## PRT451

## Project Smart Skips

## Software Requirements Specification

## Group Xeon

|  |  |
| --- | --- |
| Joel Benesha | s280615 |
| Yinrui Kang | s299806 |
| Chao Liang | s304935 |
| Yuze Li | s306890 |
| Yuchu Han | s300408 |

**Table**

[1. Introduction 3](#_Toc515491407)

[1.1 Purpose 3](#_Toc515491408)

[1.2 Scope 3](#_Toc515491409)

[1.3 Milestones 3](#_Toc515491410)

[1.4 Project plan 3](#_Toc515491411)

[1.5 User Story 5](#_Toc515491412)

[1.6 Scrum – Xeon group 5](#_Toc515491413)

[1.7 Overview 6](#_Toc515491414)

[1.8 References 7](#_Toc515491415)

[2. Overall description 8](#_Toc515491416)

[2.1 Product perspective 8](#_Toc515491417)

[2.2 Product functions 8](#_Toc515491418)

[2.3 User characteristics 8](#_Toc515491419)

[2.4 Constraints 9](#_Toc515491420)

[2.5 Assumptions and dependencies 9](#_Toc515491421)

[2.6 Interface design 9](#_Toc515491422)

[2.7 version control 10](#_Toc515491423)

[2.8 Continuous Integration 11](#_Toc515491424)

[2.9 Map functions 12](#_Toc515491425)

[3. Specific requirements 14](#_Toc515491426)

[3.1 Functional requirements 14](#_Toc515491427)

[3.2 Non-functional requirements 18](#_Toc515491428)

[4. UML patterns 22](#_Toc515491429)

[5. MVC patterns 24](#_Toc515491430)

[6. Database 27](#_Toc515491431)

[6.1 SQLite database 27](#_Toc515491432)

[6.2 database design 27](#_Toc515491433)

[7. Security measures 29](#_Toc515491434)

[7.1 security in client sever 29](#_Toc515491435)

[7.2 Authorization certification 31](#_Toc515491436)

[7.3 White list 32](#_Toc515491437)

[7.4 post message 33](#_Toc515491438)

[7.5 Error page 33](#_Toc515491439)

[7.6 MD5 34](#_Toc515491440)

[7.7 Login analysis 35](#_Toc515491441)

[8. Unit test 36](#_Toc515491442)

# Introduction

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.

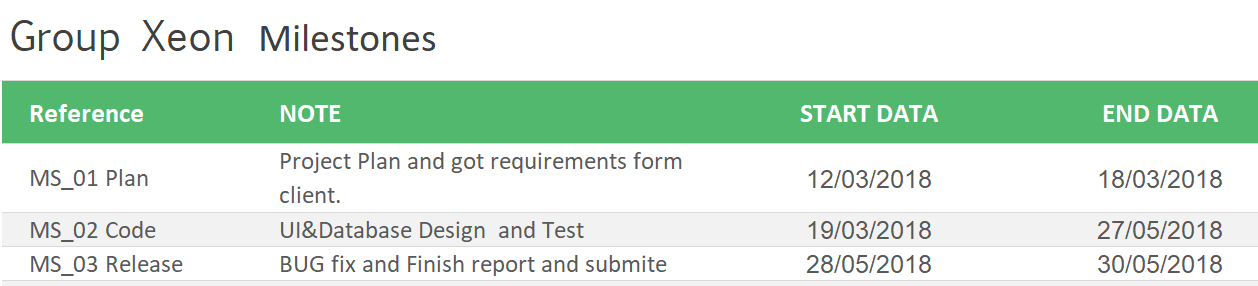
### 1.1 Purpose

The purpose of this document is to give a detailed description of the requirements for the “Smart Skip” android APP. 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 and a reference for developing the first version of the system for the development team.

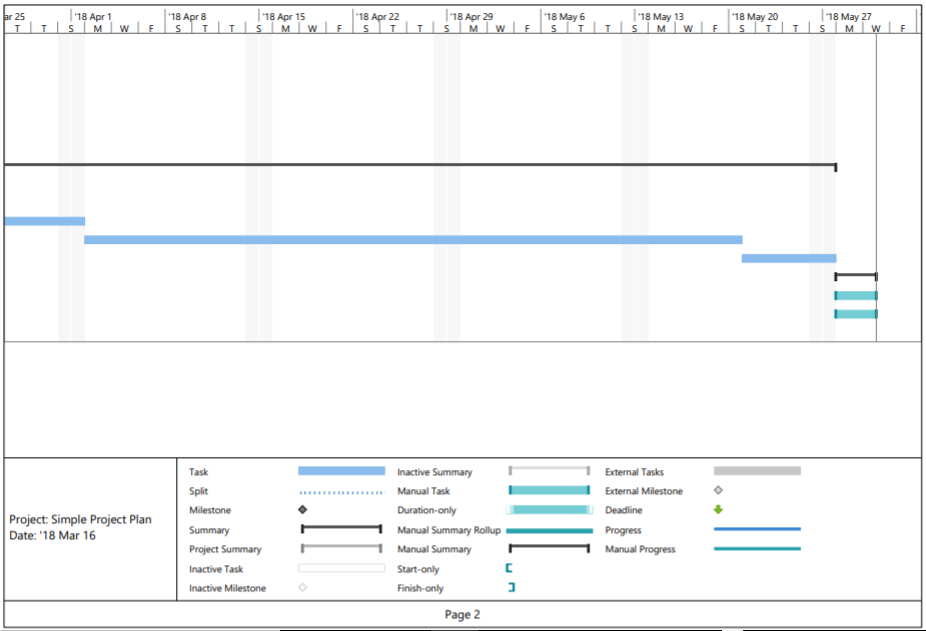
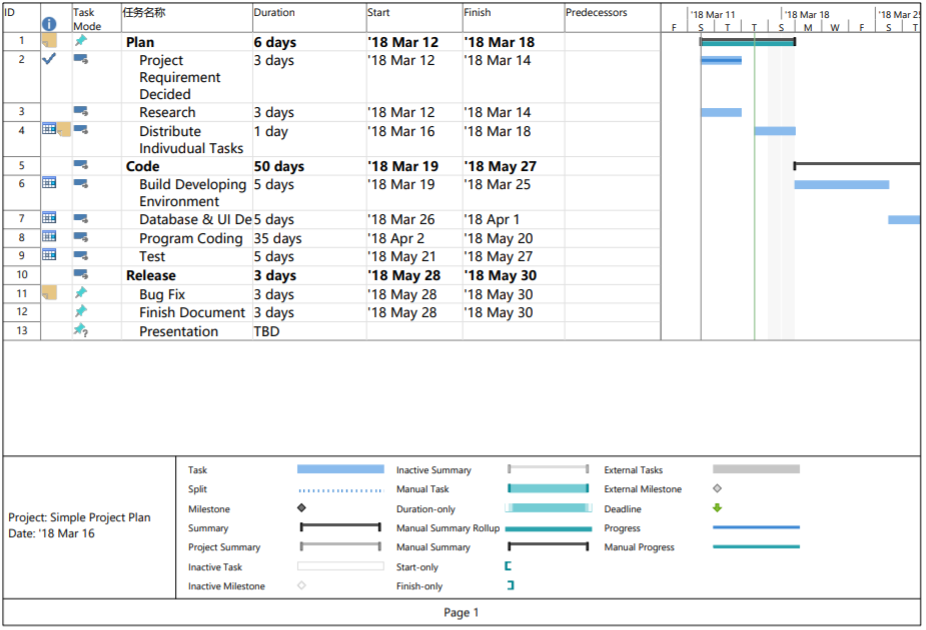
### 1.2 Scope

Skips are rubbish bins collected by the city council from constructions and commercial sites. These skips are sometimes full and need to be emptied. The objective of the project is to develop a smart skip sensor which sends signals to a Mobile application. The intention of the app is to detect whether a skip is full or empty. The result obtained from the sensor will determine whether the skip needs to be emptied or not. The application will notify the users in the process and Alert landfill operators about the skip load. The design will incorporate route mapping and if full the skip will be taken to different destination depending on it contains. The skip with concrete needs to be taken to the concrete crushers and the skips with mixed waste to the landfill. The application will provide the shortest route from source to the intended destination of the skip through implemented features such as Google map. The project will not focus on a physical component but will use dummy database to achieve the task. Furthermore, the project will not address any other feature or issue and will focus on developing an intelligent routing for smart skips.

