Software Requirements Specification

for

Handicapped Parking System

Version 1.0 approved

Prepared by Richu Mathew, Leif Nevener, Austin Taylor, Curt Collins

University of Central Oklahoma

September 25, 2019

Table of Contents

Table of Contents ii

Revision History ii

1. Introduction 1

1.1 Purpose and Scope 1

1.2 Definitions, Acronyms and Abbreviations 1

1.3 References 1

1.4 Overview 1

2. Overall Description 1

2.1 Product Functions 1

2.2 User Classes and Characteristics 1

2.3 Constraints 2

2.4 Assumptions and Dependencies 2

3. Specific requirements 2

3.1 External Interface Requirements 2

3.1.1 User Interfaces 2

3.1.2 Hardware Interfaces 2

3.1.3 Software Interfaces 2

3.1.4 Communications Interfaces 3

3.2 Functional requirements 3

3.2.1 Functional requirement 1 3

3.2.2 Functional requirement 2 3

3.3 Performance Requirements 3

3.4 Software System Attributes 4

3.4.1 Availability 4

3.4.2 Security 4

3.5 Design constraints 4

3.5.1 Standards Compliance 4

3.5.2 Hardware Limitations 4

Appendix A: Issues List 4

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
| Leif Nevener | 9/25/19 | Add use case diagrams | 1.0` |
| Richu Mathew | 9/26/19 | Wrote user interface section( section 3.1.1) and created GUI interface. | 1.0 |
| Austin Taylor | 9/27/19 | Wrote Introduction (section 1). | 1.0 |
| Curt Collins | 9/27/19 | Wrote functional requirements | 1.0 |

# Introduction

## Purpose and Scope

The Handicapped Parking System (HPS) is designed to solve the problem of finding unoccupied handicap parking spaces at the University of Central Oklahoma in Edmond, Oklahoma. HPS is designed to be accessed via web browsers on both Android and iOS smart devices and Windows and OSx operating systems. Release version 1.0 includes the ability for a user to observe unoccupied as well as the ability to reserve unoccupied handicap parking spaces.

## Definitions, Acronyms and Abbreviations

(HPS) Handicapped Parking System.

## References

Resources used in this project are StarUML(staruml.io), Microsoft Productivity Suite(www.office.com).

## Overview

This system is designed to assist disabled faculty, staff, and students to find handicapped parking spaces on the UCO campus. Besides finding spots the actor will also be able to set the spot reserved for a certain amount of time.

# Overall Description

## Product Functions

In this project, the system can let the actor register an account using their phone number. By using actor’s phone number will be used by the system primarily for administrative purposes. A subset of administrative use includes push notification to actor 15 minutes prior to reservation time expires, updating actor’s information using phone number as a key. When the actor wants to check for a spot the app will utilize Google maps API to display spots available on the map. After the actor reserves that spot, that spot will be marked as reserved, so that spot will be locked, and therefore preventing from other users from selecting the same spot.

A close up of a map

Description automatically generated

A close up of a map

Description automatically generated

A close up of a map

Description automatically generated

## User Classes and Characteristics

The actors present in this system consist of the driver, time, messaging system, and navigation system.

Characteristics of the actors:

Driver: name, email, phone number, license plate number,

Reservation: relation to the driver, it has a start time and end time, a relation to the space.

Messaging system: relation between sender and recipient, communication system.

Navigation system: GPS coordinates, images fed to it, step-by-step navigational directions

## Constraints

The constraints on this project are the three-month development time frame, number of developers, and customer cost of Microsoft Azure’s hosting.

