Software Requirements Specification

for

Geo-sensing Parking System

Augustine Thorbjornsen

Wujun Song (Carl)

Xuemin Guo (Emma)

Qijing Huang (Joshua)

Content

[1. Introduction 3](#_Toc7611488)

[1.1 Purpose 3](#_Toc7611489)

[1.2 Document Conventions 3](#_Toc7611490)

[1.3 Intended Audience and Reading Suggestions 3](#_Toc7611491)

[1.4 Product Scope 3](#_Toc7611492)

[1.5 References 3](#_Toc7611493)

[2. Overall Description 4](#_Toc7611494)

[2.1 Product Perspective 4](#_Toc7611495)

[2.2 Product Functions 4](#_Toc7611496)

[2.3 User Classes and Characteristics 4](#_Toc7611497)

[2.4 Operating Environment 4](#_Toc7611498)

[2.5 Design and Implementation Constraints 4](#_Toc7611499)

[2.6 User Documentation 4](#_Toc7611500)

[2.7 Assumptions and Dependencies 5](#_Toc7611501)

[3. External Interface Requirements 5](#_Toc7611502)

[3.1 User Interfaces 5](#_Toc7611503)

[3.2 Hardware Interfaces 5](#_Toc7611504)

[3.3 Software Interfaces 5](#_Toc7611505)

[3.4 Communications Interfaces 5](#_Toc7611506)

[4. System Features 5](#_Toc7611507)

[4.1 System Feature 1 5](#_Toc7611508)

[4.1.1 Description and Priority 5](#_Toc7611509)

[4.1.2 Stimulus/Response Sequences 6](#_Toc7611510)

[4.1.3 Functional Requirements 6](#_Toc7611511)

[4.2 Booking function 6](#_Toc7611512)

[4.2.1 Description and Priority 6](#_Toc7611513)

[4.2.2 Stimulus/Response Sequences 6](#_Toc7611514)

[4.2.3 Functional Requirements 6](#_Toc7611515)

[5. Other Nonfunctional Requirements 6](#_Toc7611516)

[5.1 Performance Requirements 6](#_Toc7611517)

[5.2 Safety Requirements 7](#_Toc7611518)

[5.3 Security Requirements 7](#_Toc7611519)

[5.4 Software Quality Attributes 7](#_Toc7611520)

[5.5 Business Rules 7](#_Toc7611521)

[6. Other Requirements 7](#_Toc7611522)

Revision History

None.

1. Introduction (Augustine)

This section gives a scope, description and overview of everything included in this SRS document. Also, the purpose for this document is described and a list of abbreviations and definitions is provided.

1.1 Purpose (Augustine)

The purpose of this document is to give a detailed description of the requirements for the “Geo Sensing Parking System” software. It will illustrate the purpose and complete declaration for the development of system. It will also explain system constraints, interface and interactions with other external applications. This document is primarily intended to be proposed to a customer for its approval.

1.2 Document Conventions (Wujun Song)

This Document was created based on the IEEE template for System Requirement Specification Documents.

1.3 Intended Audience and Reading Suggestions (Xuemin Guo)

This Software Requirements document is intended for:

1. Developers to review the project’s functions then to reuse and improve this project.

2. The end users to receive a guideline to utilize the project appropriately.

3. The testers to test all functions in this software.

1.4 Product Scope (Augustine/Wujun Song)

The “Geo Sensing Parking System” is a GPS-based mobile web application which helps people to find the closest parking based on the user’s current position and automatically book their position once parked. The application should be free to access on all devices.

Furthermore, the software needs both Internet and GPS connection to fetch and display results and a live feed from sensors in the car parks. All system information is maintained in a database, which is located on a web-server. The software also interacts with the Google Maps API so users can view available parking spots on a map and be navigated to them. The application also has the capability of manually pre-booking a parking spot.

1.5 References (Wujun Song)

[1]IEEE Software Engineering Standards Committee, “IEEE Std 830-1998, IEEE Recommended Practice for Software Requirements Specifications”, October 20, 1998.

[2]Davis M A, “Just Enough Requirements Management: Where Software Development Meets Marketing”, New York, Dorset House Publishing, 2005.