### 1.3 Milestones



### 1.4 Project plan



### 1.5 User Story

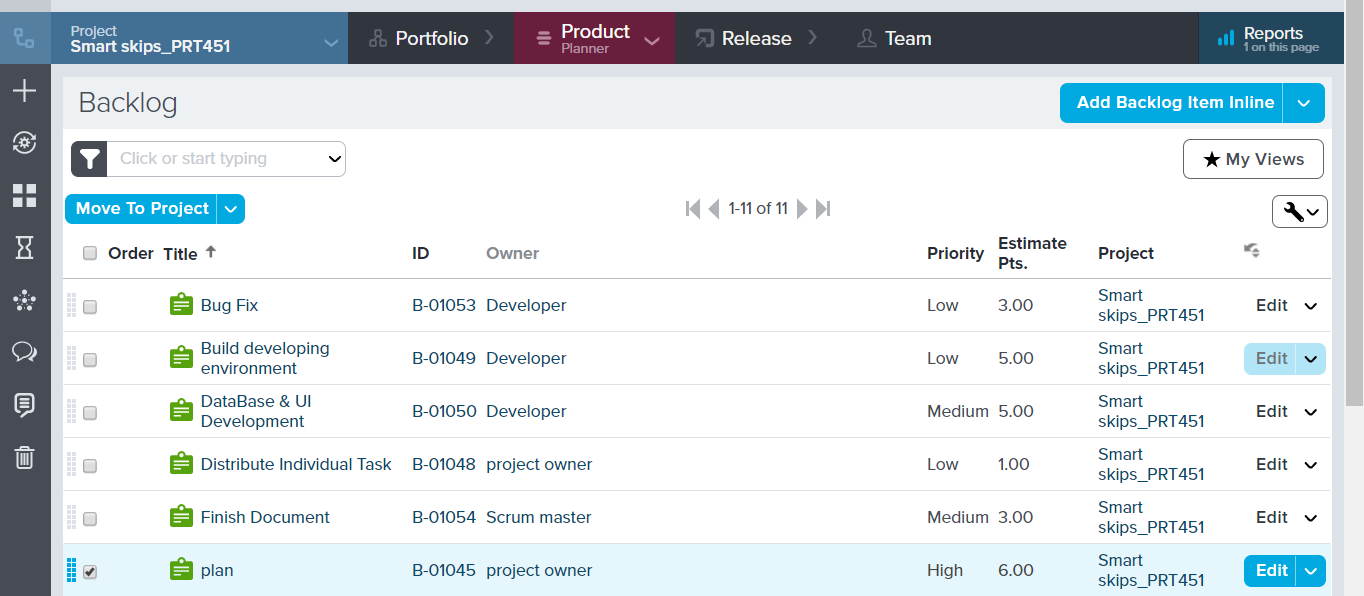
As a truck driver, I would like a simple application to be able to notify me if the skip is full. The application should have route mapping for skips, finding the shortest path from the source to the destination. It should also alert the landfill operator about the skipload. The application would allow me to work effectively and efficiently.

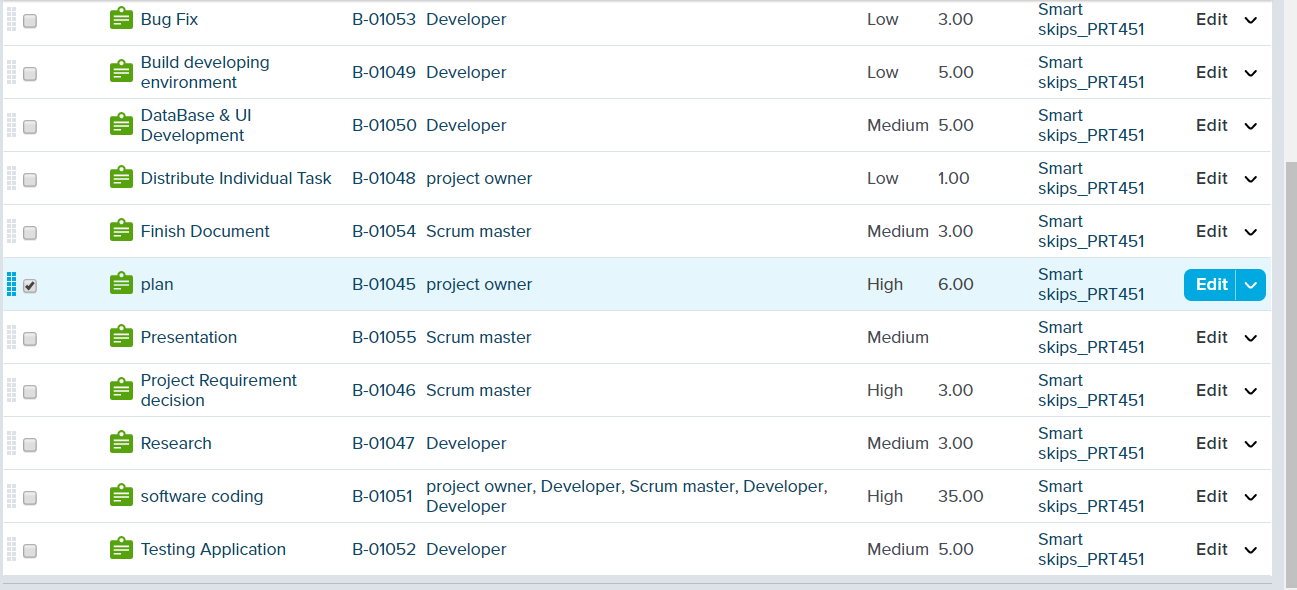
### 1.6 Scrum – Xeon group

Link:<https://www51.v1host.com/CharlesDarwinUniversity87/Default.aspx?menu=PrimaryBacklogPage>

Account detail:

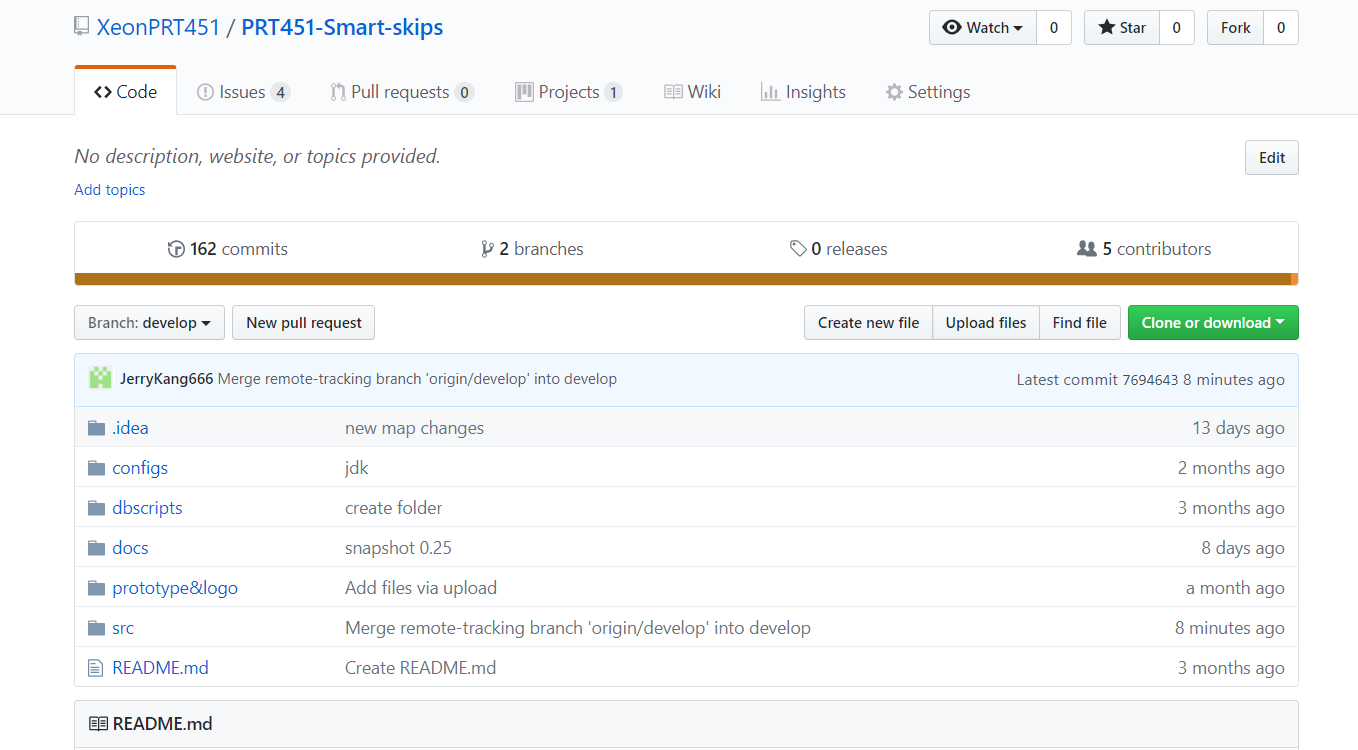
* User detail: admin
* Password: admin





### 1.7 Overview

This document consists of eight chapters.This article provides an overview of system functions and interactions with other systems, as well as different type of **users** and their interactions with the system.In addition, system constraints and assumptions about products are mentioned.Detailed requirements specifications and descriptions of different system interfaces are provided. Different specification techniques are used to specify requirements for different users more accurately. The article also introduces UML design and NVC design, database design, unit testing and a series of security measures.

Group members use Github to develop this project. Link: https://github.com/XeonPRT451/PRT451-Smart-skips

### 1.8 References

[1] Rajam, S., Cortez, R., Vazhenin, A., & Bhalla, S. (2011). Modified MVC-design patterns for service oriented applications. Frontiers in Artificial Intelligence and Applications, 231, 108-126.

[2] Le, T. Q., & Pishva, D. (2015). Optimization of convenience stores' distribution system with web scraping and Google API service. International Conference on Advanced Communication Technology, ICACT, 2015, 596-606.