## Assumptions and Dependencies

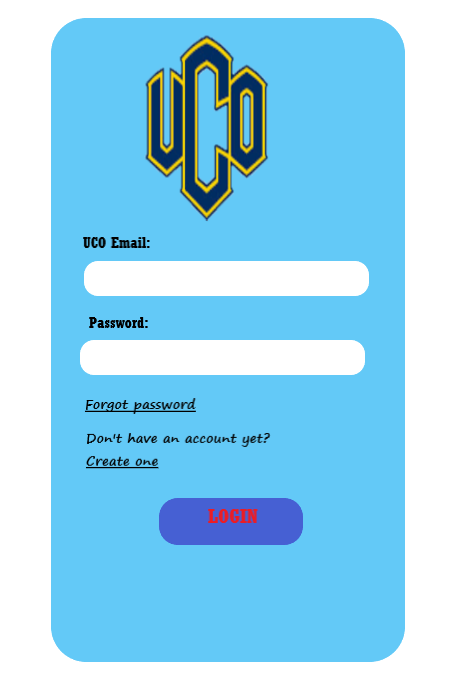
The University GSM modem server or SMSS is a dependency. Similarly, the University provided Wi-Fi or user-provided mobile data.

# Specific requirements

## External Interface Requirements

### User Interfaces

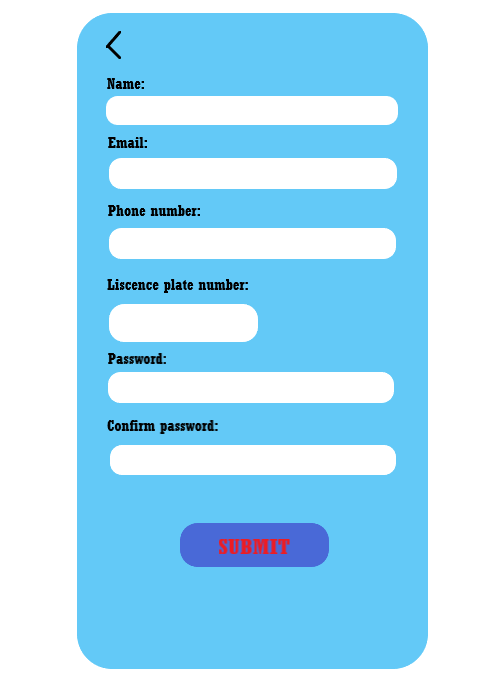
When the application starts, the user is presented with login screen. In this GUI the user will enter their UCO email/User ID and password. After the user enters their credentials and if it matches the information in the database, the user will be taken to the home page.



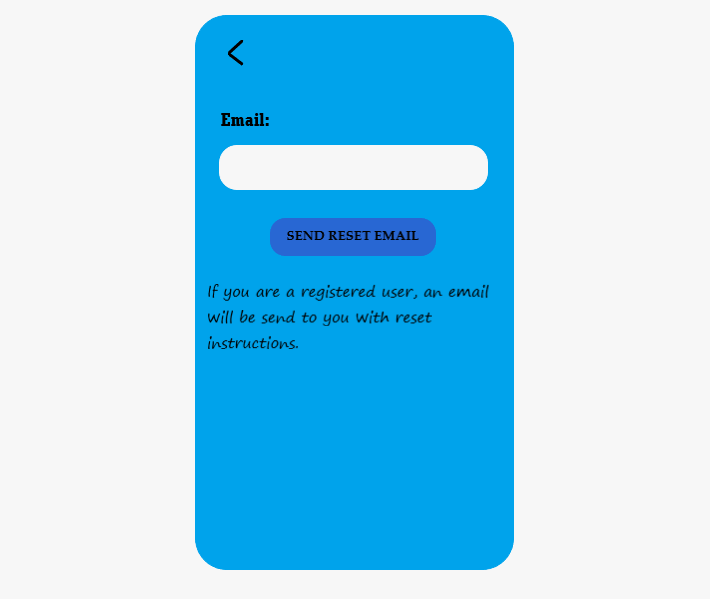
After entering the home page, the user will be able to view spots which are available (will be highlighted in green) and spots which are already reserved (will be highlighted in red). From the available spots' user will select a desired spot. After the user selects the spot, the information will be saved in the database. The user will be prompted for starting time and ending time. Later, the user will click “TAKE ME THERE” button to show the route from users to the user’s reserved spot. The user will also have a choice to cancel the reservation.



