

Software Requirements Specification

for

Geo-sensing Parking System

Augustine Thorbjornsen

Wujun Song (Carl)

Xuemin Guo (Emma)

Qijing Huang (Joshua)

Content

[Software Requirements Specification 1](#_Toc10649339)

[for 1](#_Toc10649340)

[Geo-sensing Parking System 1](#_Toc10649341)

[Revision History 2](#_Toc10649342)

[1. Introduction 3](#_Toc10649343)

[1.1 Purpose 3](#_Toc10649344)

[1.2 Document Conventions 3](#_Toc10649345)

[1.3 Intended Audience and Reading Suggestions 3](#_Toc10649346)

[1.4 Product Scope 3](#_Toc10649347)

[1.5 References 3](#_Toc10649348)

[2. Overall Description 4](#_Toc10649349)

[2.1 Product Perspective 4](#_Toc10649350)

[2.2 Product Functions 5](#_Toc10649351)

[2.3 User Classes and Characteristics 6](#_Toc10649352)

[2.4 Operating Environment 6](#_Toc10649353)

[2.5 Design and Implementation Constraints 6](#_Toc10649354)

[2.6 User Documentation 6](#_Toc10649355)

[2.7 Assumptions and Dependencies 6](#_Toc10649356)

[3. External Interface Requirements 7](#_Toc10649357)

[3.1 User Interfaces 7](#_Toc10649358)

[3.2 Hardware Interfaces 10](#_Toc10649359)

[3.3 Software Interfaces 10](#_Toc10649360)

[3.4 Communications Interfaces 11](#_Toc10649361)

[4. System Features (Augustine/Wujun) 11](#_Toc10649362)

[5. Other Nonfunctional Requirements 14](#_Toc10649363)

[5.1 Performance Requirements 14](#_Toc10649364)

[5.2 Safety Requirements 15](#_Toc10649365)

[5.3 Security Requirements 15](#_Toc10649366)

[5.4 Software Quality Attributes 15](#_Toc10649367)

[5.5 Business Rules 16](#_Toc10649368)

[6. Other Requirements 16](#_Toc10649369)

[6.1 Ethical requirements (Qijing/Xuemin) 16](#_Toc10649370)

[6.2 Final requirement 16](#_Toc10649371)

[Appendix A: Glossary (Qijing Huang) 18](#_Toc10649372)

[Appendix B: Analysis Models 18](#_Toc10649373)

[Appendix C: To Be Determined List 27](#_Toc10649374)

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 (Anon, 2019)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]Anon, (2019). [online] Available at: https://www.studocu.com/en/document/addis-ababa-university/software-engineering/mandatory-assignments/ieee-830-1998-standard-srs-document/1981874/view [Accessed 5 Jun. 2019].

[2] Ptgmedia.pearsoncmg.com. (2019). [online] Available at: http://ptgmedia.pearsoncmg.com/images/9780133491296/samplepages/0133491293.pdf [Accessed 5 Jun. 2019].

[3] Robertfeldt.net. (2019). [online] Available at: http://www.robertfeldt.net/courses/reqeng/papers/karlsson\_1997\_cost\_value\_prioritization\_of\_requirements.pdf [Accessed 5 Jun. 2019].

# 2. Overall Description

2.1 Product Perspective (Qijing Huang)

The Geosensing parking system is a web-based application. Users do not need to install an extra app on their mobile phone but only the browser to book a slot on the website. Base on this principle, the smartphone is not compulsory for using this web-based application, which allows most people to benefit from this booking system.

The Booking system needs permission from users to access the GPS part. This booking system uses the users’ location to create the nearest route to the target slot. The google map API is integrated to the JSP page showed on the browser. It provides the location of car parks around users according to their coordinate.

Users’ information is highly sensitive in any application. The database will be separate from the main server and has different authentication. The database contains the users’ username, password, and payment information. All position information about car parks and parking slots also are recorded in the database. The model layer of the application is the only layer in MVC model can access the database. This layer will retrieve the position information then pass them to Google API to create maker of car parks in Google map.

Currently, the coordinate of car parks needs manually input from managers, which is heavy work for a large city. For the future, the software could analyze the coordinate of car parks from Google map.

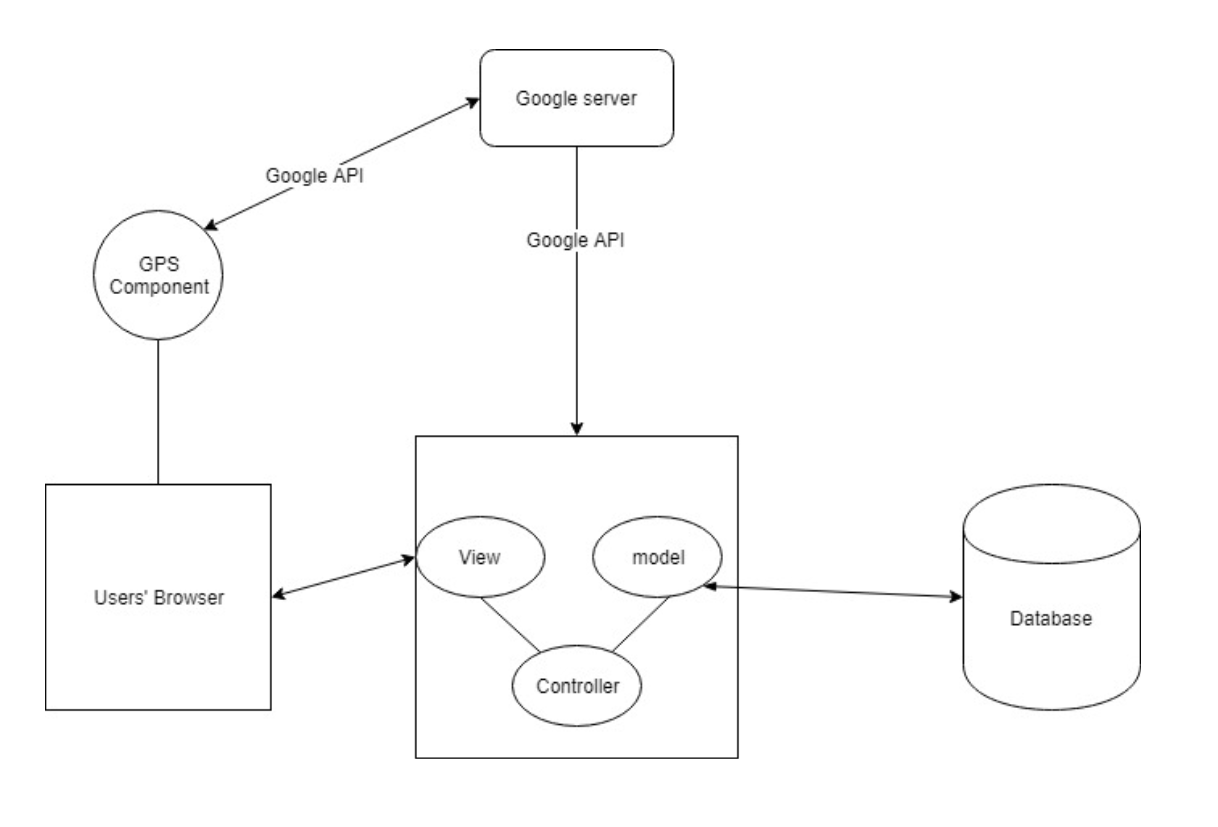


Figure 1 basic part in the software

2.2 Product Functions (Qijing Huang)

Different from some booking software in the market. This product attempts to reduce the typing from users. It replaces typing by selecting to retrieve users’ input. A list view will be displayed in the users’ browser, and users can choose any slot to preserve. Public interest is considered in this system. The web page only lists all car parks around a user rather than all car parks in the database. This rule reduces the situation that users book a slot from where theoretically they cannot arrive to the slot within 2 hours. This system should efficiently serve most people rather than the minor.

