

**Software Requirements Specification**

**for**

Parking Tracker

**Version <0.1>**

**Prepared by**

**Group Name: Team 4**

|  |  |  |
| --- | --- | --- |
| **Ryan “Bob” Dean** | **11399597** | **ryan.dean@wsu.edu** |
| **Vitaliy Gavrilov** | **11205700** | **vitaliy.gavrilov@wsu.edu** |
| **Jason Moss** | **11017154** | **jason\_moss@wsu.edu** |
| **Joel Uyesugi** | **<student #>** | **j.uyesugi@wsu.edu** |
|  |  |  |

|  |  |
| --- | --- |
|  |  |
| **Date:** | **<place the date of submission here>** |
|  |  |
|  |  |
|  |  |

**Contents**

**REVISIONS**

**1** **INTRODUCTION**

1.1 Document Purpose

1.2 Product Scope

1.3 Intended Audience and Document Overview

1.4 Definitions, Acronyms and Abbreviations

1.5 Document Conventions

1.6 References and Acknowledgments

**2** **OVERALL DESCRIPTION**

2.1 Product Perspective

2.2 Product Functionality

2.3 Users and Characteristics

2.4 Operating Environment

2.5 Design and Implementation Constraints

2.6 User Documentation

2.7 Assumptions and Dependencies

**3** **SPECIFIC REQUIREMENTS**

3.1 External Interface Requirements

3.2 Functional Requirements

3.3 Behaviour Requirements

**4** **OTHER NON-FUNCTIONAL REQUIREMENTS**

4.1 Performance Requirements

4.2 Safety and Security Requirements

4.3 Software Quality Attributes

**APPENDIX A – DATA DICTIONARY**

**APPENDIX B - GROUP LOG**

**Revisions**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Primary Author(s)** | **Description of Version** | **Date Completed** |
| Draft Type and Number | Full Name | Information about the revision. This table does not need to be filled in whenever a document is touched, only when the version is being upgraded. | 00/00/00 |

# 

# *<In this template you will find text bounded by the “<>” symbols. This text appears in italics and is intended to guide you through the template and provide explanations regarding the different sections in this document. There are two types of comments in this document. These comments that are in black are intended specifically for that course. These comments that are in blue are more general and apply to any SRS. Please, make sure to delete all of the comments before submitting the document.*

# *The explanations provided below, do not cover all of the material, but merely, the general nature of the information you would usually find in SRS documents. It is based on the IEEE requirements and was adapted specifically for the needs of Software Engineering 3K04/3M04 courses. Most of the sections in this template are required sections, i.e. you must include them in your version of the document. Failure to do so will result in marks deductions. Optional sections will be explicitly marked as optional. If you have any questions regarding this document please refer to the MiniThermostat SRS example on the course web-site.>*

# **Introduction**

*<TO DO: Please provide a brief introduction to your project and a brief overview of what the reader will find in this section.>*

## **Document Purpose**

*<Identify the product whose software requirements are specified in this document, including the revision or release number. Describe the scope of the product that is covered by this SRS, particularly if this SRS describes only part of the system or a single subsystem.*

*TO DO: Write 1-2 paragraphs describing the purpose of this document as explained above.>*

The product herein described is the Parking Tracker. The major scope of this product will be a smart phone app that can be used a to identify cars parked in individual parking slots. To do this, the product should be able to take a picture of the license plate, process the image to retrieve the license plate number and store that information along with the specified parking slot, geotag and time/date. Besides the smart phone app, a data store along with an OCR dedicated server are envisioned for the future, but that is not in the scope of this document. Parking Tracker will be built according to this specification with a release prototype version of 0.1.

## **Product Scope**

*<Provide a short description of the software being specified and its purpose, including relevant benefits, objectives, and goals.*

*TO DO: 1-2 paragraphs describing the scope of the product. Make sure to describe the benefits associated with the product.>*

Parking Tracker is an app that helps automate the process of tracking car parking density. This will be tailor made for South Waterfront Community Relations (<http://www.southwaterfront.com/>). The goal of this software will be to automatically populate as much data as needed, including location and time while the user simply walks from car to car taking pictures of the licensee plates. To date this data has been collected by hand but this software comes at a great benefit as it will do these things automatically and greatly improve time efficiency.

