>

**1.5**  **Overview**

Campus Parking App provides users the details of the availability of the parking spaces in the Campus Parking Lot which also differentiates between Student, Faculty, and Visitors. This helps the users in saving big time in searching for the Parking space in the Campus.

**Chapter 2**

**General Description**

**2.1**  **Product Perspective**

The main perspective of campus parking application is to be helpful for students, faculty, or visitors to book a parking slot without any struggle during peak times.

**2.2**  **Product Functions**

The functions of this application are divided among a team of 6 members and assigned their roles to develop, with daily meetings and discussions for daily progress in the app.

**2.3 User Characteristics**

There are different types of users in this application such as students, faculty, and visitors. Firstly, everyone has to register in to the app to book a slot in the parking lot. After that, their characteristics depend on the type of parking they need such as handicapped, visitor, or faculty/worker parking. They are restricted to the availability of slots in a parking lot. This application serves as a first come first serve for users. Users can have their profile and report other vehicles with an image.

**2.4**  **General Constraints**

We as a team of 6 members have divided the application into various segments and decided to complete in 5 months and allocated a deadline for phase 1 which is the 1st week of July.

**2.5**  **Assumptions and Dependencies**

According to our project plan, we have to gather all our requirements which are necessary for the app development, then to have use cases and design ER-Diagram to represent the relationships between entities. At this stage, we have these assumptions for our project

• Setting up a developer environment according to required technologies.

• Setting up the project for both frontend and backend in Github.

• Designing the prototype of the project

• Listing out all the functionalities to be provided in the project and work distribution

• Designing UI – Login Page, functionality – A login page, Designing UI – Home page

and Designing UI – Login Page.

• Routing the page navigations and building the database where we populating data into

database.

• Merging the frontend and backend and debugging till then.

**Chapter 3**

**Specific Requirements**

1. **External Interface Requirements**

**3.1.1 User Interfaces**

**3.1.1.1 Login page**

This pagewill show the title of the app, prompts, and text boxes that allow the user input username/password.

Button “Sign up” for creating a new account.

Button “Sign in” for login

Button “Reset” for resetting one’s password.

**3.1.1.2 New User Page**

This page will show text boxes, prompts where users can input their details and register in the application.

**3.1.1.3 Reset password**

This page will allow the user to reset his forgotten password with a new password through a valid username.

**3.1.1.4 Home page**

This page will have buttons such as maps, reports, logout, and parking lots, where there are various parking lots in the university.

**3.1.1.5 Map Page**

This page will have a top view image of all parking lots where users can zoom in and out, back, and navigation buttons.

**3.1.1.6 Parking Area Page**

This page will show available lots for parking which is interactive like select/unselect and two buttons such as home and reserve.

**3.1.1.7 Select Slot Page**

This page will allow users to select an available slot in the parking lot which has slot id as down drop text box and back, cancel, reserve as buttons.

**3.1.1.8 Profile Page**

This page will have text boxes such as vehicle number, tickets, and vehicle type and a button back to return home.

**3.1.1.9 Report page**

This page will have vehicle number and image as text boxes and back, cancel, report as buttons.

**3.1.2 Hardware Interfaces**

• Processor: Minimum 1 GHz; Recommended 2GHz or more

• Wireless adapter (Wi-Fi) OR Mobile Data (3G/LTE)

• Hard Drive: Minimum 16 GB; Recommended 64 GB or more

• Memory (RAM): Minimum 1 GB; Recommended 4 GB or above

**3.1.3 Software Interfaces**

• Flutter

• SQLite

• Dart

• Visual Studio

**3.1.4 Communications Interface**

• IOS

• Android

1. **Functional Requirements**

A new user can sign up as a common user which could add one’s vehicle to the account. A common user could sign in, view, and manage his account including user information, vehicle, and balance.

Admin users not only have all functions of the common user but also can add funds for users.

**3.2.1 Login page**

• Can log in

• Reset the password if forgotten

• Signup

**3.2.2 Home page**

• Vacate the reserved parking lot

• Logout from app

• View list of all parking lots

• Report incorrect input by the user

• Visit university map page

**3.2.3 Parking area page**

• Return to home page

• Reserve a slot

• View the available slots in a particular parking lot

**3.2.4 Reserve page**

• Return to Parking area page

• Cancel entered data

• Reserve particular slot

**3.2.5 Map page**

• Return back to home page

• Image of the campus map with zoom in and out functionality

**3.2.6 Report Page**

• Return back to home page

• Cancel entered data

• Report inconsistent data- when the application shows a particular slot as reserved even if it is vacant

• Report incorrect updates – When the user forgets to reserve a particular slot after

parking. In this case, the app shows that a particular slot is available even if it is

reserved.