The system allows users to choose a specific one slot from a car park because there could be multiple floors car park. All unavailable slots will be marked in booking page for avoiding conflict with others.

A general booking process is running as Figure 2 shown. The users’ input will be validated with the database. Every user is recognized by a unique session ID. Then the user’s location information will be passed by the GPS device to web page. The web page will check nearest car park around the user and list them. Moreover, the navigation function also is integrated in the website. It will lead the user to target car park. Finally, an email with evidence for this booking will sent to the user.

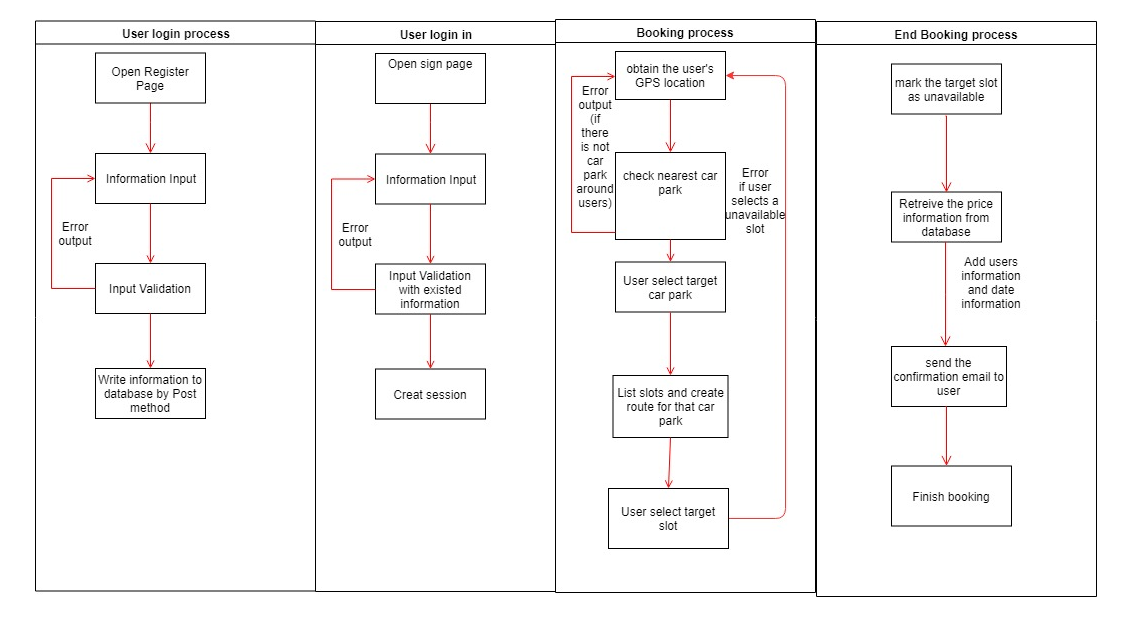


Figure 2 process modeling

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 index page will be shown, if it is the first for the user, he or she can do register first by clicking the “register”. Then the user will be able to do the register first.

If the users already have an account, he or she can sign in by clicking the “sign in” on the right top of the index, and then use their email and password to sign in.