Our objectives are simply to get everything done that the customer has asked of us, including the app and back end support and we do believe that will be possible by May. However, our goal for this semester is to get the app GUI part of the app down, along with the OCR of pictures, integrate maps and develop a modular system of storing the data to be flexible in the future. We aim to have the process of moving the data from the phone to the back end done as soon as possible, but do not necessarily foresee that happening this semester.

## **Intended Audience and Document Overview**

*<Describe the different types of reader that the document is intended for, such as developers, project managers, marketing staff, users, testers, and documentation writers (In your case it would probably be the “client” and the professor). Describe what the rest of this SRS contains and how it is organized. Suggest a sequence for reading the document, beginning with the overview sections and proceeding through the sections that are most pertinent to each reader type.>*

This document us for our group to organize our plan, for Dr. Zhao and Dr. Cochran, and for Pete Collins from South Waterfront.

The document is meant to be read sequentially and should be straightforward to the use of the software as opposed to the technical innards, this document will explain how and where the software can be used.

## **Definitions, Acronyms and Abbreviations**

*<Define all the terms necessary to properly interpret the SRS, including acronyms and abbreviations. You may wish to build a separate glossary that spans multiple projects or the entire organization, and just include terms specific to a single project in each SRS.*

*TO DO: Please provide a list of all abbreviations and acronyms used in this document sorted in alphabetical order.>*

App/Software - Parking Tracker smartphone application

OCR - Optical Character Recognition

Tesseract - Open source OCR engine maintained by Google

Tess-Two - Android library utilizing Tesseract

Back end - Permanent data store and future OCR server

SWF - South Waterfront

## **Document Conventions**

*<In general this document follows the IEEE formatting requirements. Use Arial font size 11, or 12 throughout the document for text. Use italics for comments. Document text should be single spaced and maintain the 1” margins found in this template. For Section and Subsection titles please follow the template.*

*TO DO:* *Describe any standards or typographical conventions that were followed when writing this SRS, such as fonts or highlighting that have special significance.* *Sometimes, it is useful to divide this section to several sections, e.g., Formatting Conventions, Naming Conventions, etc.>*

## **References and Acknowledgments**

*<List any other documents or Web addresses to which this SRS refers. These may include user interface style guides, contracts, standards, system requirements specifications, use case documents, or a vision and scope document.*

*TO DO: Use the standard IEEE citation guide for this section. An example citation guide is posted for you on the website.>*

[1] (2014, October). Android Style Guide [Online]. Available: http://developer.android.com/design/style/index.html

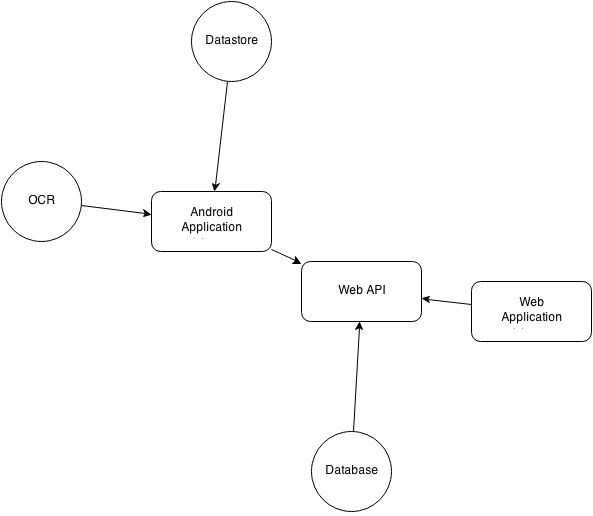
# **Overall Description**

## **Product Perspective**

*<Describe the context and origin of the product being specified in this SRS. For example, state whether this product is a follow-on member of a product family, a replacement for certain existing systems, or a new, self-contained product. If the SRS defines a component of a larger system, relate the requirements of the larger system to the functionality of this software and identify interfaces between the two. In this part, make sure to include a simple diagram that shows the major components of the overall system, subsystem interconnections, and external interface. In this section it is crucial that you will be creative and provide as much information as possible.*