[3] Deutz, P., Neighbour, G., McGuire, M., & Lyons, Donald I. (2010). Integrating sustainable waste management into product design: Sustainability as a functional requirement. Sustainable Development, 18(4), 229-239.

[4] Jie, J. (2016). Industrial Case Study of Transition from V-Model into Agile SCRUM in Embedded Software Testing Industries. ACM SIGSOFT Software Engineering Notes, 41(2), 1-3.

[5] http://www.androidbegin.com/tutorial/android-ormlite-with-sqlite-database-tutorial/

[6] https://stackoverflow.com/questions/2938502/sending-post-data-in-android

[7] http://ormlite.com/javadoc/ormlite-core/doc-files/ormlite\_9.html#License

[8] http://hibernate.org/community/license/

[9] http://www.apache.org/licenses/LICENSE-2.0

# Overall description

This section will give an overview of the whole system. The system will be explained in its context to show how the system interacts with other systems and introduce the basic functionality of it. It will also describe what type of stakeholders that will use the system and what functionality is available for each type. At last, the constraints and assumptions for the system will be presented.

### 2.1 Product perspective

The mobile application will be used to find the status and location of the trash can and the garbage station, and to view the user's personal information and work status, including administrator and driver. The program needs to integrate the GPS navigation function of the Google map to report the status of the garbage can by sending signals through the sensors embedded in the garbage. GPS will provide drivers with the location of garbage dumps and trash cans, provide the shortest route and the distance between them to navigate and display on the map. Both drivers and administrators can view historical records for more convenient work, and can view individual work status, but drivers cannot modify personal information.

The user can login and logout, and the logout system will clear the history to ensure the security of the account.

### 2.2 Product functions

With the mobile application, the users will be able to see the state of every skip (red: full, green: empty state) on the app and search for garbage. The result will be based on the criteria the user inputs. There are several search criteria and it will be possible for the administrator of the system to manage the options for those criteria that have that.

The result of the search will be viewed either in a map view, depending on what criteria included in the search. The map view will show each garbage location as a pin on the map as well as the user’s own location. Maps can be zoomed in and zoomed out to display accurately.

### 2.3 User characteristics

There are two types of users interacting with the system: the driver and the administrator. Both type of users use the same APP, and the administrator account login can get more information about the user information of all drivers.

### 2.4 Constraints

* Lack of human resources
* Limited resources
* Cost – check with the client
* Time
* Development knowledge

The Internet connection is also a constraint for the application. Since the application fetches data from the database over the Internet, it is crucial that there is an Internet connection for the application to function. Press sensors and capacity sensors we used might add a large deal of money at the project beginning.

### 2.5 Assumptions and dependencies

The assumption about the product is that it will always be used for mobile phones with sufficient performance, with good navigation and Internet access.

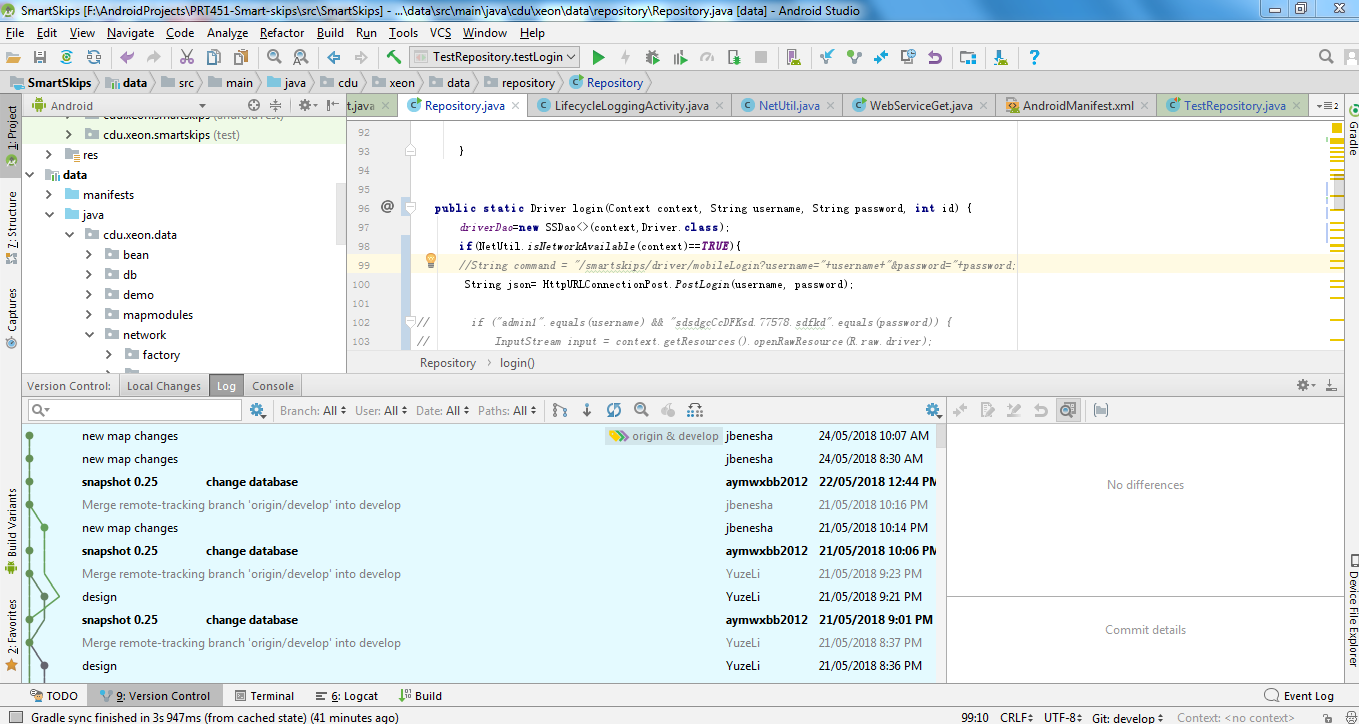
If the phone does not have enough supply application using hardware resources, such as the user may have already assign them to other applications, so there may be applications cannot work as expected or even does not work.

### 2.6 Interface design

This section provides a detailed description of interfaces and provides basic prototypes of the user interface.

### 

### 2.7 version control

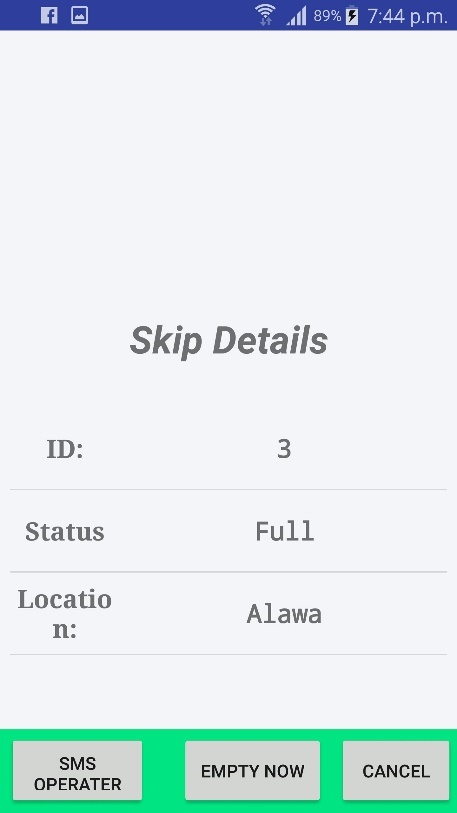


### 2.8 Continuous Integration

gradle: for android studio

maven: for local server

### C:\Users\hyc\AppData\Local\Temp\WeChat Files\745458209267172135.jpgC:\Users\hyc\AppData\Local\Temp\WeChat Files\17017850173091904.jpgC:\Users\hyc\AppData\Local\Temp\WeChat Files\567289891443107760.png2.9 Map functions

The smart skip application will notify the user when the skip is full. Implement route mapping using the shortest path first from source to destination. The application will be integrated with google maps.

Google mapping uses geolocation to be able to map out skips to a different location of the map frame. Geolocation is the identification of the real world geographical location of an object. The geolocation API allows the user to pinpoint a set location.

The Map also incorporates the shortest path first using Dijkstra’s algorithm which finds the shorts path between the nodes in this case identified as the road network. Using the algorithm, the API is also able to calculate the distance and the time it will take from one to skip to landfill.

# Specific requirements

This section contains all of the functional and quality requirements of the system. It gives a detailed description of the system and all its features.

### 3.1 Functional requirements

This section includes the requirements that specify all the fundamental actions of the software system.

ID: FR1

**TITLE**: Administrator and drivers login

**DESC**: Given that the administrator and drivers has registered, then they should be able to login to the App. The login information should delete automatically when user closes App.

**DEP**: NONE

ID: FR2

**TITLE**: User profile page

**DESC**: On the App, a user should have a profile page. On the profile page a user can find whole information about themselves, including user name, user ID, level, count of emptying skips.

**DEP**: FR1

ID: FR3

**TITLE**: User logout

**DESC**: Given that the user has login, when they want logout or switch user, they can use logout function.

**DEP**: FR1 FR2

ID: FR4

**TITLE**: Skips map view

**DESC**: On the App map page, there will should a google map which include all skips location.

**DEP**: FR1

ID: FR5

**TITLE**: Skips detail page

**DESC**: On the App skips details page, user can easy find all information about skips details that user will be choice.