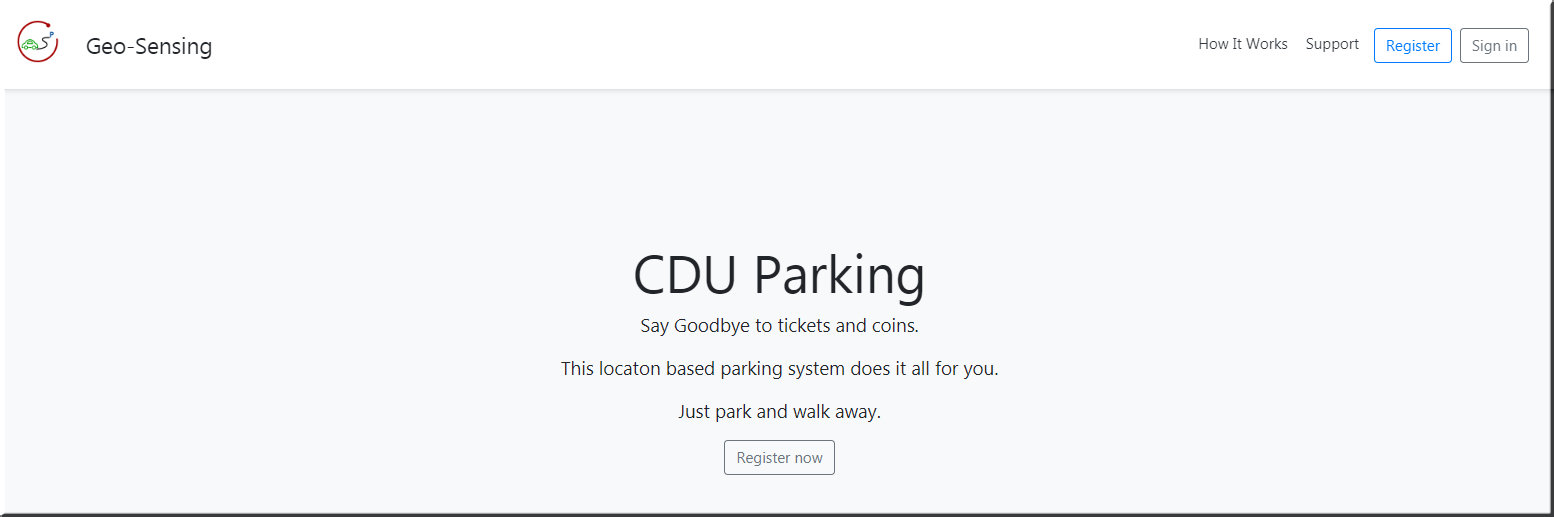


Figure 3: Index page

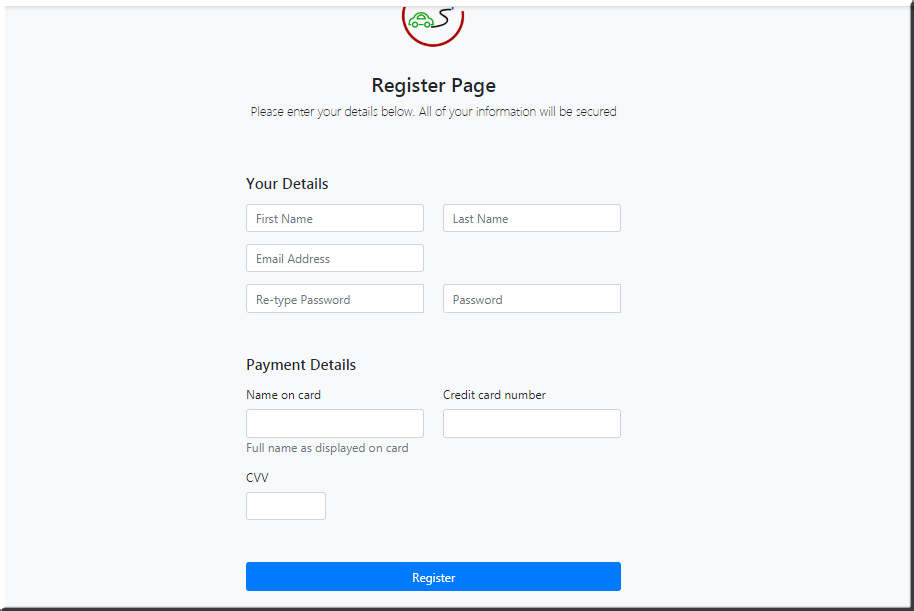


Figure 4: Register page

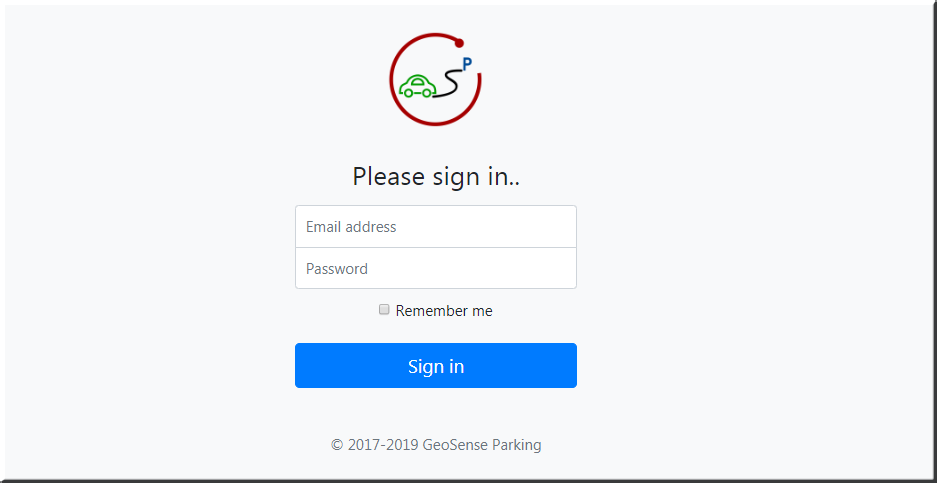


Figure 5: Sign in page

After sign in, the google map of CDU campus will be shown with the Geo fencing, when the user is outside the fence, the system will alert.

If the user is inside the fence, their location will be shown in the map, and the nearest parking station will pop out with a sign, the user can choose that parking station by clicking the “booking”.

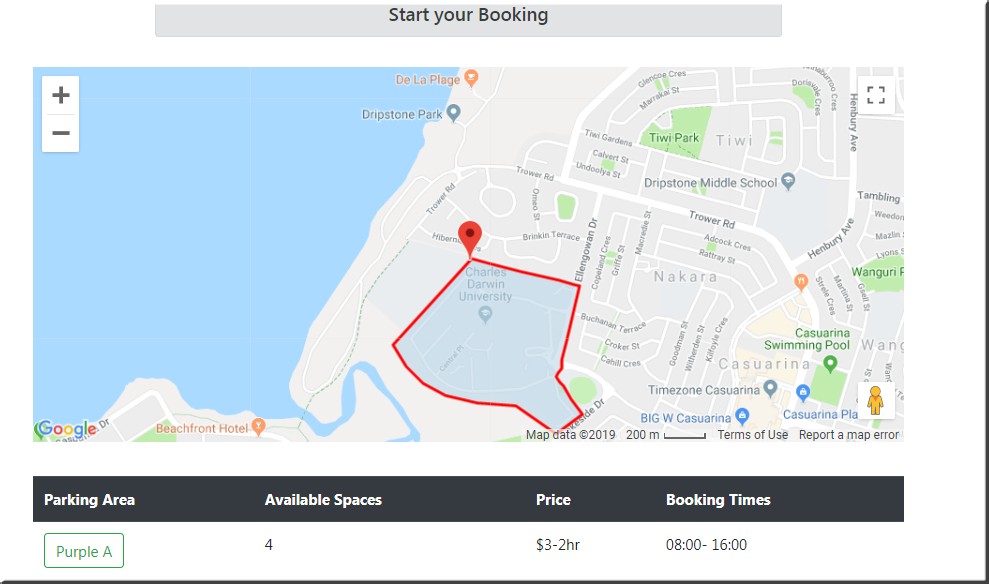


Figure 6: Geo fencing

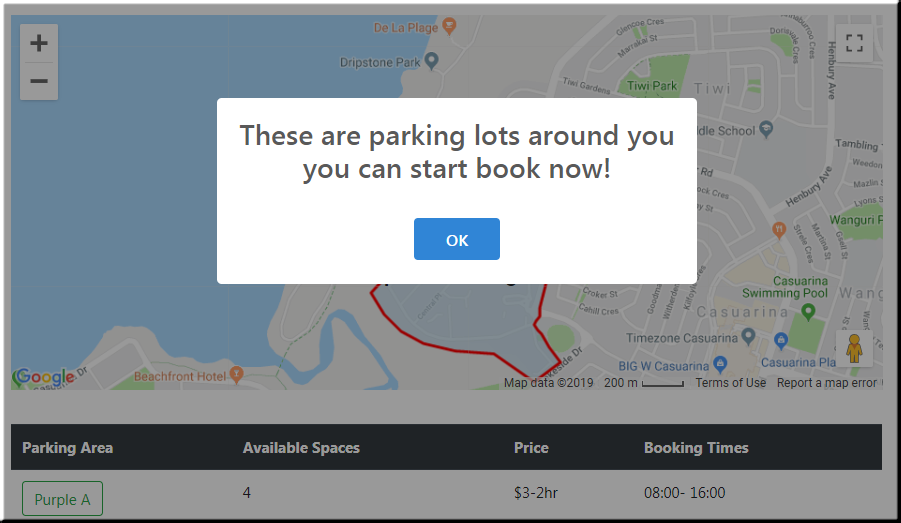


Figure 7: Alert when inside the fencing

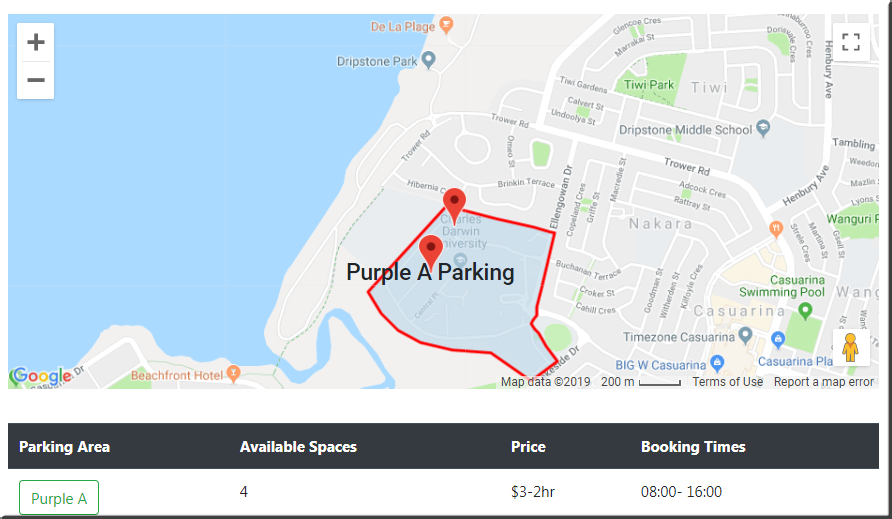


Figure 8: The nearest parking station pop out

After the user click the parking station, all of the available parking slot will be shown in the map, the user can choose one of the available slots they prefer by clicking the “book now” button.