*TO DO: Provide at least one paragraph describing product perspective. Provide a general diagram that will illustrate how your product interacts with the environment and in what context it is being used.>*

This product will be entirely new, as previously all the data was recorded and organized by hand. This mean we will be designing both the app and the database from the ground up, with input from SWF. The app will pull text from a photo taken by an Android smartphone or tablet, and will later put all gathered text into a database. The intended use of this is taking license plate numbers and storing them to track trends and data on parking usage.



## **Product Functionality**

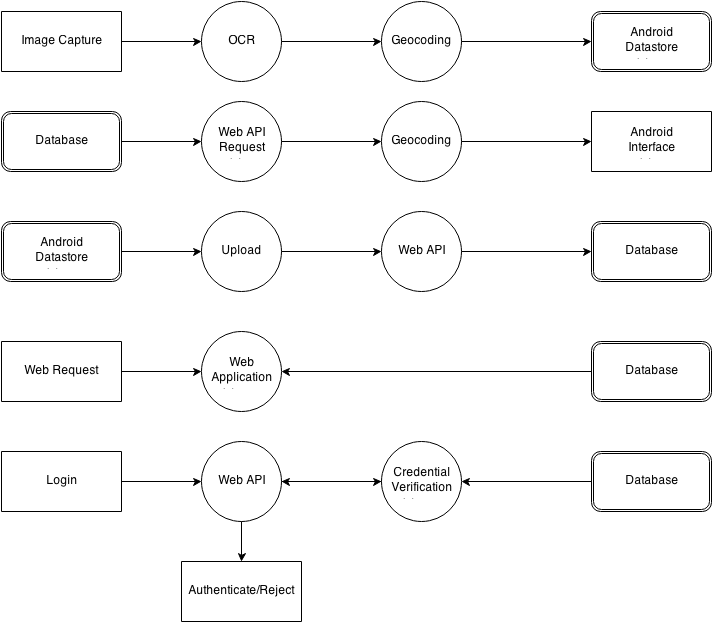
*<Summarize the major functions the product must perform or must let the user perform. Details will be provided in Section 3, so only a high level summary is needed here. Organize the functions to make them understandable to any reader of the SRS. A picture of the major groups of related requirements and how they relate, such as a top level data flow diagram or object class diagram, will be effective.*

*TO DO:*

*1. Provide a bulleted list of all the major functions of the system*

*2.* ***(Optional)*** *Provide a Data Flow Diagram of the system to show how these functions relate to each other.>*

* The app must be able to take pictures
  + Those pictures are to be analyzed to read a license plate number
  + The recognized number must be available to confirm/edit
* The app must be able to collect the license plate data with the date and time and location
* The app must be able to buffer the data of parked cars until it is wifi connected
* The app must be able to communicate with a data store as indicated by SWF
* The app must be able to map the neighborhood into a series of blocks and block faces and then re-create the map to collect data.
* The data store must be able to authenticate users to control access to its interface.
* The data store must be able to present an analysis of parking usage over time and the parking behavior of individual vehicles.



## **Users and Characteristics**

*<Identify the various users that you anticipate will use this product. Users may be differentiated based on frequency of use, subset of product functions used, technical expertise, security or privilege levels, educational level, or experience.*

*TO DO:*

*1. Describe the pertinent characteristics of each user. Certain requirements may pertain only to certain users.*

*3. Distinguish the most important users for this product from those who are less important to satisfy.>*

The front end will have one type of user, a SWF employee who is scanning parking spots.