[3]Karlsson J, “A Cost-Value Approach for Prioritizing Requirements”, Norges Teknisk- Naturvitenskapelige Uni. 1997

[4]http://www.population.net.au/darwin-population/

[5]http://stat.abs.gov.au/itt/r.jsp?RegionSummary&region=70101&dataset=ABS\_REGIONAL\_ASGS2016&geoconcept=ASGS\_2016&measure=MEASURE&datasetASGS=ABS\_REGIONAL\_ASGS2016&datasetLGA=ABS\_REGIONAL\_LGA2017&regionLGA=LGA\_2017&regionASGS=ASGS\_2016

# 2. Overall Description

2.1 Product Perspective (Qijing Huang)

The Geosensing parking system uses the google API to establish booking system for users. It is a web based system implementing client-server model. The Booking System provides simple mechanism for users to book a slot in advance.

2.2 Product Functions (Qijing Huang)

The users will be able to register and login in the web first. After the users search for the parking space, the results will be shown in a map; the users can choose the parking space by click on the map. A map for the parking space will show the entire available parking slot. The users can choose one of the parking slot and book for it.

When the users parking their car into the slot, the sensor signal turn red into yellow, in the meanwhile, the GPS location will be sent to the system, after the system matches the location of the GPS and slot, it will create a order. When the parking is completed, the signal will turn to red and the payment will be automatically done, the users will get a receipt sending from the system.

2.3 User Classes and Characteristics (Xuemin Guo)

The system has two types of users: web application users and administrators. Each of these two types of users uses the system differently so that each user has its own needs.

In this parking system, the users will be able to search for the parking space, choose one of them in the map and then book the available parking slot. The users will get the receipt of the payment from the system after finished the parking, it will show the start time, end time and total payment.

The administrators also interact only with the web application. They manage the whole system so that there is no wrong information being stored. The administrators may manage information for the web application.

2.4 Operating Environment (Wujun Song)

The Geo-sensing parking system program runs on Windows 10 and Windows 7, for 64-bit/x64 PC architectures. The webpage will be written in HTML, the backend will be written in JAVA, suing Myeclipse. The software will use Oracle for database, using VMware as virtual machine.

2.5 Design and Implementation Constraints (Augustine)

The implementation for this project is heavily depends on the network performance of user-end, because of its web-based architecture. The delay of network will influence the feeling of users. Furthermore, this project needs the negotiation with managers of parking lot. Requiring permission is needed to connect the software with charging system of parking lot and the sensors that detect status of parking.

2.6 User Documentation (Augustine)

The Geo-sensing parking system is user friendly and easy to use, the user can see the bottom such as ‘login’, ‘register’ and ‘search’ clearly.

2.7 Assumptions and Dependencies (Qijing Huang)

The Geo-sensing parking system requires the entire user have GPS in their phone and it allows to access to the Internet.

# 3. External Interface Requirements

3.1 User Interfaces (Xuemin Guo)

When the user opens our website, the login in page will be shown, if it is the first for the user, he or she can do register first by clicking the “sign up”. Then the user will be able to do the register first.

3.2 Hardware Interfaces (Xuemin Guo)

Since the mobile application has no designated hardware, it has no direct hardware interfaces. The physical GPS is managed by the user’s device and the hardware linked to the database server is managed by the underlying mobile and web server operating system.

3.3 Software Interfaces (Augustine)

In order to get geographical information about the location of the user and the view and the information on the parking spots available, the web application communicates with Google Maps API. The communication between the database and the web application consists of read and modify data.

3.4 Communications Interfaces (Wujun Song)

Web application will use HTTP protocol and default port is 80 .Database use TCP protocol and port 1521 as default port which transfers data with web application. Simple Mail Transfer Protocol (SMTP) is used in mail service to send mail with receipt to end user.

# 4. System Features

4.1 System Feature 1 (Wujun Song/Qijing Huang)

### 4.1.1 Description and Priority

High priority: Allowing user to display all near parking lots near to user by click bottom at UI.

### 4.1.2 Stimulus/Response Sequences

When users type the name of target building, all parking lots related to it will be list based on google map

### 4.1.3 Functional Requirements

1. Allow user to input

2. Retrieve related parking lot from database

3. Interact with Google map API

4.2 Booking function (Augustine/Xuemin Guo)

### 4.2.1 Description and Priority

High priority: Allowing user to place a order for booking slot.

### 4.2.2 Stimulus/Response Sequences

By selecting any slot showed on a list, a order will be automatically created on back-end server.

### 4.2.3 Functional Requirements

1. Allow user to select slot

2. Insert new order into database.

# 5. Other Nonfunctional Requirements

5.1 Performance Requirements (Wujun Song)

The login page should not take significant amount of time for operations; it will be good all the operations will be finished in five seconds. There are some example actions that could take significant time:

* Login
* Register
* Locating to the carpark
* Showing available parking slot
* Booking for parking slot

The receipt should be sent to the user within half an hour after the parking is finished.