After the user choose the slot, a routing will be shown from where the user is to the parking slot been choose.

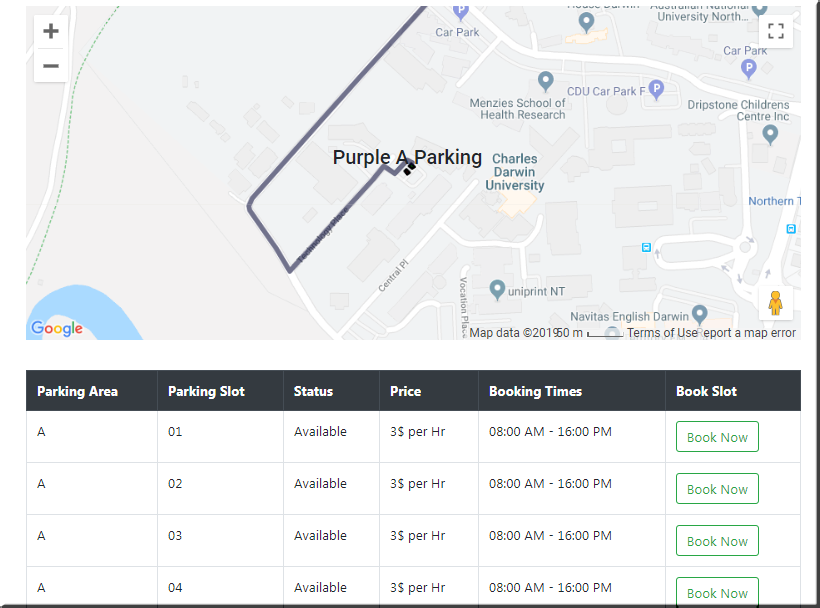


Figure 9: Routing to the parking slot

After the parking is successfully finished, an email will be sent to the user with the receipt, the receipt contains the start time and the end time of the parking, the amount this parking cost.

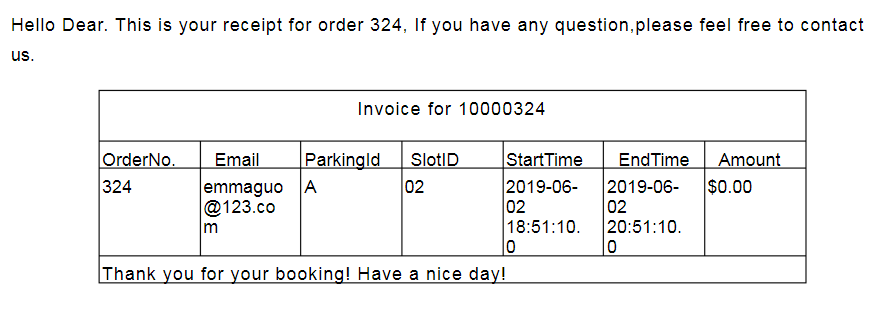


Figure 10: Receipt from e-mail

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 8080. 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.

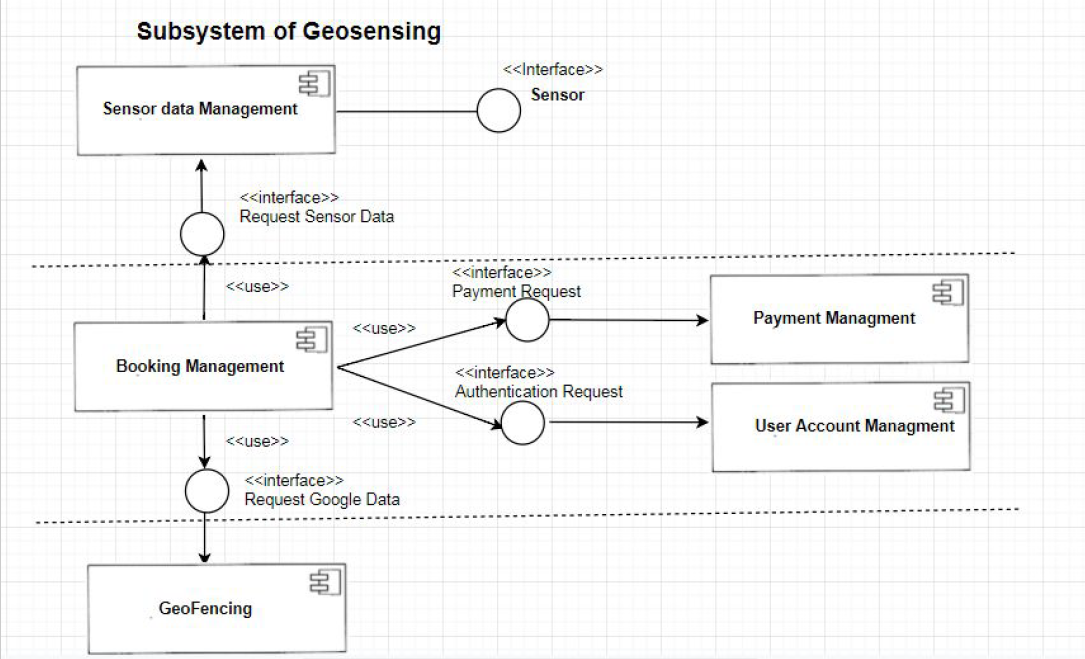


Figure 11: Subsystems Model

# 4. System Features (Augustine/Wujun)

**ID: FR1**

TITLE: User access – Web application

DESC: The user has to have access to the internet, and then the user can access the web application.

RAT: In order for a user to access the web application.

DEP: None

**ID: FR2**

TITLE: User registration – Web application

DESC: Given that a user has accessed the web application via a browser, then the user should be able to register. The user must provide email address, password and credit card details.

RAT: In order for a user to register on the mobile application.

DEP: FR1

**ID: FR3**

TITLE: User login – Web application

DESC: Given that a user has registered, then the user should be able to log in to the web application. The log-in information will be stored in the database.

RAT: In order for a user to login into the web application.

DEP: FR2

**ID: FR4**

TITLE: Google Map – Web application

DESC: When the user has logged into the web application, the user should be able to see the google maps that is zoomed into the CDU campus. The user needs to accept to share location.

RAT: In order for a user view the Google maps api inside the web application.

DEP: FR3

**ID: FR5**

TITLE: Show available parking areas

DESC: When the user is logged into the web application and drives into CDU campus, a pop up shows the available parking areas.

RAT: In order for a user view the available parking areas.

DEP: FR3, FR4

**ID: FR6**

TITLE: Routing to parking area

DESC: When the user selects an available parking arear, an automated routing system will direct him to the parking area.

RAT: In order for a user route to parking area.

DEP: FR5

**ID: FR7**

TITLE: Show available parking slots

DESC: After selecting a parking area, the table should render to show available parking slots and slots that are unavailable.

RAT: In order for a user show the available parking slots.

DEP: FR6

**ID: FR8**

TITLE: Mobile Friendly

DESC: The web application should work on all platforms including mobile devices.

RAT: In order for the application to render successfully.

DEP: FR1

**ID: NFR1**

TITLE: Multi-layered defense

DESC: The web application should prevent attacks from all possible areas including front end and back end attacks.

RAT: In order for the application to keep user data safe.