The web aspect will have three user types. There will be a data collector, who is a user who is using the android application to build a data set and then submit it. They will have access to review submitted data and possibly edit/process recent submissions as well as to modify the neighborhood layout. There will also be an analyst user who will have access to view the data analysis and modify the neighborhood layout. To assign these roles, and preside over all aspects of the web application a manager role will be established.

## **Operating Environment**

*<Describe the environment in which the software will operate, including the hardware platform, operating system and versions, and any other software components or applications with which it must peacefully coexist. In this part, make sure to include a simple diagram that shows the major components of the overall system, subsystem interconnections, and external interface*

*TO DO: As stated above, in at least one paragraph, describe the environment your system will have to operate in. Make sure to include the minimum platform requirements for your system. >*

Parking Tracker will operate only on the Android platform. The minimum required Android version will be 4.0 Ice Cream Sandwich, Android API level 14.

The web application will require PHP 5.2 or greater and a webserver capable of interacting with it. Testing will be performed with Apache 2.2 on Debian 7, however it is assumed that any environment capable of executing PHP, accessing MySQL, and serving webcontent will be capable of this task.

## **Design and Implementation Constraints**

*<Describe any items or issues that will limit the options available to the developers. These might include: hardware limitations (timing requirements, memory requirements); interfaces to other applications; specific technologies, tools, and databases to be used; parallel operations; language requirements; communications protocols; security considerations; design conventions or programming standards (for example, if the customer’s organization will be responsible for maintaining the delivered software).*

*TO DO: In this section you need to consider all of the information you gathered so far, analyze it and correctly identify at least 5 constraints.>*

None to speak of since we are developing everything from the ground up.

## **User Documentation**

*<List the user documentation components (such as user manuals, on-line help, and tutorials) that will be delivered along with the software. Identify any known user documentation delivery formats or standards.*

*TO DO: You will not actually develop any user-manuals, but you need to describe what kind of manuals and what kind of help is needed for the software you will be developing. One paragraph should be sufficient for this section.>*

An online manual with both detailed usage information and an introductory tutorial tailored specifically to the client’s use cases will be provided. Between these two items the users should be able to use and fully understand the software. As questions on usage are asked of us, they can also be compiled into a Frequently Asked Questions list.

## **Assumptions and Dependencies**

*<List any assumed factors (as opposed to known facts) that could affect the requirements stated in the SRS. These could include third-party or commercial components that you plan to use, issues around the development or operating environment, or constraints. The project could be affected if these assumptions are incorrect, are not shared, or change. Also identify any dependencies the project has on external factors, such as software components that you intend to reuse from another project.*

*TO DO: Provide a short list of some major assumptions that might significantly affect your design. For example, you can assume that your client will have 1, 2 or at most 50 Automated Banking Machines. Every number has a significant effect on the design of your system. >*

This project will be dependent on a third party OCR library to be used for reading license plates. It will also be using Google Maps’ API within the Android application and the web application.

It is assumed that the application will be used to collect a full set of data once within each time period designated in order to achieve the desired analysis resolution.

# **Specific Requirements**

## **External Interface Requirements**

### **User Interfaces**

*<Describe the logical characteristics of each interface between the software product and the users. This may include sample screen images, any GUI standards or product family style guides that are to be followed, screen layout constraints, standard buttons and functions (e.g., Cancel) that will appear on every screen, error message display standards, and so on. Define the software components for which a user interface is needed.*

*TO DO: The least you can do for this section is to describe in words the different User Interfaces and the different screens that will be available to the user. Those who will be able to provide optional Graphical User Interface screenshots, will be rewarded by extra marks.>*

### **Hardware Interfaces**

*<Describe the logical and physical characteristics of each interface between the software product and the hardware components of the system. This may include the supported device types, the nature of the data and control interactions between the software and the hardware. You are not required to specify what protocols you will be using to communicate with the hardware, but it will be usually included in this part as well.*

*TO DO: Please provide a short description of the different hardware interfaces. If you will be using some special libraries to communicate with your software mention them here. In case you have more than one hardware interface divide this section into subsections.>*

Any hardware that runs Android ICS and has a camera will work.