For first-time user, he/she can create an account by selecting “Create one” link in the login page. After clicking that link, user will be prompted to input the information in the GUI below. After user inputs those data, the system will search the database for existing records. If positive result is received back from the system, account will be created. Otherwise, the user will be notified that an account already exists.



If the user forgot their password, he/she can reset it by clicking “Forgot password” link in the login page. In the password reset page, the user will be prompted to enter his/her UCO email address. If the email entered corresponds to a data in the database, an email will be sent with reset instructions.



For separate user interface specification, please refer to section 3.2.1.

### Hardware Interfaces

The hardware interfaces included are mobile devices and a GSM modem.

### Software Interfaces

The software interfaces utilized in this project are Google maps and Sequel Task Scheduler.

### Communications Interfaces

Our protocol and standardization for encryption is SSL over HTTPS. The format of the text messages need to comply to current SMS standards. The format of the URL sent to the navigation system needs to comply to simple URL standards common to any web browser. Communication with the user shall occur via HTML pages suitable for display on a mobile device.

## Functional requirements

### Login

#### Introduction

Simple user login form to collect a username and password and validate same. All users have the same authorization level within the system thus reducing complexity and advoiding many motives for “hacking”. Options are provided for a forgotten password, as well as, new user registration.

#### Inputs

Data inputs include only a username (UCO email) and a password {must be at least 8 characters and include a number}, each entered by the user in separate textbox type controls. Alternately, the user may indicate a forgotten password. Alternately, the user may indicate a “first access” (register driver) condition/button.

#### Processing

Username and password will be verified/validated by the system; successes will be transferred to the **Home** screen form, while a failure will be shown this same form with the addition of an “Invalid – try again” type message. If the “Forgot Password” option is chosen, the form will generate an alternate password, which overwrites the existing password in the system and emails same to user; the username entered must match a previously registered user. The “Create One” link transfers control to the Register Driver function.

#### Outputs

If Forgot password, then an email to the user; otherwise, nothing.

### Registered Driver

#### Introduction

The Register Driver form collects initial information from a first-time user to the system, validates correct format of the data and will message the new user/driver to confirm the registration.

#### Inputs

Data inputs include preferred password, driver first and last name, license plate number, and phone number. The email address is also “collected” but must be within the “@uco.edu” domain and will double as the username for the system.

#### Processing

The new user is presented with a form containing no data and inputs the required information as specified above and clicks the “Confirm” control. Incomplete or invalid data results in the form being redisplayed with a warning notation next to the invalid item(s). Valid and completed forms result in a new Driver record created in the database, an email confirmation sent to the new Driver and a redirect to the Login form.

#### Outputs

User email (username), password, name, license plate, and phone number are stored to the database.

An email is sent to the new user confirming registration.

### Update Driver Profile

#### Introduction

The Driver is presented with a form containing his/her existing registration information and the Driver may edit any value except the username.

#### Inputs

Data inputs include preferred password, driver first and last name, license plate number, and phone number.

#### Processing

The system retrieves the current Driver’s registration information from the database and presents the Driver with a form identical to the “New User” form. The Driver then may edit any information except their email address (username) above and clicks the “Confirm” control. Incomplete or invalid data results in the form being redisplayed with a warning notation next to the invalid item(s). Valid and completed forms result in the existing Driver record updated in the database. The Driver is then redirected to the “Home” form.

#### Outputs

User email (username), password, name, license plate, and phone number are updated in the database.

### Message Driver

#### Introduction

The Message Driver form allows a Driver to message another driver through the system by referencing a license plate number. This allows Drivers to message one another (possibly a vehicle has been parked in the current Driver’s reserved spot) without unnecessarily divulging personal information. The recipient will receive a simple message as input by the sender as well as the senders Name and Email address (this requirement is in place to reduce the possibility of any existing Driver sending unwelcome messages to another driver).