**DEP**: FR1 FR4

ID: FR6

**TITLE**: Download mobile application

**DESC**: A user should be able to download the mobile application through an application store on the mobile phone. The application should be free to download.

**RAT**: In order for a user to download the mobile application.

**DEP**: None

ID: FR7

**TITLE**: Users login in and out smart skip application

**DESC**: A user should be able to login in his or her account and also can quickly log out through a button

**RAT**: In order for a user to have personalized experience

**DEP**: FR6

ID: FR8

**TITLE**: navigation system should be searchable, quickly find location

**DESC**: A user should be able to use global position system to find where he is and the best short path to the destination

**RAT**: In order for a user to use app with accuracy.

**DEP**: FR6

ID: FR9

**TITLE**: the map can be zoomed in and out

**DESC**: A user should be able to freely use the map by zooming in and out so that he can get the best navigation experience.

**RAT**: In order for a user to get the best navigation experience.

**DEP**: FR3

ID: FR10

**TITLE**: store history information

**DESC**: A user should be able to check the place he searched before.

**RAT**: In order for a user to quickly navigate history destination.

**DEP**: FR3

ID: FR11

**TITLE**: Mobile application - Profile page

**DESC**: On the mobile application, a user should have a profile page. A user can check their details and edit their password, phone number and email address.

**RAT**: In order for a user to have a profile page on the mobile application.

**DEP**: FR1

ID: FR12

**TITLE**: Mobile application – find the shortest path to the landfill

**DESC**: When a user clicks empty button, the back-end algorithm will use user and landfill’s location and skips’ states to find shortest path for dumping trash.

**RAT**: In order for a user to get shortest route to the landfill.

**DEP**: FR11

ID: FR13

**TITLE**: Mobile application – View skips states on the map

**DESC**: On the mobile application, a user should have ability to see states of nearby skips.

**RAT**: In order for a user to have ability to see states of skips on the mobile application.

**DEP**: FR11

ID: FR14

**TITLE**: Mobile application – Inform operator

**DESC**: After a user click empty button, the system will send a message to operator’s email.

**RAT**: In order for the operator to know that a full skip will come.

**DEP**: FR11

ID: FR15

**TITLE**: Mobile application – View details of skip

**DESC**: Details of skip should be shown on a new page that pops up on the map. There should be two buttons on the new page. Left button is to trigger empty skip function and another one is to close this page.

**RAT**: The way skip details are displayed.

**DEP**: FR11

ID: FR16

**Title**: Driver Login Page

**DESC**: Drivers can login and enter the home page of This App.

**DEP**: none

ID: FR17

**Title**: User Profile Page

**DESC**: Users can see details of their personal info on this page (such as: name gender age working hours)

**DEP**: FR16

ID: FR18

**Title**: User Logout

**DESC**: User can logout the personal account and exist the app on user profile page

**DEP**: FR16 FR17

ID: FR19

**Title**: Skips map View

**DESC**: There should be a google map on this page; user can see the location, the shortest path on this page.

**DEP**: FR16 FR17

ID: FR20

**Title**: Record skips state

**Desc**: a user should be able to know the state of every skips (red: full, yellow: intermediate state, green: empty state)

**RAT**: In order user get access to the details of every skip

**DEP**: FR16 FR17

ID: FR21

**TITLE:** Locating existing account

**DESC:** User should already have detailed stored in the system from their (Government) account and the application should synch user’s access directly.

**RAT:** In order for users account to synch directly from login.

DEP: FR7

ID: FR22

**TITLE:** Switch result view (switch from different skips)

**DESC:** User should be able to switch to a result view when selecting the skip which as been emptied or needs to be emptied.

**RAT:** User should be able to view skip clearly with more details.

DEP: FR15

ID: FR23

**TITLE:** Skip’s daily record

**DESC:** User should be able to select a skip and see how many times a certain skip has been emptied on that day.

**RAT:** To ensure user keeps track of skips record and not overuse a certain skip.

**DEP:** FR20

ID: FR24

**TITLE:** Automatic refreshing of the google maps

**DESC:** Application map needs to constantly refresh to remain updated which skip is full or empty.

**RAT:** Increase efficiency and usability

DEP: FR13

ID: FR25

**TITLE:** Time taken to destination

**DESC:** User should be able to view the location and time it will take for skip to be emptied.

**RAT:** Increase efficiency and usability by providing user feedback

**DEP:** FR4

### 3.2 Non-functional requirements

ID: NFR1

**TITLE**: User login response time

**DESC**: Given users a good login experience, accelerating response time.

ID: NFR2

**TITLE**: Logout/switch user response time

**DESC**: Given users a good logout/switch user account experience, accelerating response time.

ID: NFR3

**TITLE**: Failure in login

**DESC**: When user failure in login, they cannot do anything. If this problem occurs, you need to find the cause of the problem and fix it

ID: NFR4

**TITLE**: Missing interface functions

**DESC**: Missing inter face functions will lead to a lot of problems, such as no login button, no logout button, no user profile button, no skips details button.

ID: NFR5

**TITLE**: compatibility

**DESC**: The app can be used in different type of mobile phones and system version.

**RAT**: In order for all users can use the application.

ID: NFR6

**TITLE**: Backup

**DESC**: copying data in a file system or database system. In case of disaster or error operation, it is convenient and timely to restore the effective data and normal operation of the system

**RAT**: In order for all users can get back data in case of disaster.

ID: NFR7

**TITLE**: Performance

**DESC**: The reaction time of the application shall be more and quicker.

**RAT**: In order to make application have excellent performance.

ID: NFR8

**TITLE**: Availability

**DESC**: Users can get access to the mobile application anytime and anywhere. Denial of service should not happen.

ID: NFR9

**TITLE**: Database response time

**DESC**: Adding, updating, querying and deleting data should take less than 3 seconds and take less than 5 seconds on peak time.

ID: NFR10

**Title**: Scalability

**DESC:** Database should have enough room for saving data and avoid overloading and data overflow. Data volume will grow, as the number of registered users increase.

ID: NFR11

**TITLE**: Compatibility

**DESC**: Operation system of this mobile application is Android. The mobile application should be compatible for most types of mobile phone.

ID: NFR12

**TITLE**: User login response time

**DESC**: Users can receive the response in a short time.

ID: NFR13

**TITLE**: Logout/switch user response time

**DESC**: users should have good logout/switch user account experience, short response time.

ID: NFR14

**TITLE**: Failure in login

**DESC**: When user failure in login, they cannot do anything. If this problem occurs several times, the app should be locked

ID: NFR15

**TITLE**: Missing interface functions

**DESC**: Missing interface functions will lead to a lot of problems, such as no login button, no logout button, no user profile button, no skips details button.

ID: NFR16

**TITLE**: The software requiring a lot of memory to run

**DESC**: The application requires a large amount of memory to be able to run smoothly due to feature such as google map.

ID: NFR17

**TITLE**: Failure in detecting the shortest path

**DESC**: The application failing to detect the shortest path could lead to several levels of inefficiency and affecting the user’s time. The algorithm needs to be integrated correctly.

**RAT**: In order to increase user travel efficiency

ID: NFR18

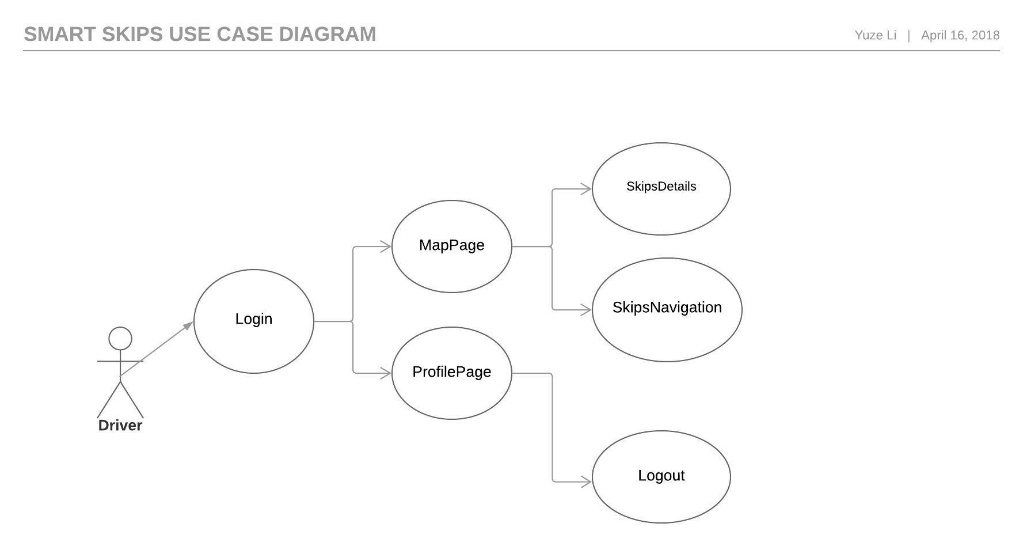
**TITLE**: Failing to locate full and empty skips