DEP: None

**ID: NFR2**

TITLE: Prevention from Cross site scripting

DESC: The web application should prevent cross site scripting, so all user data is kept private.

RAT: In order for the application to keep user data safe.

DEP: NFR1

**ID: NFR3**

TITLE: Prevention from SQL Injection

DESC: The web application should prevent SQL Injection attacks and store user data in database with validation scripts.

RAT: In order for the application to keep user data safe.

DEP: NFR1

# 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)

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

2. The password of the database should be encrypted by using HASH+SALT

3. All the user information should be stored into the secured database.

4. The Geo-sensing parking system must pass Unit Test, Integration Test.

5.3 Security Requirements (Xuemin Guo)

**Non-functional requirements of Geo-fencing:**

1. The Geo-sensing parking system is a multilayered system so that it can do multi-layered defense in MVC pattern, both on front end and back end.

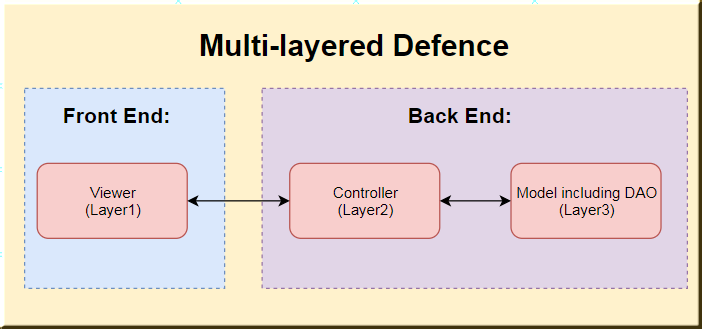


Figure 10: Multi-layered Defense diagram

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

3. The Geo-sensing parking system could do effective validation for input value to defense No Cross site scripting attacks or forgery

4. The Geo-sensing parking system must pass the Security Test via SQLMAP

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)

* Every slot should only be allocated to one user.
* The email of every user should not be repeated.
* Any user only can occupy one slot at one time.
* Payment information is compulsory for all users.
* Car parks only is displayed when users near to them.
* 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

## 6.1 Ethical requirements (Qijing/Xuemin)

PUBLIC - Software engineers shall act consistently with the public interest.

Software engineers may violate the PUBLIC code of ethics, he/she may want to develop the software as his/her wish according to personal interest. For example, he/she might leave a loophole in the system on purpose so that he/she could use the system without causing any fee in the future.

To solve this problem, the manager of the team should promote an ethical approach to educate team member to follow the software specification and finish all tasks with high professional standard according to the public interest.

During the development of the system, we should try our best to make sure our products meet the highest professional standards depending on the ethical principles. For example, we may meet some problems such as miscalculating the location of the slot, we need to solve this problem so that our client can see the exact location.

In our project, we need ensure the users cannot book one parking spot for unlimited time. Otherwise, long time to occupy a slot without using it could be a potential damage to public interest.

## 6.2 Final requirement (Wujun/ Augustine)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| User Story | Functional Requirement & priorities | | | |
|  | High Priority | Medium Priority | Low Priority | No Priority |
| 1.As a client , I want security test to ensure that System could prevent SQL Injection . | Ensure there must be fundamental way like escape special character to filter SQL injection | Add @ to make attackter be blind |  |  |
| 2.As a client , I want security test to ensure that System could prevent Cross site scripting attacks or forgery . | Apply basic str\_replace() to remove all script() from input | use htmlspecialchars()to escape all special character | do not reflect any input to user |  |
| 3.As a client , I want multilayered system so that the system could have a multi-layered defence. | Separate database from server |  | set VPN for every users |  |
| 4.As a end user(driver) , I want the system could show available slots in each area so that slots could be easily found. | give number of available slots for all parking | Show detail of available slots in every parking lot | give micro map to every parking slot | present status of every slots in each parking slot |
| 5.As a end user(driver) , I want 2 hours free period .The parking time starts from 8 a.m-4:30 p.m. | Every parking lots need be informed by email from server | a ticket will be contained in order to get free 2 hours | first 2 hours will be automatically deducted from total parking time |  |
| 6.As a end user(driver) , I want an web-based mobile friendly application so that I can visit it also from mobile phone. | Application will based on web but not other custom function for mobile | BootStrap will be used to adapt mobile screen | Mobile network quality will be consider in network design | Every type mobile will get custom version for best experience |
| 7.As a client , I want an system integrated with Google Map so that the system could provide Google service as well . | Google navigation will be intergarted only for GPS function . | electronic fence surrounding slot will be used to detect the order progress |  |  |
| 8.As a client , I want an registration details including phone number and bank details so that user could make a payment. | Register Page will contain form to input bank detail | bank detail will be stored separately for security | A bank card camara scanner will be intergrated to register page, which allow users input bank detail conveniently. |  |

# Appendix A: Glossary (Qijing Huang)

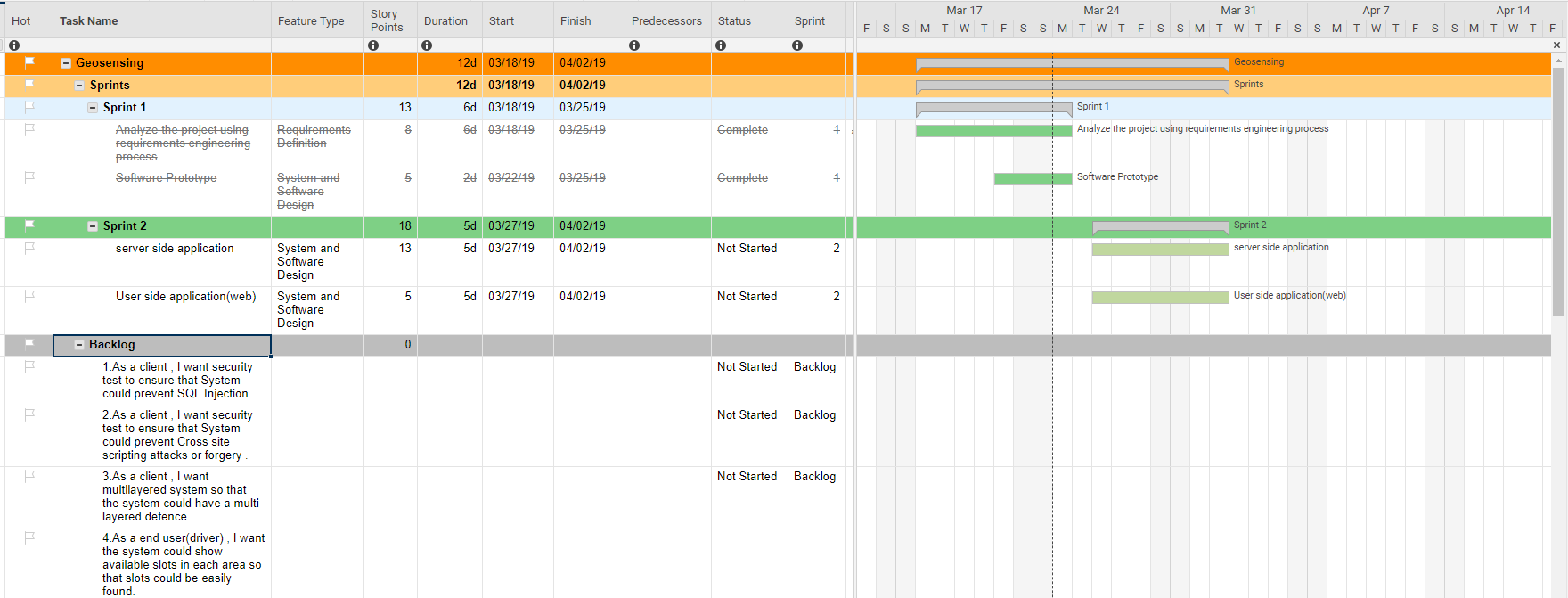
|  |  |
| --- | --- |
| **Term** | **Definition** |
| User | Someone who interacts with the web application |
| Admin/Administrator | System administrator who is given specific permission for managing and controlling the system |
| GPS | Global Positioning System |
| GPS-Navigator | An installed software 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