**3.2.7 Profile page**

• Can view the number of tickets and their profile information like the following

1. Vehicle number

2. Type of the vehicle

3. Parking subscription expiry

4. Charge for their tickets

5. Type of the user – faculty/ student

• User can edit their profile information like

1. vehicle number

2. Type of the vehicle

3. password etc

1. **Use Cases**
2. **Class/Objects**
3. **Non-Functional Requirements**

**3.5.1 Performance**

**3.5.2 Reliability**

**3.5.3 Availability**

**3.5.4 Security**

**3.5.5 Portability**

1. **Inverse Requirements**

1. **Design Constraints**

1. **Logical Database Requirements**

1. **Other Requirements**

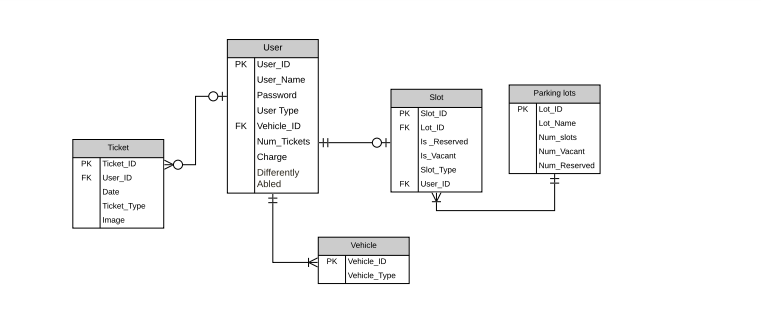
1. **Prototypes (for the complete project)**
2. **Use Case Diagrams**

**Chapter 4**

**Design**

**4.1 ER Diagram**

Below is the ER Diagram of our application where we have 5 entities: token, user, slot, vehicle and parking lots.



**4.1.1 About ER Diagram**

There are 5 entities namely Tickets, User, Slot, Parking lots, Vehicle.

**4.1.1.1 Ticket**

This entity is used for finding the user if they forget to update the application. The users will be charged based on the number of tickets they have. This entity will be having few attributes like

· Date : which will be telling the user on which day they got fined.

· Image : image of the screenshot or user vehicle

· Token type : reserve/ vacant

· UserID : This is the foreign key of the table and user to identify an individual

Relations : Each ticket is for 1 user only.

**4.1.1.2 User**

This entity is used to represent an individual using this app. The following are the attributes

· UserID : Unique number for each user

· User\_Name : Name of the user

· Password : Password of the user

· User Type: faculty or student

· Vehicle\_ID: this is a foreign key used to track individuals vehicle

· Num\_Ticket: Number of tokens per user

· Charge: will be calculated based on the number of tickets users have.

· Differently abled: This is used for directing to reserved parking slots for specific users

Relations : Each user can have

· multiple tickets

· multiple vehicles

· 0 or 1 slot

**4.1.1.3 Vehicle Entity**

This entity represents the vehicle used by the user. The following are its attributes

· Vehicle\_ID: This is a primary key used to uniquely identify the vehicle

· Vehicle\_Type: 2 or 4 wheeler

Relations: 1 Vehicle will be having only 1 user

**4.1.1.4 Parking lots**

This entity represents all the parking lots in the university. The following are its attributes

· Lot\_ID: This is the primary key and uniquely identify particular parking lot

· Lot\_Name: name of the parking lot

· Num\_slots: this gives the capacity of the parking lot

· Num\_Vacant: Number of available slots for parking

· Num\_Reserved: Number of reserved slots

Relations: Each parking lot can have 1 or more slots

**4.1.1.5 Slot**

This entity represents each slot in any parking lot. This entity have the following attributes

· Slot\_ID: This is the primary key and uniquely identifies each slot

· Lot\_ID: This is the foreign key and represents the Lot\_ID in parking lots entity

· Is\_Reserved: states whether the entity is reserved or not

· Is\_Vacant: states whether the entity is vacant or not.

· Slot\_Type: slot meant for differently abled persons or not.

· User\_ID: this is a foreign key which identifies users uniquely.