**DESC**: The system should be able to constantly scan the map and identify which skips are full or empty.

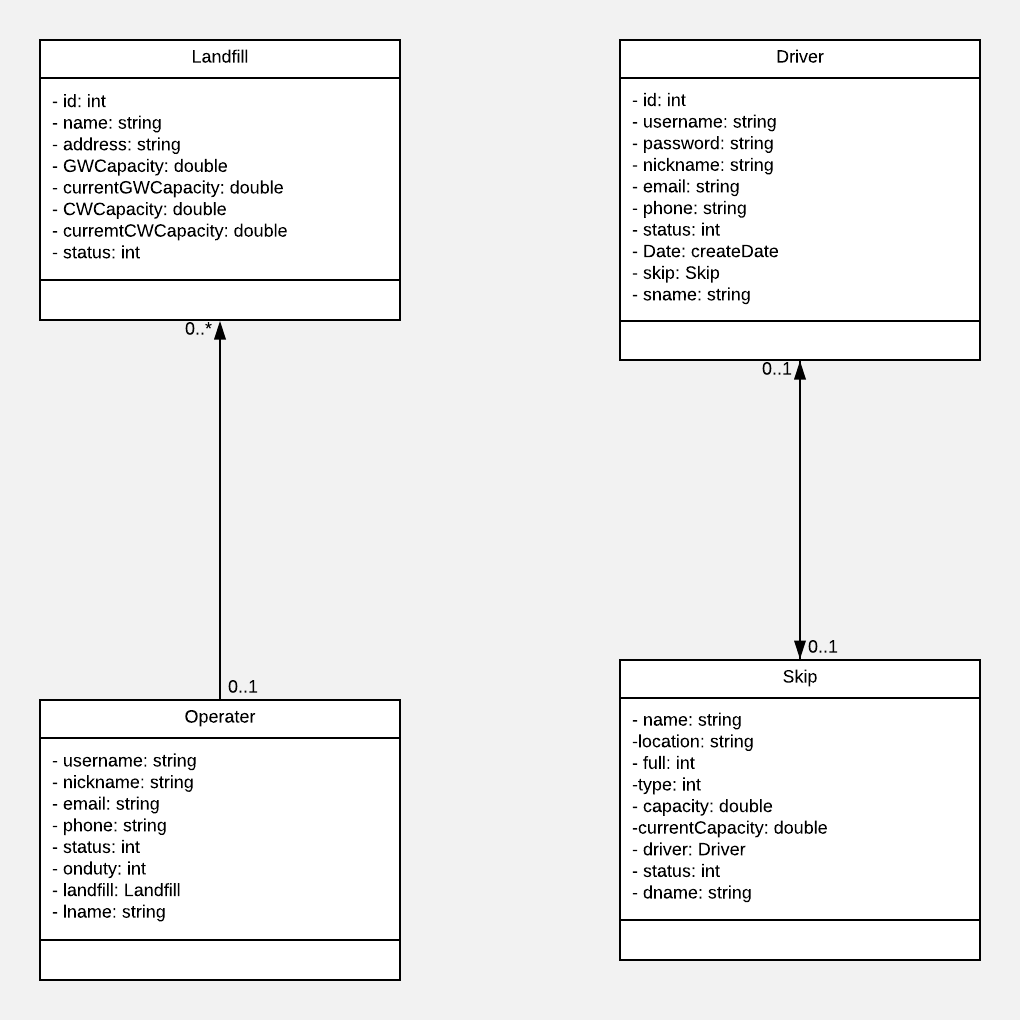
**RAT**: The effectiveness of the application depends on the ability to scan for skips.

# C:\Users\kangyinrui\Desktop\skips navigation sequence diagram.jpegUML patterns

**skips navigation sequence diagram**

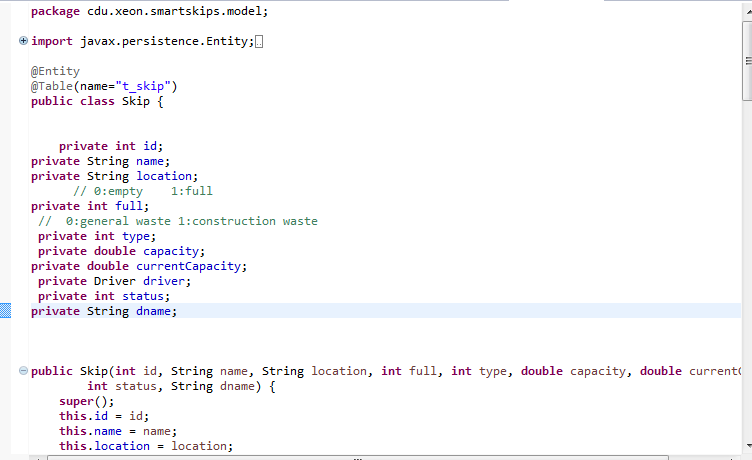


**Smart Skips Use Case Diagram**



**Smart Skips Use Case Diagram**

# MVC patterns

Model:

View:

Controller:



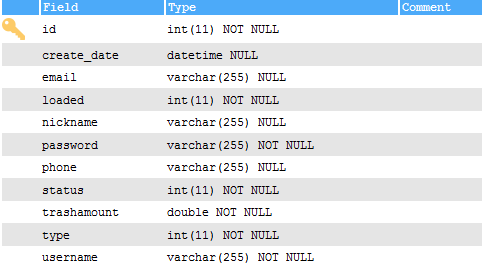
Verify the output:

# Database

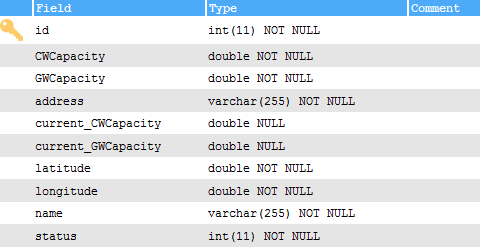
### 6.1 SQLite database

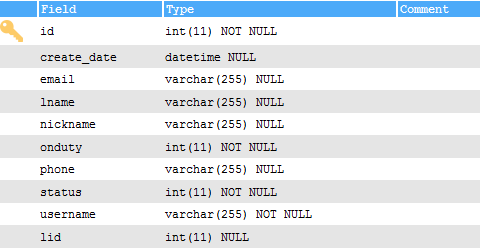
This is mobile application internal SQLite database, after the first load data from the cloud server, part of the data will be stored in the database in the mobile phone, when the user lose the network connection, but has been login, can call these local data offline, continue to use the software.

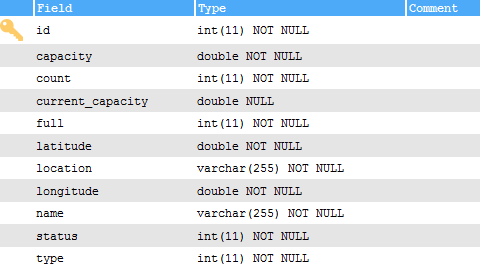
### 6.2 database design

Driver data table

Landfill

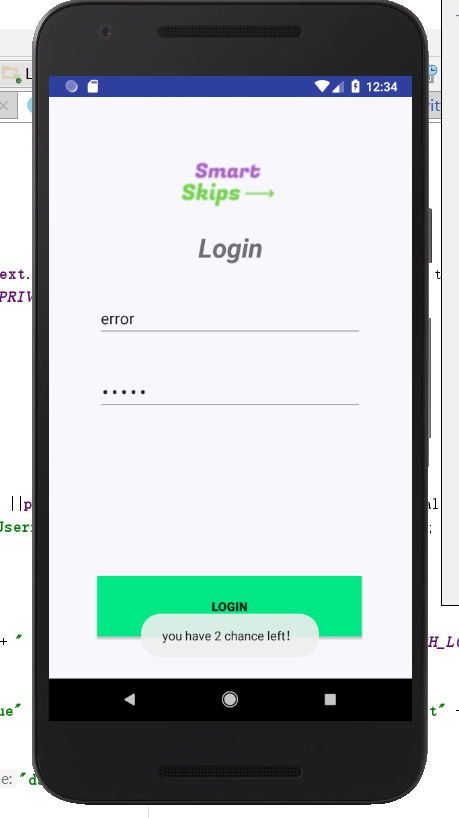
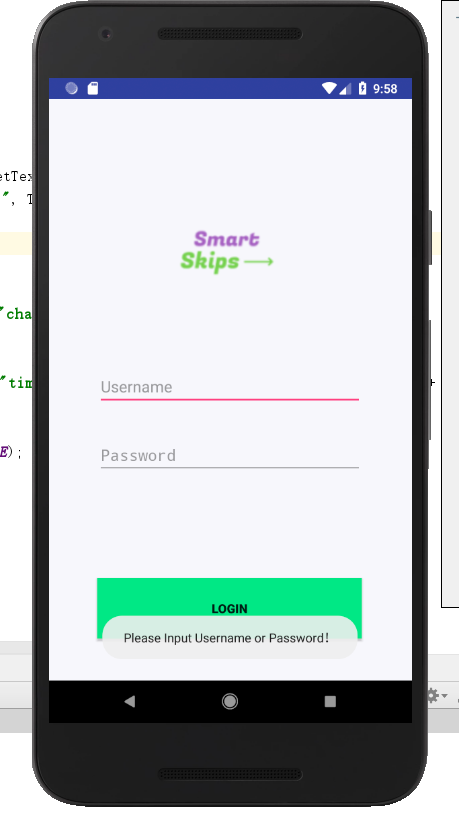