5.2 Safety Requirements (Qijing Huang)

The operator of database needs to confirm the identity by using username and password.

5.3 Security Requirements (Xuemin Guo)

Users name and password should be stored in a security database.

The Geo-sensing parking system is a multilayered system so that it can do multi-layered defence.

The Geo-sensing parking system could do effective validation for input value to defence SQL Injection.

5.4 Software Quality Attributes (Augustine)

The Geo-sensing parking system will be written in JAVA to insure the system is modular and flexible, it also easy for maintenance and reusable. All naming in Coding part will follow the criteria of Java standard and will be case-sensitive. All files are encoded in UTF-8.

5.5 Business Rules (Qijing Huang)

If the user try to login in with an incorrect user name then the system will alert ‘error’.

If the user finished parking and the email already exists, the receipt will be sent to the email.

# 6. Other Requirements

**Appendix A: Glossary** (Qijing Huang)

|  |  |
| --- | --- |
| **Term** | **Definition** |
| User | Someone who interacts with the web application |
| Admin/Administrator | System administrator who is given specific permission for managing and controlling the system |
| GPS | Global Positioning System |
| GPS-Navigator | An installed software on mobile phone which could provide GPS connection and data, show locations on map and find paths from current position to defined destination |
| Stakeholder | Any person who has interaction with the system who is not a developer. |
| PLAN | The level at which good success can be claimed contained in a language statement |
| DESC | Description |
| RAT | Rational |
| DEP | Dependency |