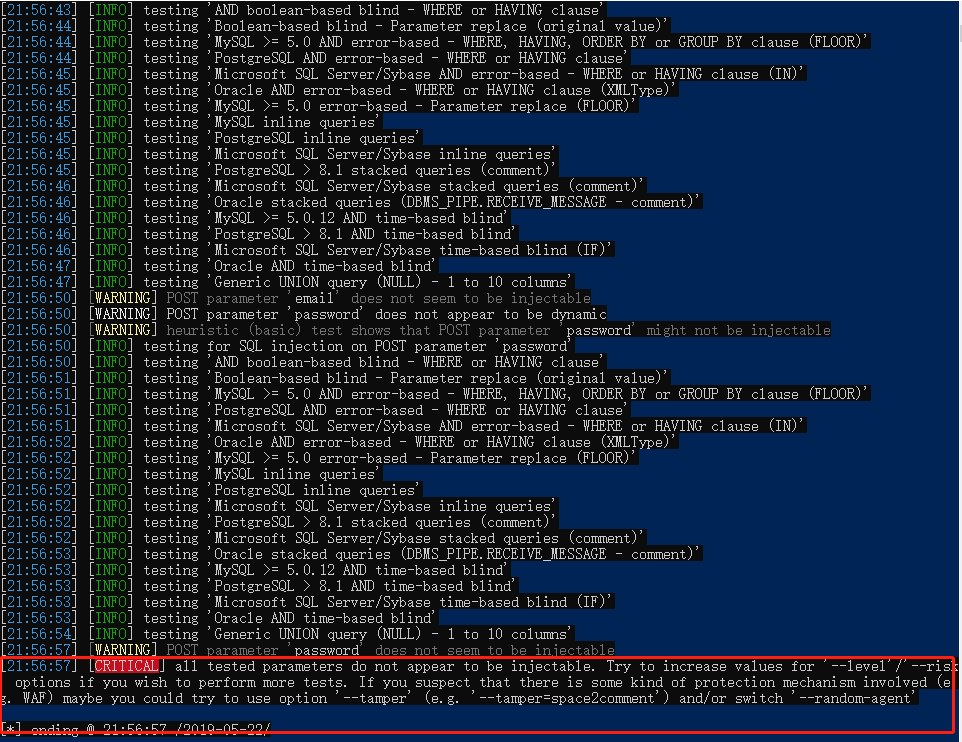
Milestones – Augustine/Xuemin

|  |  |  |
| --- | --- | --- |
| **Project Milestone** | **Initial Estimated Completion Date** | **Actual Completion Date** |
| Meet with Client | 18/03/2019 | 18/03/2019 |
| Paper Prototype | 25/03/2019 |  |
| Finalizing software requirements | 25/04/2019 |  |
| Finalizing the software specifications | 26/04/2019 |  |
| Demo of the designed system | 05/05/2019 |  |
| Usability Testing | 10/05/2019 |  |
| Report of development | 28/05/2019 |  |
| Report of Testing(function & security) | 31/05/2019 |  |
| Delivery result Report | 01/06/2019 |  |

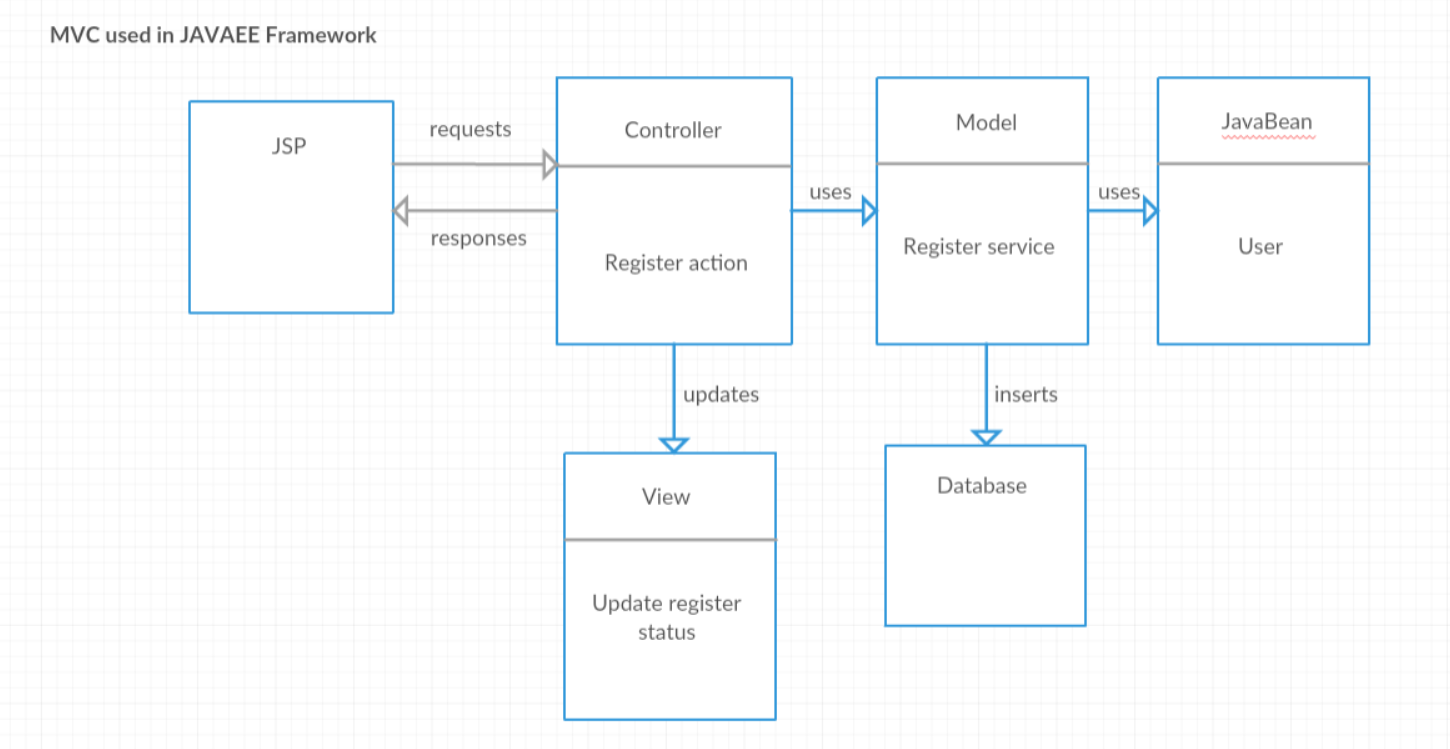
Project Plan (Gantt Chart or Sprint Planning) – Qijing Huang