Operator

Skips

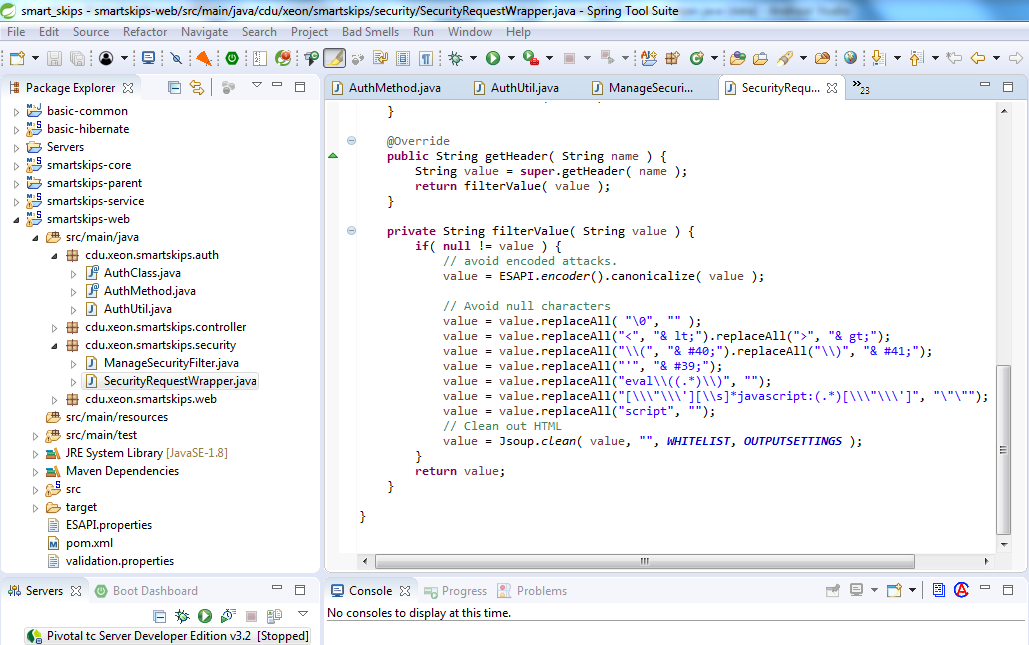
# Security measures

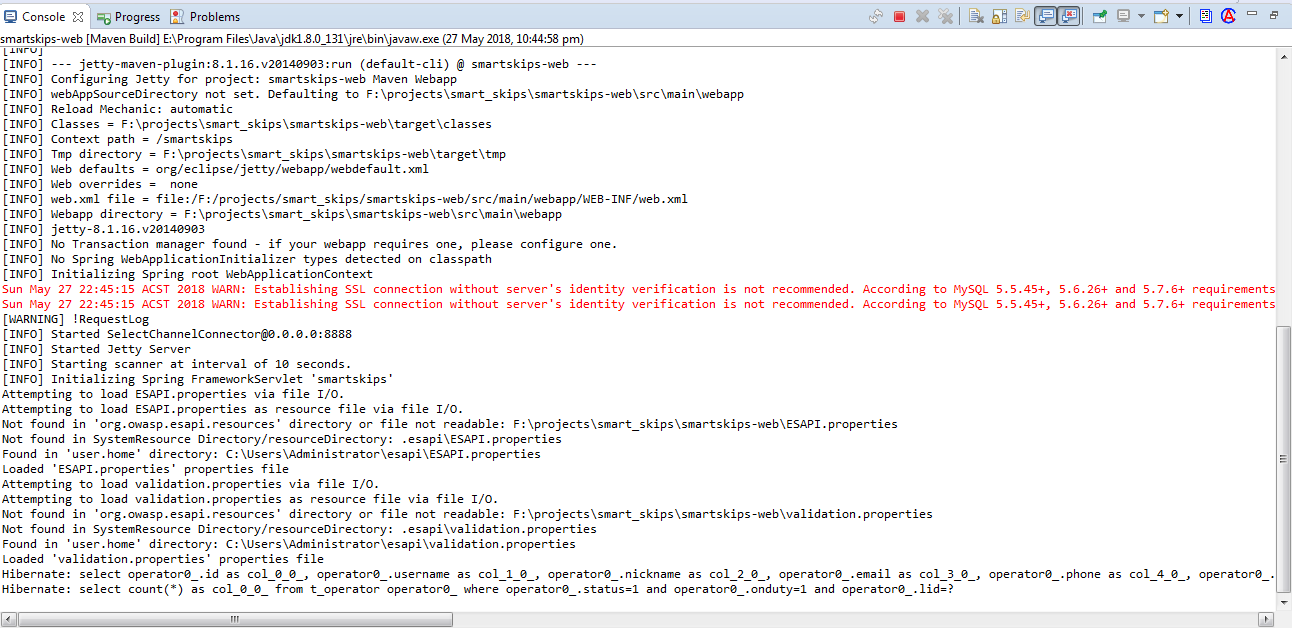
### 7.1 security in client sever

The user has three chances to enter the account password, and the account will be temporarily frozen for 30 seconds if the input is incorrect. 