**Appendix B: Analysis Models**

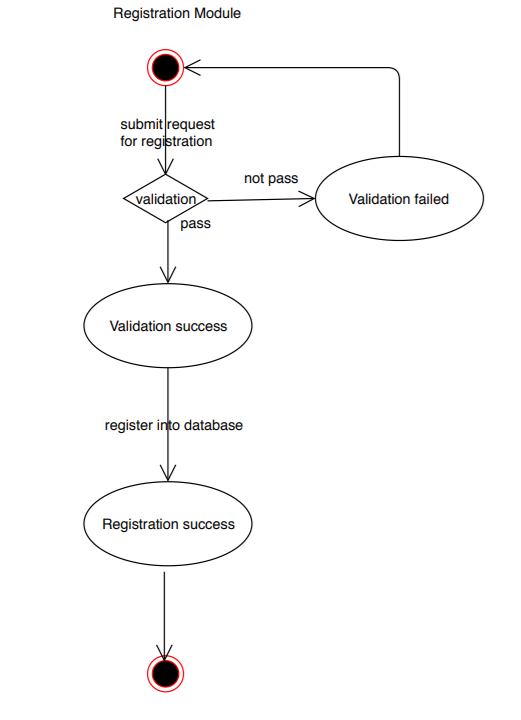


Figure1 Registration Module

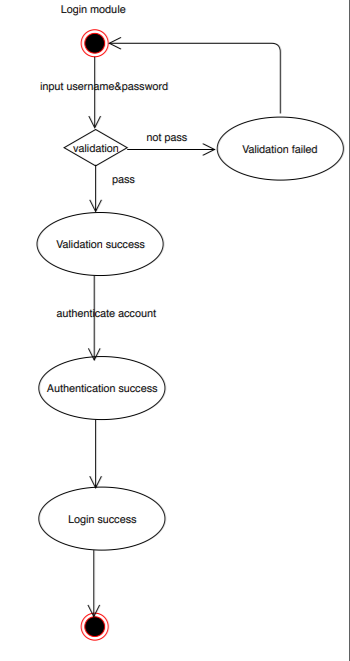


Figure2 Login Module

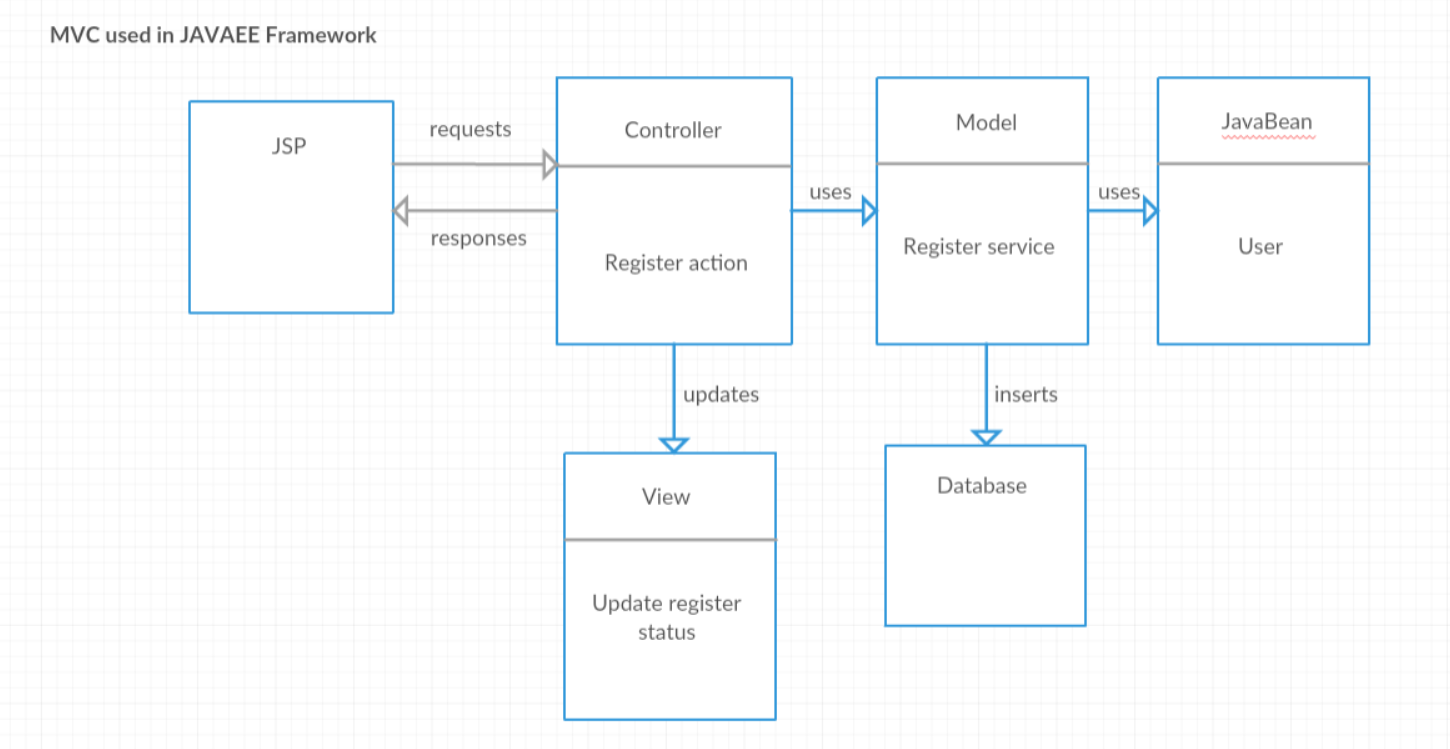


Figure3 MVC Pattern

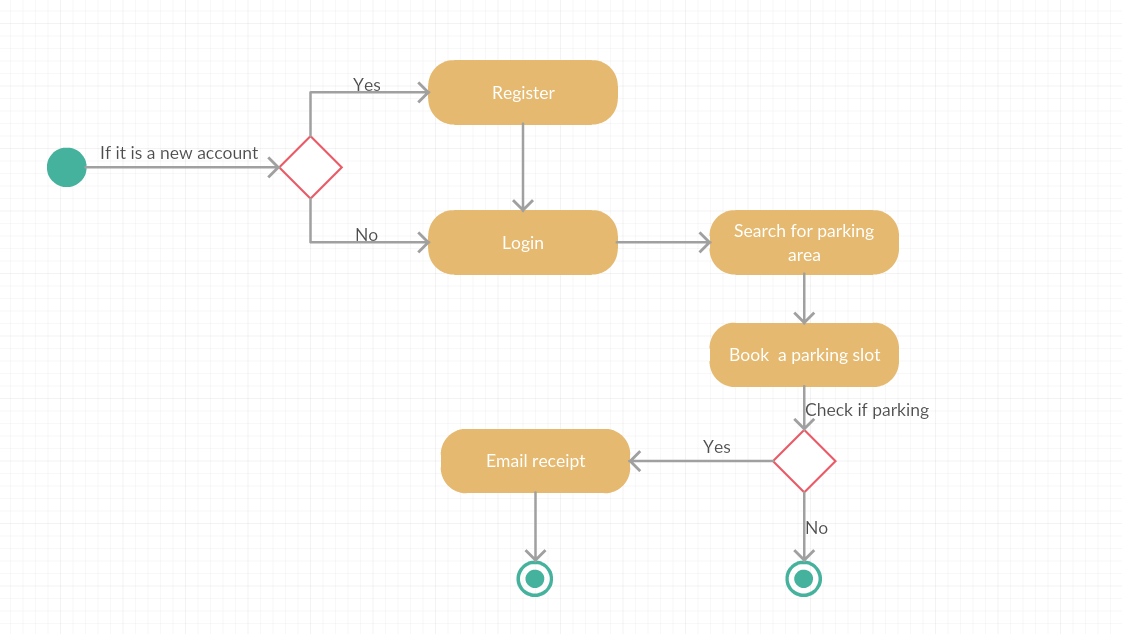


Figure4 UML Activity diagram