### **Software Interfaces**

*<Describe the connections between this product and other specific software components (name and version), including databases, operating systems (Windows? Linux? Etc…), tools, libraries, and integrated commercial components. Identify the data items or messages coming into the system and going out and describe the purpose of each. Describe the services needed and the nature of communications. Identify data that will be shared across software components. If the data sharing mechanism must be implemented in a specific way (for example, use of a global data area in a multitasking operating system), specify this as an implementation constraint.*

*TO DO: The previous part illustrates some of the information you would usually include in this part of the SRS document. To make things simpler, you are only required to describe the specific interface with the operating system.>*

The app will run on an Android system and the incorporated libraries will be Google Maps and the Tesseract OCR engine. Data storage will be done with MySQL through PHP’s PDO binding to MySQL.

### **Communications Interfaces**

*<Describe the requirements associated with any communications functions required by this product, including e-mail, web browser, network server communications protocols, electronic forms, and so on. Define any pertinent message formatting. Identify any communication standards that will be used, such as FTP or HTTP. Specify any communication security or encryption issues, data transfer rates, and synchronization mechanisms.*

*TO DO: Do not go into too much detail, but provide 1-2 paragraphs were you will outline the major communication standards. For example, if you decide to use encryption there is no need to specify the exact encryption standards, but rather, specify the fact that the data will be encrypted and name what standards you consider using. >*

The communications present in this project will be exclusively over HTTP. The Android application will communicate its data to the backend via HTTP requests to a RESTful API. SSL secured HTTP is a possibility, however it will currently require the use of a self-signed certificate and so will not be considered a “safe” connection, and will detract from the image of the software somewhat. Data upload to the API would be possible over a cellular 3G or 4G connection, however in order to minimize, or completely eliminate, the data transfer required over a cellular connection the primary data transfer is expected to happen over a wifi network.

## **Functional Requirements**

*< Functional requirements capture the intended behavior of the system. This behavior may be expressed as services, tasks or functions the system is required to perform. This section is the direct continuation of section 2.2 where you have specified the general functional requirements. Here, you should list in detail the different product functions with specific explanations regarding every function.*

*TO DO: Break the functional requirements to several functional areas and divide this section into subsections accordingly. Provide a detailed list of all product operations related to these functional areas.*

Photograph - The software must have the ability to take a picture, this is required to read the license plates.

Optical Character Recognition - The software must be able to process the picture and extract a license plate number.

Data storage - The software must be able to move all of the captured data to the database.

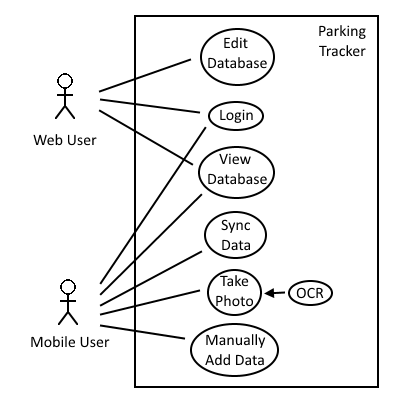
Analysis - The software must have the ability to present a user with information regarding the parking usage over time, and the parking behavior of individual vehicles.

## **Behaviour Requirements**

### **Use Case View**

*<A use case defines a goal-oriented set of interactions between external actors and the system under consideration. Since sometimes we will not be able to specify completely the behaviour of the system by just State Diagrams, we use use-cases to complete what we have already started in section 3.3.1.*

*TO DO: Provide a use case diagram which will encapsulate the entire system and all possible actors. Do not include detailed use case descriptions (these will be needed when you will be working on the Test Plan), but make sure to include a short description of what every use-case is, who are the actors in your diagram. For more information please refer to your UML guide and the MiniThermostat SRS example file.>*



# **Other Non-functional Requirements**

## **Performance Requirements**

*<If there are performance requirements for the product under various circumstances, state them here and explain their rationale, to help the developers understand the intent and make suitable design choices. Specify the timing relationships for real time systems. Make such requirements as specific as possible. You may need to state performance requirements for individual functional requirements or features.*

*TODO: Provide at least 5 different performance requirements based on the information you collected from the client. For example you can say “1. Any transaction will not take more than 10 seconds, etc…>*

While there is no specific performance metric, the app should be quick enough to allow the user to move from car to car efficiently. That is to say. using this app should be faster than writing down each license plate number by hand.