### C:\Users\hyc\AppData\Local\Temp\WeChat Files\339968727503417303.png7.2 Authorization certification

### 7.3 White list

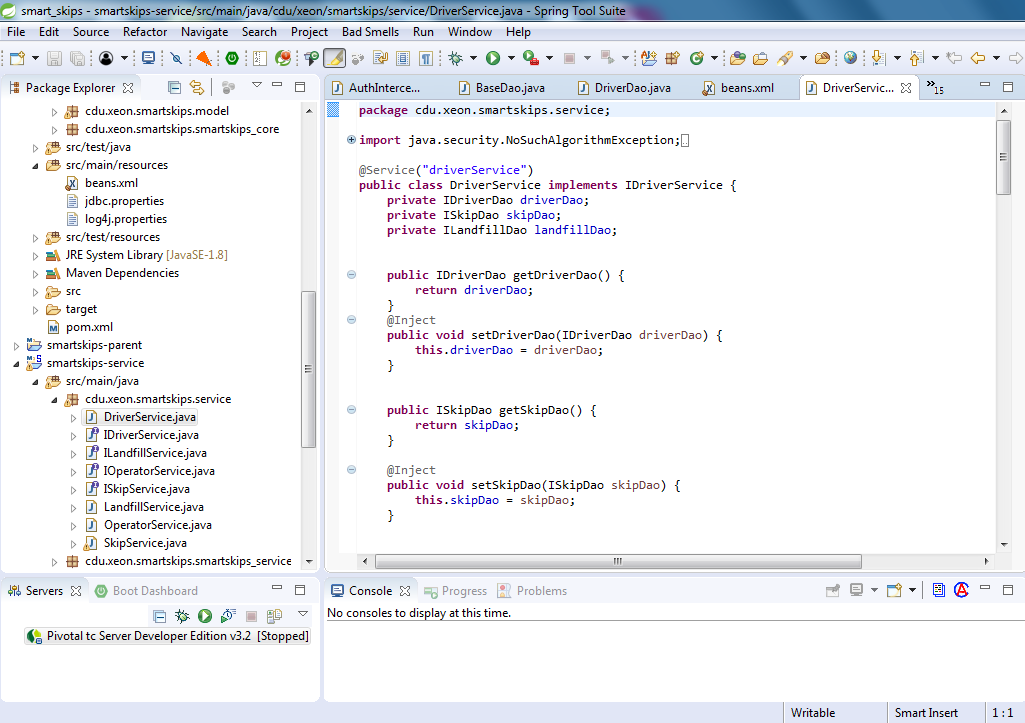
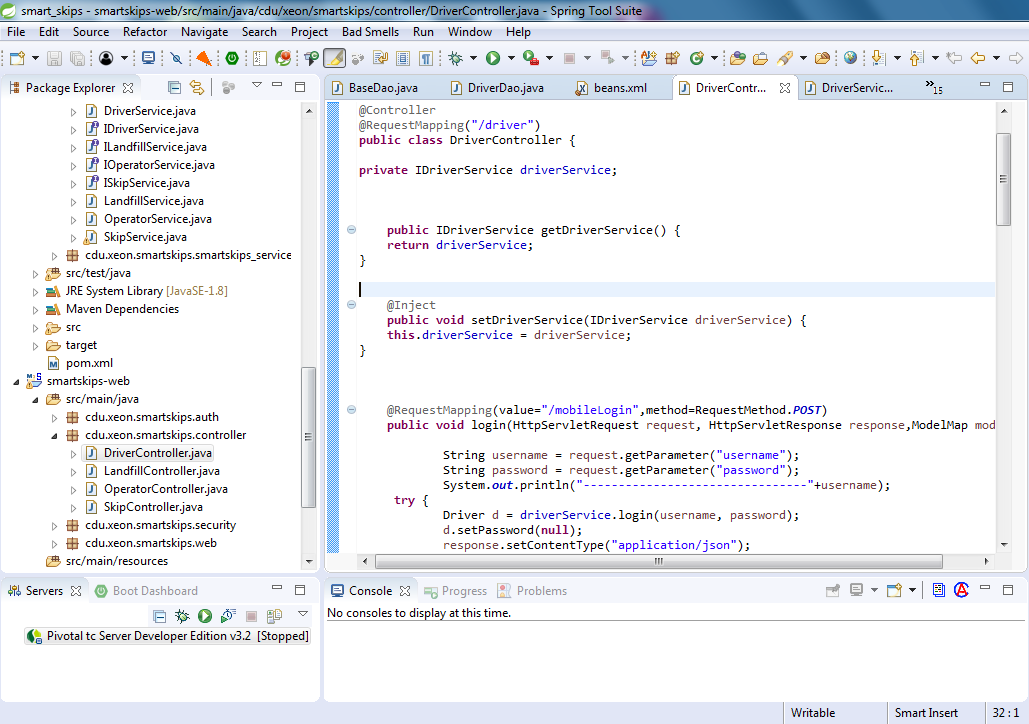
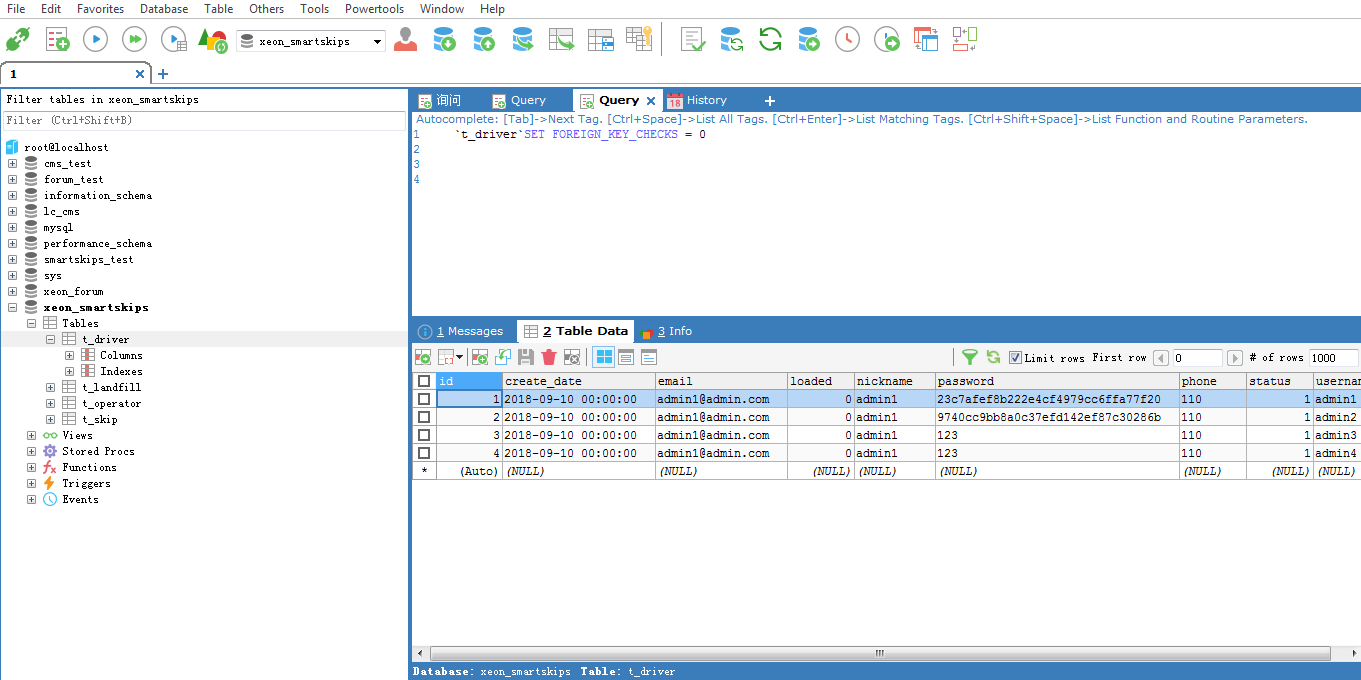
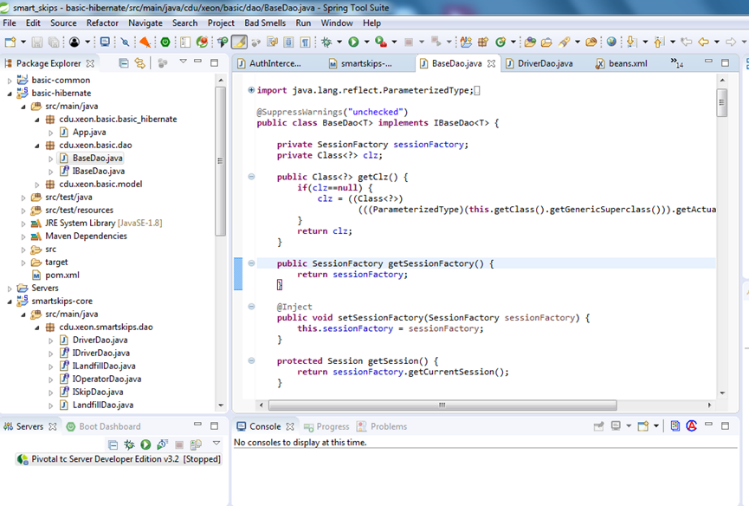