Relations:

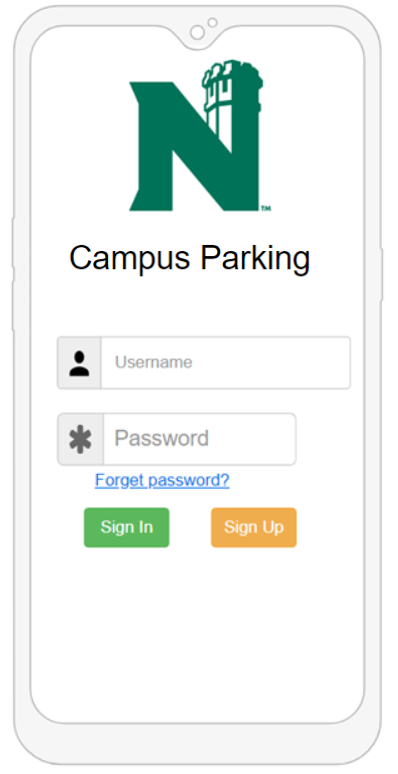
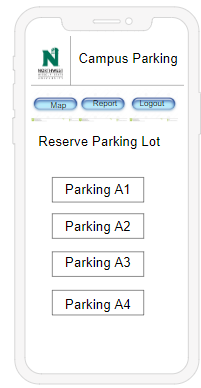
· Each slot have 1 user

· Belongs to only 1 parking lot

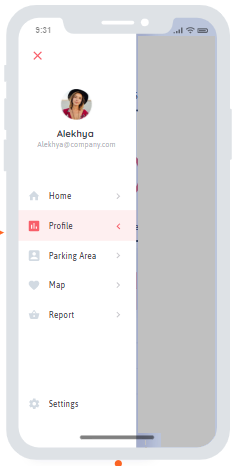
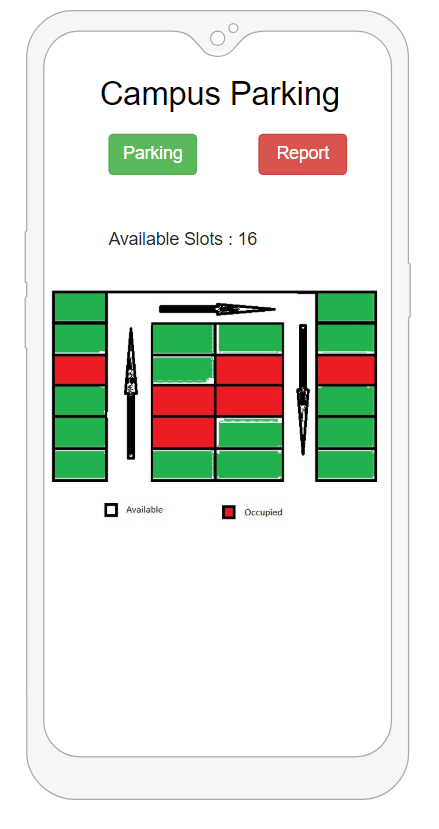
**4.2 GUI**

These are mockup’s of our application’s Graphical user interface.

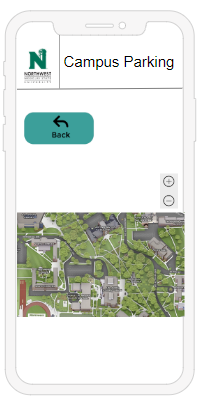
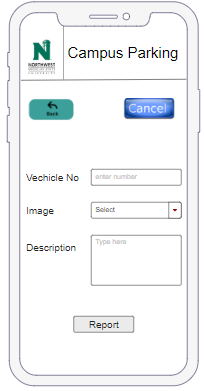
**Login Page Home Page**

** **

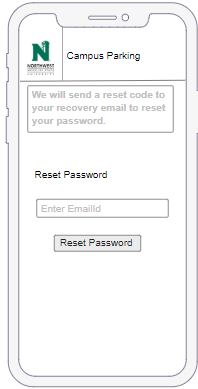
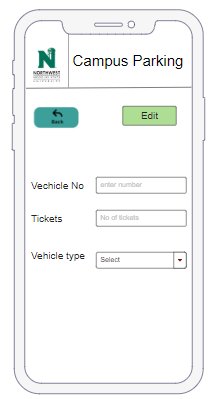
**Index Page Parking Page**

** **

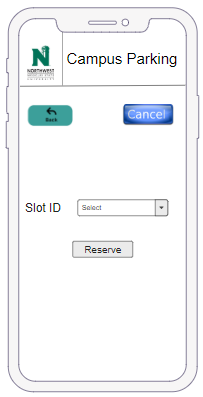
**Map Page Report Page**

** **

**Forgot Password Page Profile Page**

** **

**Connection Page**

****