#### Inputs

Data inputs include the license plate number of the recipient and a message of 100 characters or less.

#### Processing

The system retrieves the logged in Driver’s name and email address and is presented with a form similar a text messing form which is prefilled with their email as the sender, which cannot be changed. The recipient Driver’s license plate number and message text are input by the Driver and clicks the “Confirm” control. The system searches the database to match the license plate number to a Driver’s email address. If found, an email is generated and sent to that address; if not, no further processing occurs. The Driver is then redirected to the Home form.

#### Outputs

An email is sent to a Driver upon a matching plate #, otherwise nothing.

### Logout

#### Introduction

This is a simple system logout function that may be accessed from the Home form and terminates the Driver’s current session. Subsequent requests for system functions will require logging in again.

#### Inputs

None.

#### Processing

The system presents a message for the Driver to confirm the intention to logout. The Driver may confirm the request and be redirected to the Login form, or cancel the request and remain on the current form.

#### Outputs

None.

### View Available Spaces

#### Introduction

View Available Spaces will display a form to the Driver in order to select a lot or lots from a list (as a sort of filter) and present a list of all handicapped spaces available within the selected lot(s) and their current statuses: Available, Occupied, or Reserved. All spaces will show the next reservation start/end time for the space, and spaces marked “Reserved” or “Occupied” will display that Driver’s license plate number.

#### Inputs

None.

#### Processing

The system retrieves list of all handicapped spaces based on the Driver’s input and retrieves the availability information from the database. The list of spaces is presented to the Driver (along with the ability to filter the list based on parking lot) and existing reservations and occupations for each space. The Driver may select a space in the list to reserve provided no reservation (or occupation) currently exists for this Driver and the Make Reservation form will be displayed.

#### Outputs

If the Driver opts to make a reservation, then the parking space designation; otherwise, none.

### Logout

#### Introduction

This is a simple system logout function that may be accessed from the Home form and terminates the Driver’s current session. Subsequent requests for system functions will require logging in again.

#### Inputs

None.

#### Processing

The system presents a message for the Driver to confirm the intention to logout. The Driver may confirm the request and be redirected to the Login form, or cancel the request and remain on the current form.

#### Outputs

None.

### Make Reservation

#### Introduction

Make Reservation allows the Driver to select a space from the list presented in View Available Spaces and reserve the space for a time period of no more than 8 hours and up to 24 hours in advance provided no reservation (or occupation) currently exists for this Driver.

#### Inputs

The parking space designation selected by the Driver in the View Available Spaces form. The intended timeframe for the reservation from the user is entered. The times may not exceed an 8- hour period and must start no more than 24 hours later than the current system time; additionally, the start time must be prior to the end time and neither may be prior to the current system time.

#### Processing

The system displays a form to collect the timeframe information from the Driver for the requested space and validates the data according to the rules above. Upon data validation, the system stores the reservation in the database and updates the status of the space to “Reserved”.

#### Outputs

The Reservation is stored to the database, or nothing.

### Park

#### Introduction

Park allows the Driver to change the status of a current Reservation from “Reserved” to “Occupied”. Any Reservations in the system that have not been marked as “Occupied” within 30 minutes of their reserved start time will be changed to “Available” by the system independent of any user input, and a message will be sent to the Driver.

#### Inputs

None.

#### Processing

The system retrieves the existing Reservation for the currently logged in Driver and displays a form to display a Driver’s current Reservation with a button/link for the Driver to confirm that that Driver now occupies the space. Upon confirmation, the system updates the status of the Reservation to “Occupied” in the database and the Driver is presented with the previous form.

#### Outputs

The Reservation is updated in the database.

### Cancel Reservation

#### Introduction

Cancel Reservation allows the Driver to change the status of a currently reserved space from “Reserved” to “Available”. Note that any existing Reservation prevents a Driver from making a new Reservation.

#### Inputs

None.