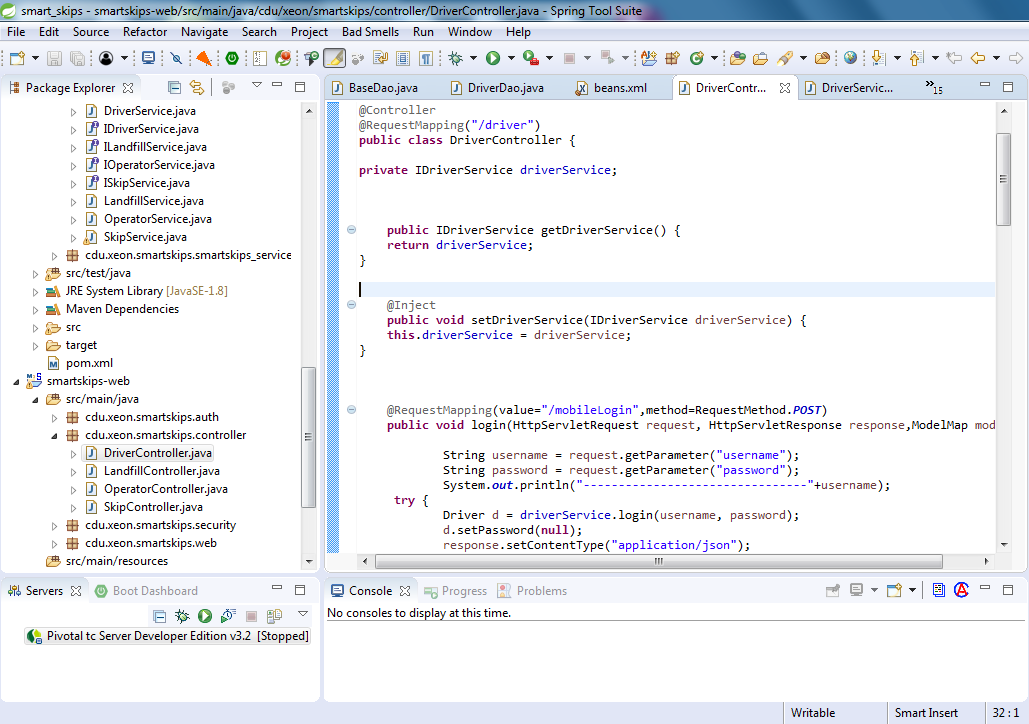


### C:\Users\hyc\AppData\Local\Temp\WeChat Files\327841657463498547.png7.4 post message

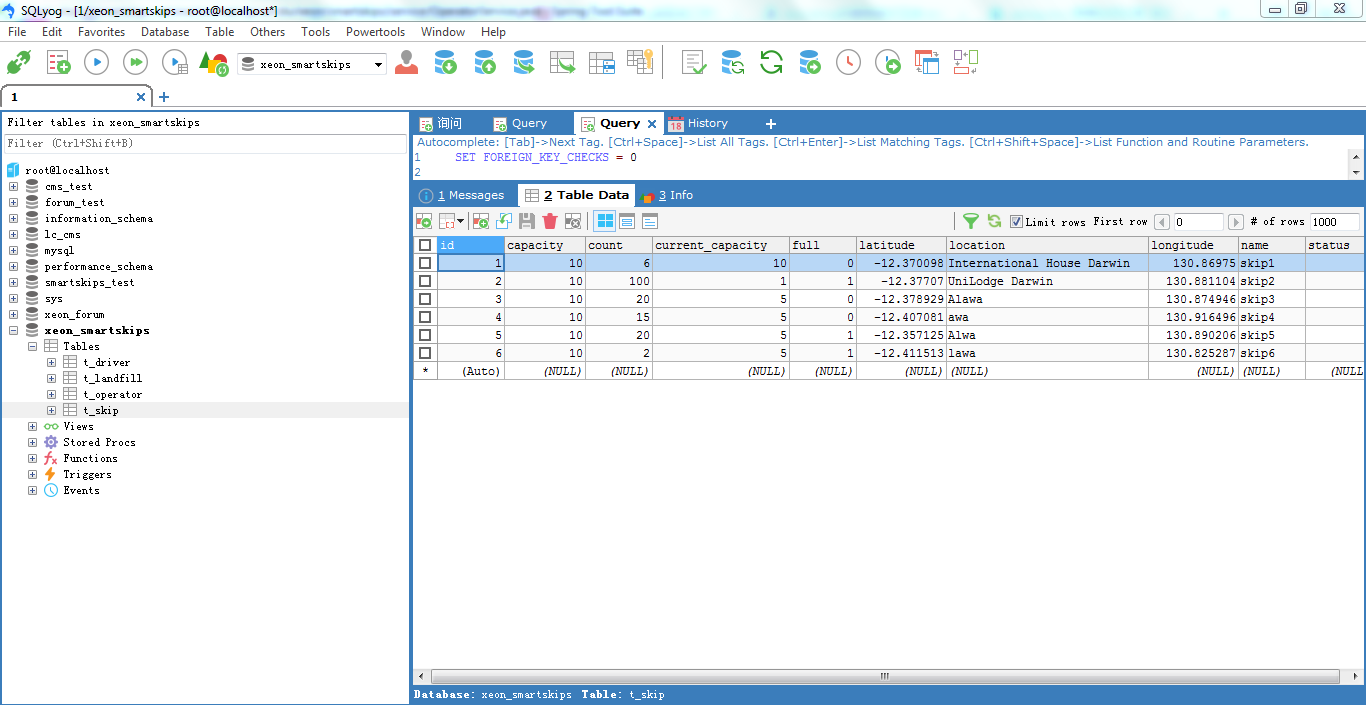
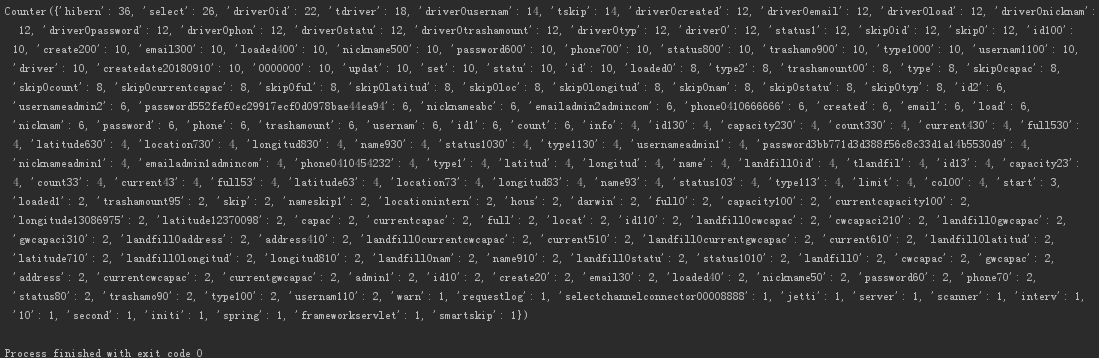
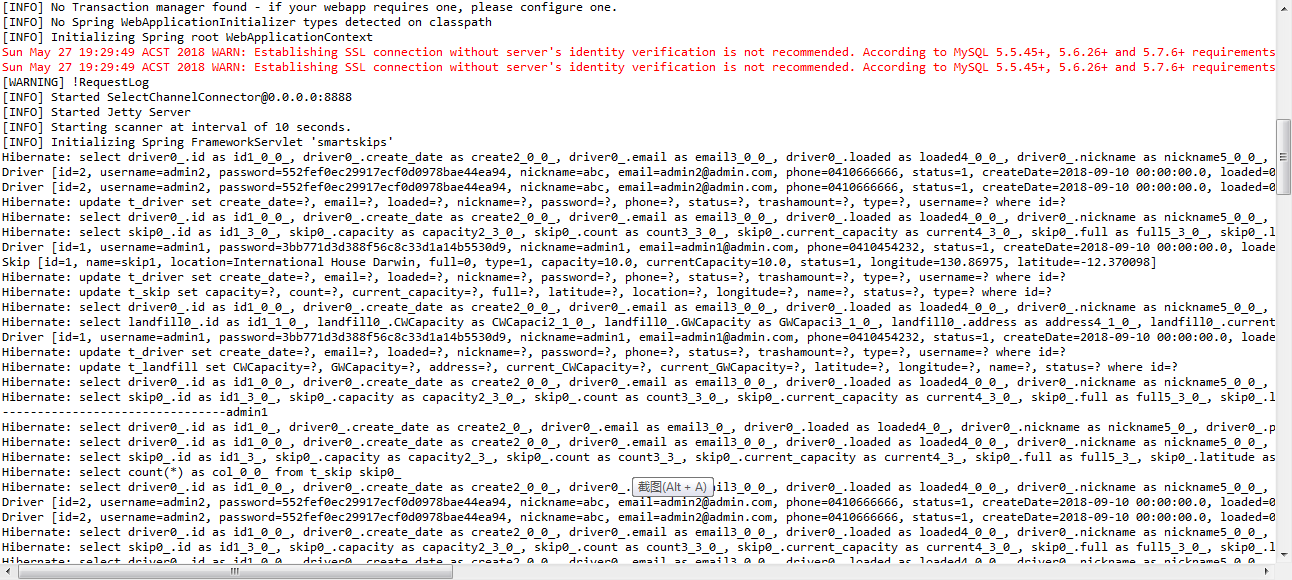
### C:\Users\hyc\AppData\Local\Temp\WeChat Files\893564654216530955.pngC:\Users\hyc\AppData\Local\Temp\WeChat Files\19065595733444670.png7.5 Error page

### 7.6 MD5

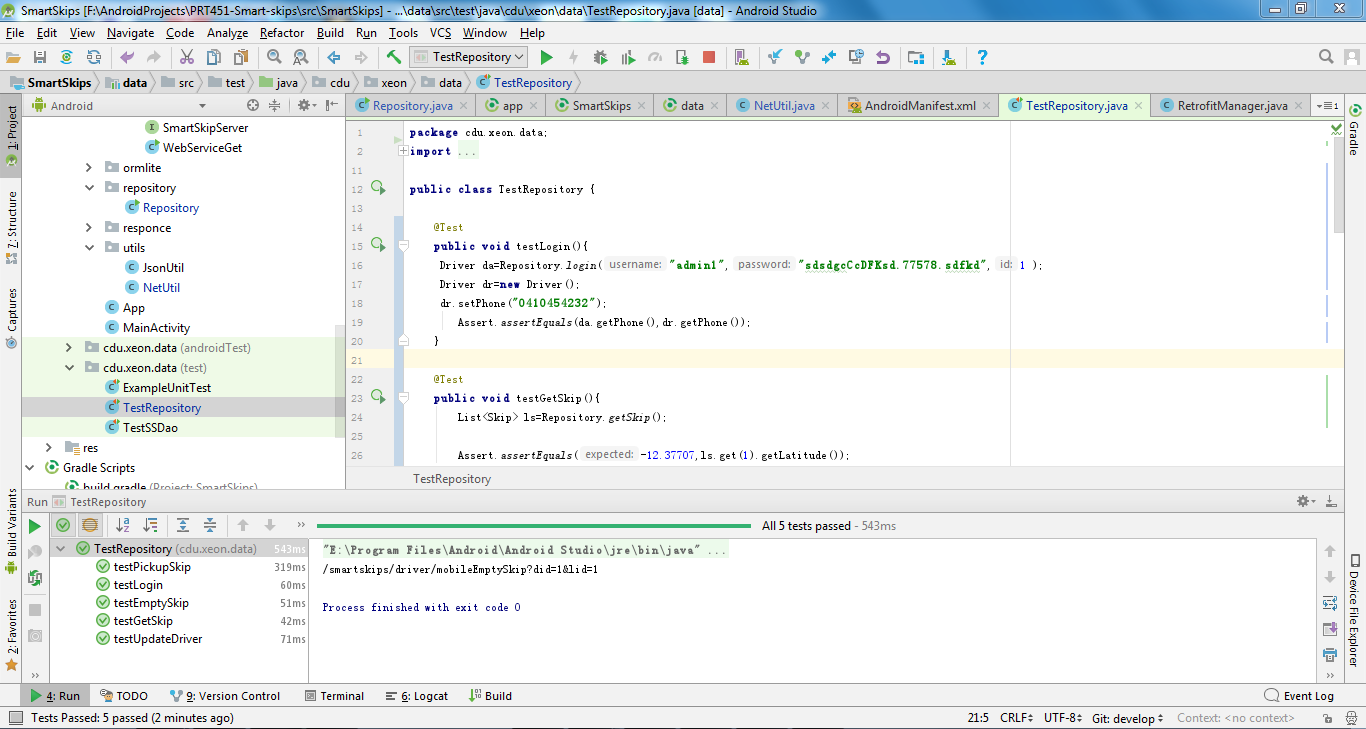
Rivest developed a more technically mature md5 algorithm. It adds the concept of less danger on top of MD4. While MD5 is more complex than MD4,and more secure.



### Login analysis



# Unit test

Unit Testing, also known as module Testing, is the Testing of program modules, the smallest Unit of software design, to test correctness. A program unit is the smallest testable part of an application. In procedural programming, a unit is a single program, function, procedure, etc.

**For local server**