## **Safety and Security Requirements**

*<Specify those requirements that are concerned with possible loss, damage, or harm that could result from the use of the product. Define any safeguards or actions that must be taken, as well as actions that must be prevented. Refer to any external policies or regulations that state safety issues that affect the product’s design or use. Define any safety certifications that must be satisfied. Specify any requirements regarding security or privacy issues surrounding use of the product or protection of the data used or created by the product. Define any user identity authentication requirements.*

*TODO:*

* *Provide at least 3 different safety requirements based on your interview with the client or, on your ABM related research, and again you need to be creative here.*
* *Describe briefly what level of security is expected from this product by your client and provide a bulleted (or numbered) list of the major security requirements.>*

The Android application should be built in such a way as to discourage use of the application while crossing streets. This would entail potentially a warning to not use the device while crossing streets, but also a period of inactivity between the conclusion of mapping a block face and transitioning to a different block.

Security is not a foremost concern of the client, however leaking resident and visitor’s license plate data would never be well received. Standard web application security will be in place for the web application and API. This will include secure password storage, authenticated access, Cross-site request forgery and cross site scripting prevention, and secure data acceptance.

As the use of the Android application will involve taking pictures of cars to indicate the usage of a parking spot, there may be some situation that arise in which a person is in their car or near it and the operator may feel uncomfortable photographing a vehicle at that time. In order to relieve the stress of the situation, it will be possible to manually input a license plate and the presence of a vehicle in a particular stall without actually taking a picture, to be used at the discretion of the operator.

## **Software Quality Attributes**

*<Specify any additional quality characteristics for the product that will be important to either the customers or the developers. Some to consider are: adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability. Write these to be specific, quantitative, and verifiable when possible. At the least, clarify the relative preferences for various attributes, such as ease of use over ease of learning.*

*TODO: Use subsections (e.g., 4.3.1 Reliability, 4.3.2 Portability, etc…) provide requirements related to the different software quality attributes. Base the information you include in these subsections on the material you have learned in the class. Make sure, that you do not just write “This software shall be maintainable…” Indicate how you plan to achieve it, & etc…Do not forget to include such attributes as the design for change. Please note that you need to include at least 2 quality attributes, but it is the mere minimum and it will not receive the full marks.>*

The Parking Tracker needs to be strong in correctness and reliability. This software will be rendered useless if it doesn’t map license plate numbers to parking slots correctly. In the same sense, reliability will be a huge factor. The data correctness should be held from the car to the database and everywhere in between. Also, the app must be reliable in every facet of use. That is, it should not stop functioning while being used and even if there is a crash, the accumulated data should not be lost.

The web application will also focus on correctness in accepting input from the API. It will strive to prevent any erroneous (malformed) data from being accepted into the production data set by verifying the form of each field submitted. This should aid the effort to make the software robust as well. Any errors that can occur should be tested for, caught, and handled before reaching the user directly. This will allow issues to not only be nicely presented to the user, increasing the perceived quality of the software, but will also enable detailed logging to encourage easy debugging and maintenance in the future.

**Appendix A – Data Dictionary**

*<Data dictionary is used to track all the different variables, states and functional requirements that you described in your document. Make sure to include the complete list of all constants, state variables (and their possible states), inputs and outputs in a table. In the table, include the description of these items as well as all related operations and requirements.>*

**Appendix B - Group Log**

*<Please include here all the minutes from your group meetings, your group activities, and any other relevant information that will assist the Teaching Assistant to determine the effort put forth to produce this document>*