#### Processing

The system retrieves the existing Reservation for the currently logged in Driver and displays a form to display a Driver’s currently occupied space with a button/link for the Driver to confirm that that Driver no longer wishes to reserve the space. Upon confirmation, the system updates the status of the Reservation (occupation) to “Available” in the database and the Driver is presented with the previous form.

#### Outputs

The Reservation is cleared in the database.

### Leave

#### Introduction

Leave allows the Driver to change the status of a currently occupied space from “Occupied” to “Available”. Any occupied spaces in the system that have not been marked as “Occupied” within 30 minutes of their reserved end time will be changed to “Available” by the system independent of any user input, and a message will be sent to the Driver.

#### Inputs

None.

#### Processing

The system retrieves the existing Reservation for the currently logged in Driver and displays a form to display a Driver’s currently occupied space with a button/link for the Driver to confirm that the Driver has now vacated the space. Upon confirmation, the system updates the status of the Reservation (occupation) to “Available” in the database and the Driver is presented with the previous form.

#### Outputs

The Reservation is updated in the database.

### Remind Driver

#### Introduction

Remind Driver facilitates the system’s internal “garbage collection” feature wherein the Scheduler enforces the rules that a Driver must occupy a reserved space within 30 minutes of its start time and must vacate the space within 30 minutes of its end time or the space will be marked as Available.

#### Inputs

None.

#### Processing

The Scheduler actor initiates a search of all spaces marked as “Reserved” with a start time more than 15 minutes before the current system time (with no “reminded” flag set) and all spaces marked “Occupied” with an end time more than 15 past the current system time. The system then retrieves the email addresses for these Drivers and sends a standard reminder message to each through the Messaging System and flags the reservation as “reminded” to avoid multiple warning messages being sent as the 30-minute deadline approaches. The Scheduler also initiates a search of all spaces marked as “Reserved” with a start time more than 30 minutes before the current system time and all spaces marked “Occupied” with an end time more than 30 past the current system time. The system then retrieves the email addresses for these Drivers and sends a standard “reservation cancelled” message to each through the Messaging System and clears any “reminder sent” flags in the database.

#### Outputs

Text messages sent to selected Drivers via the external messaging system

The reservation is flagged as “reminded” if a 15-minute message was sent

The reservation is cancelled and the “reminded” flag cleared if a 30-minute message was sent.

### Navigate to Space

#### Introduction

Navigate to Space provides the Driver with the ability to navigate to a parking space that has been reserved by the Driver.

#### Inputs

None.

#### Processing

The system searches an existing Reservation for the currently logged in Driver in the database, and retrieves the GPS coordinates for that space that are also stored in the database. The GUI then spawns another window and sends these coordinates to the Navigation system for display in that window.

#### Outputs

GPS coordinates sent to Navigation system

## Performance Requirements

Should the availability of the GSM provider drop, driver text-messaging aspects of the system shall be directly impacted. Given the estimated number of users, standard deployments of Window server IIS and Sequel Server within an Azure cloud far exceed the projected demands on the system.

## Software System Attributes

### Availability

Factors include hardware and communication reliability, handled by our choice of Microsoft Azure with 99.99% or better reliability. Internal code reliability measurements shall be determined after system testing. The availability of the GSM communication provider (Verizon, AT&T, etc.) is also factor in the performance of the system.

### Security

Server-based security features shall be implemented to prevent access to any other part of the system beyond the login screen to anyone who is not successfully logged in.

## Constraints

### Standards Compliance

Given the standards required for obtaining and maintaining a driver’s license, we believe that no special accommodation needs to be made to the user interface according to ADA guidelines.

### Hardware Limitations

The estimated number of users is approximately less than two hundred per day. If this population increases dramatically, server scalability in terms of memory, storage, and communication throughput is handled by Microsoft Azure hosting and can be scaled by payment options. The GSM modem can handle well beyond the estimated number of users, should this number dramatically increase beyond current predictions, additional modems can be procured for additional support.

Appendix A: Issues List

Currently we do not have a GSM modem, we have a modem simulator to test text-based communications.