Smart Parking System Description

Danyil, Kovalchuk

WSU Cpt S 322

# Introduction

* <insert a description of the document – 1 paragraph>
* <Purpose of the System>
* <Audiences and Goals>
* <insert a description of the app/game– capturing business requirements at a high level>

This document is indented to describe the Smart Parking system, which is going to use a set of sensors and beacons to compute the location of each car, and then show the user what parking slots are available and what slots are taken. This system is intended to be used by the system administrator and a regular user, and is intended to make finding the parking spot mush more efficient and easier.

The system must be able to show taken and free parking slots, must allow the administrator to add new cars, users, and beacons. It also needs to display some statistical information.

# Users

This section describes the types of users of the app/game

* Admin
* Regular User

# Participants

This section describes the participants of the app/game

* Admin
* Regular User
* Drivers
* Cars
* Beacons
* Parking slots
* Sensors

# User Interface Wireframes

This section describes the main screens that the application will have.

Examples

* Login Page– include a description>

This screen will prompt the user to login in the account.

Graphical user interface

Description automatically generated

* Parking Lot Display– include a description>

This screen will show all parking slots and show which slots are occupied, and which are not. This also is going to show some statistics in the left panel and will have a login button.

Graphical user interface

Description automatically generated

* Add/remove users– include a description>

This screen will allow admin to add new users and delete existing users.

Graphical user interface, application

Description automatically generated

# User Stories

List out user stories in the format

*As a <user type> I want <feature> so that <reason>*

*Smart Parking example*

* <Parking available>

As a user I want to see available parking lots, so that I can find parking spot more efficiently.

* <Admin login>

As an admin I want to have login into account that has more features than a regular account, so that not every user has admin privileges.

* < Parking lot display>

As a user I want to have display which shows a parking lot, so that it is easier to find a free parking spot.

* <Integrative menu>

As a user I want to have integrative menu, so that it is easier to navigate through the app.

* <add/Remove Users>

As an admin I want to be able to add or remove users, so that I can have control over who is allowed to use the system and who is not.

* <Trilateration>

As a user I want to use a Trilateration, so that I can determine the location of each car in the parking lot.

* < Adding a car>

As an admin I want to be able to add the cars, so that I can have new cars shown in the app.

* <Add beacon>

As an admin I want to be able to add beacons, so that I can I associate them with the cars and be able to find their location.

* <Loading Data from firebase>

As an user, I want to load data from the firebase, so that I can use it to find the location of each a car.

* How to star the simulation.

As the user I want to know how to start the simulation, so that I can use this app.

# Tasks

This description describes how the project will be managed. The below are high level tasks that need to be completed, identifying risks, and other

<below are some major tasks that need to be completed. I’ve laid out some ideas – you should have sub-tasks. Use the user stories as a way to determine what you’ll need to clarify, what elements you think you’ll need to design, etc. The purpose here is to focus on tasks you’ll need to do complete the assignnment

* <Prototyping>
  + Firebase
  + C# for implementing the app
  + JSON Files
  + Parsing
* <Requirements Doc>
  + Gathering the requirements
* <Design>
  + Think about all the parts of the app, and how they would relate to each other
* <Implementation>
  + Setup github
  + Verify git commit works
  + Implement triangulation function
  + Implement function for retrieving information from the database
  + Make a graphical user interface
  + Connect all the pieces of the program to one
  + Test and debug app
  + Submit using github