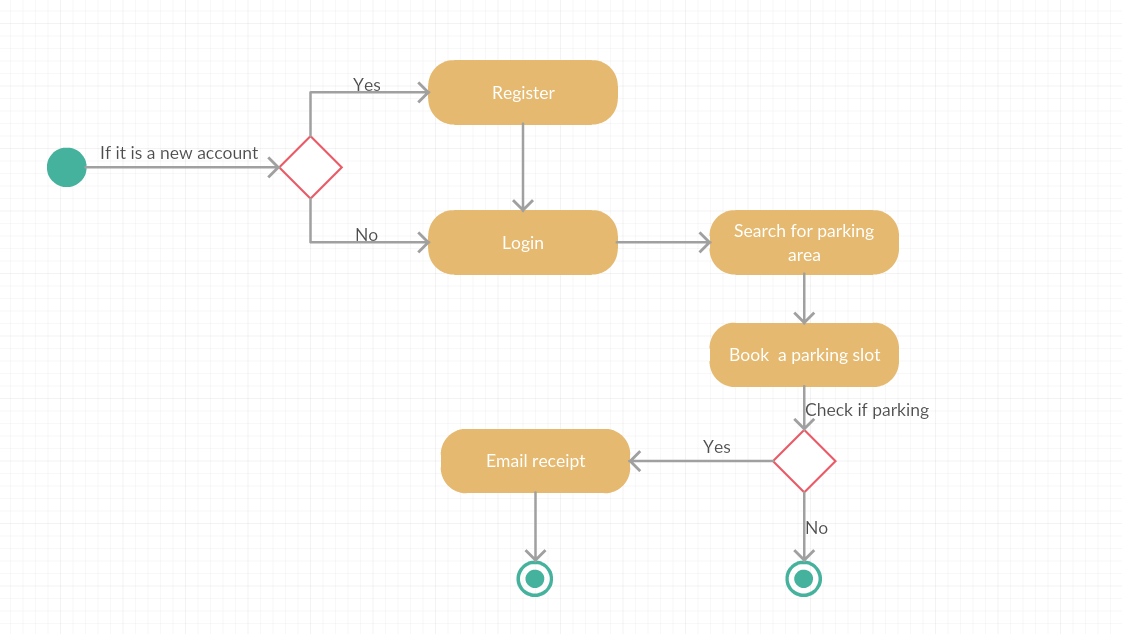
Security Test – SQL Map – Wujun Song



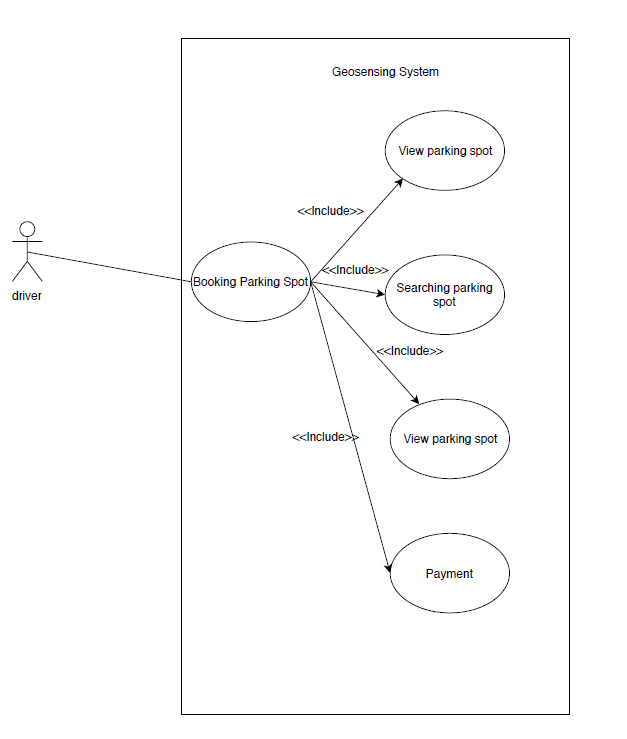
MVC Pattern – Xuemin Guo

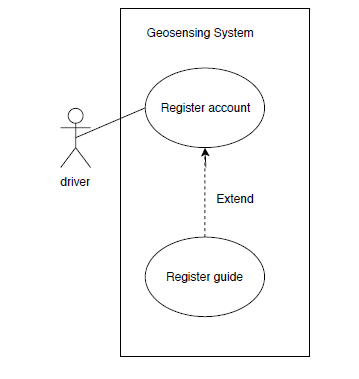


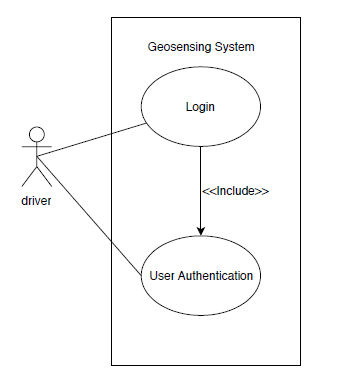
UML Activity diagram - Augustine



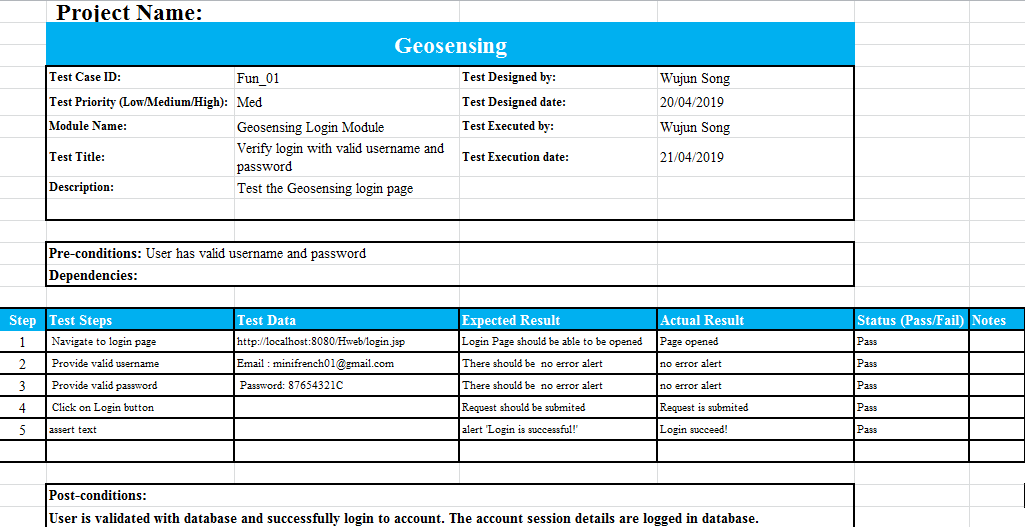
UML User case –Qijing Huang



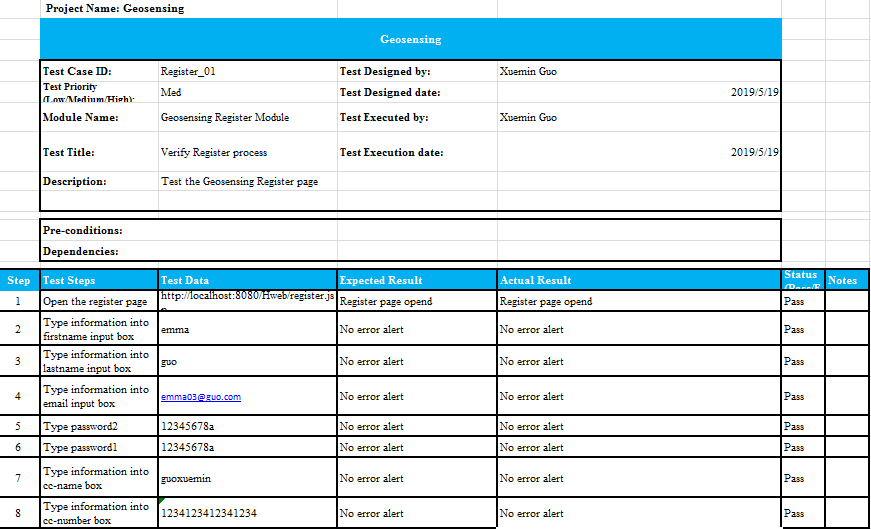




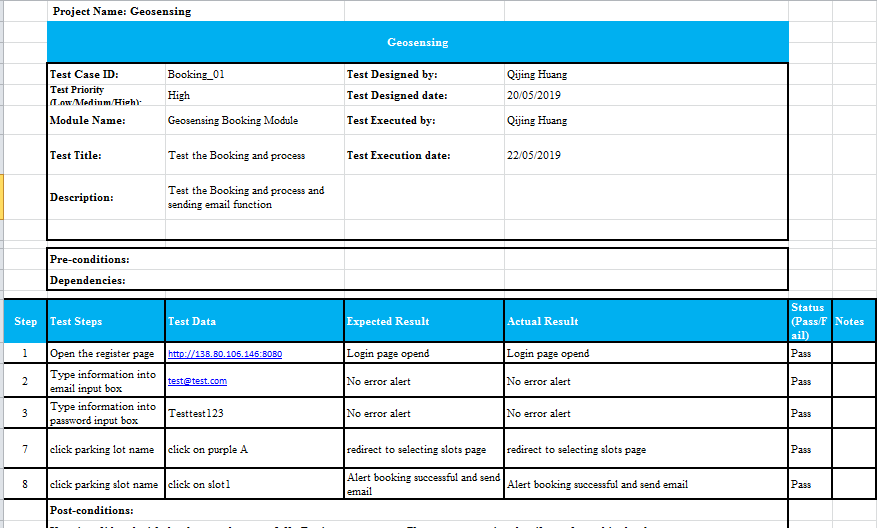
Login Test Case – Wujun Song