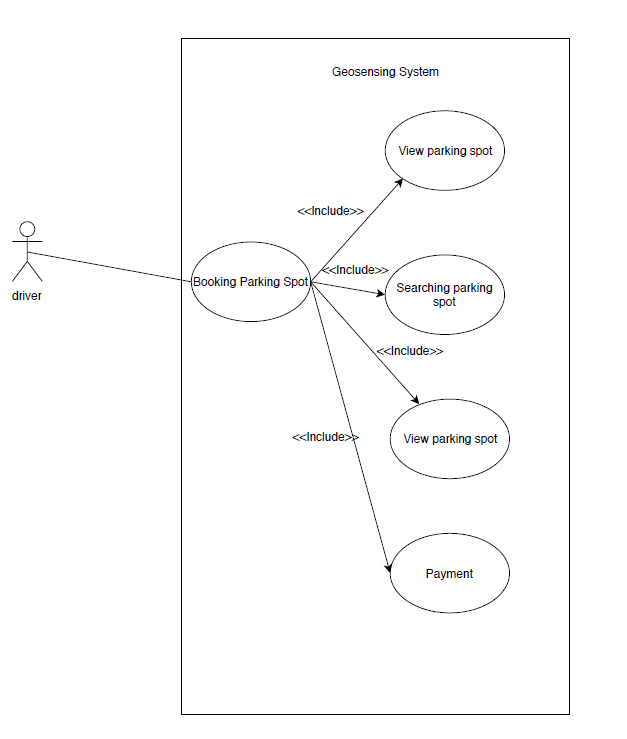


Figure5 UML User case

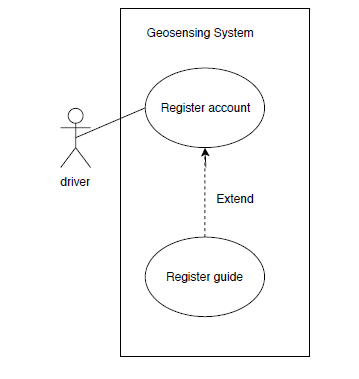


Figure6 UML User case

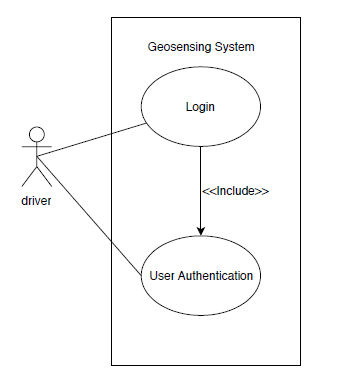


Figure7 UML User case

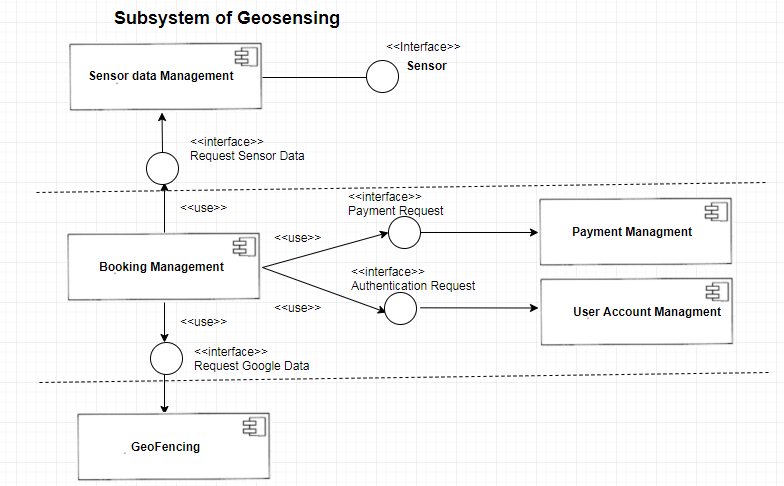


Figure8 Subsystem Model

**Appendix C: To Be Determined List**

None.