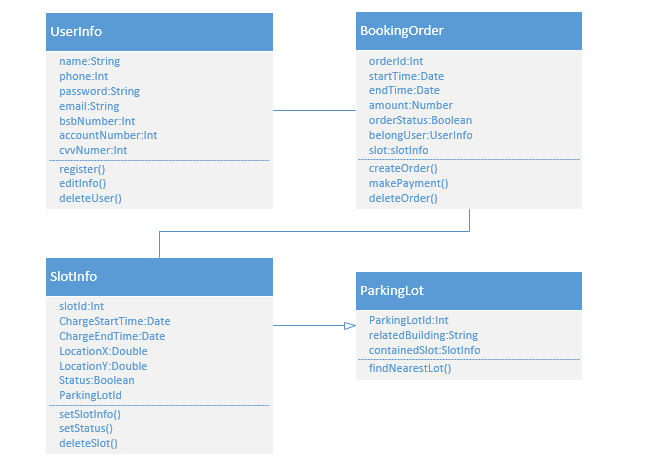
Register Test Case – Xuemin Guo



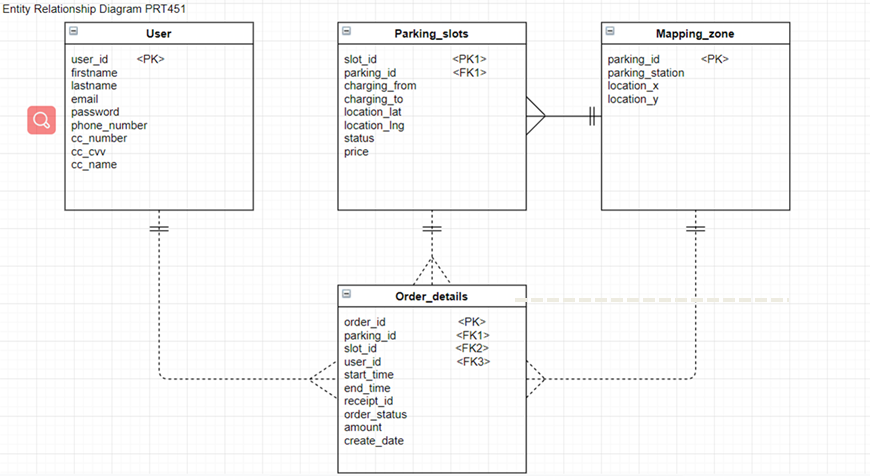
Booking Test Case – Qijing Huang



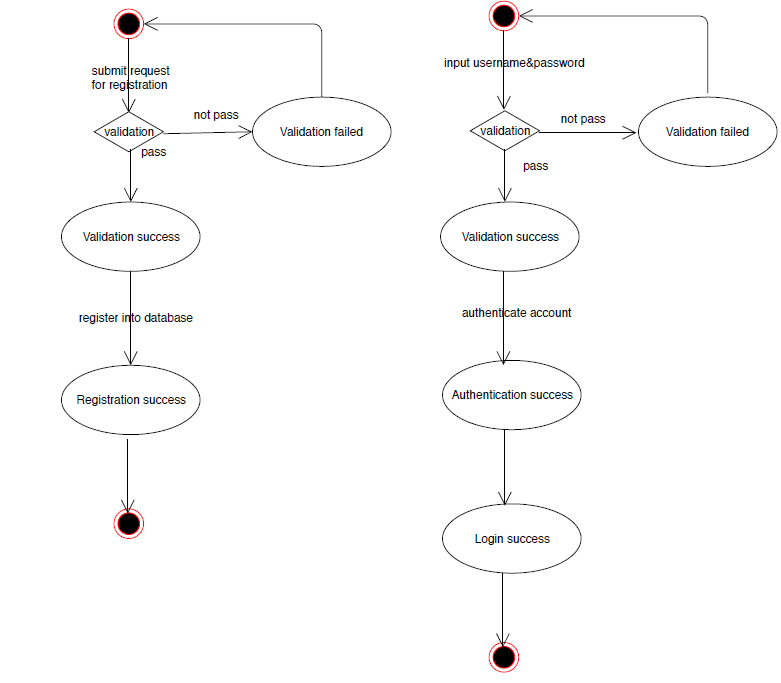
Class diagram - Augustine



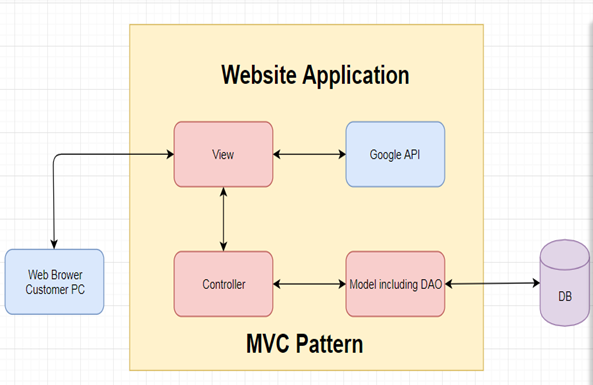
Entity - Relationship Diagram – Xuemin Guo



State Diagram – Qijing Huang



Software Architecture – Wujun Song



# Appendix C: To Be Determined List

Improvement of UI

