

Package Delivery Application System

Zeng Jiaxing
1452680

Li Yichao
1452666

Liu Lidong
1354362

Yi Siqu
1452662

May 12, 2016



Contents

1	Introduction	3
1.1	Purpose Restatement	3
1.2	Progress	3
1.3	Overview	4
2	Analysis Mechanism	4
2.1	Persistence Mechanism	4
2.2	Inter-process Communication Mechanism	6
2.3	Task Distribution Mechanism	7
2.4	Security Mechanism	7
3	High Level Architecture	8
3.1	Presentation Layer	9
3.2	Application Layer	9
3.3	Service Layer	9
3.4	Domain Layer	9
3.5	Persistence Layer	10
4	Deployment Diagram	10
5	Analysis Model	11
5.1	Sign In	11
5.2	Set Address	13
5.3	Transfer Confirmation	14
6	Initial Snapshots of the System's User Interface	16
6.1	Customer	16
6.2	Postman	19
6.3	Management Staff	21
6.4	Added User Interface	22
7	List of References	24
7.1	System design and implementation of third party logistics management based on electronic commerce	24
7.2	Design and realization of logistics management system based on uml modeling technology	24
8	Contributions of team members	24

1 Introduction

This section gives a scope description and an overview of everything included in the system analysis document. Also, the purpose for the course project is re-stated and the progress made since the previous report is shown.

1.1 Purpose Restatement

The goals of our course project is to develop a package delivery system which would increase the efficiency of delivery and reduce the errors made by human inputs, etc. This document is the second version of the system for the development team. We intend to be more realistic in this version with particular implements or workflows in diagrams.

1.2 Progress

From the design of first version, we have made several progress as follow:

We firstly consider the analysis mechanism, as the foundation and boundary conditions for further analysis. In this part, we not only pay attention to the progress of realizing the system, but take the security, persistence, data storage and so on into account.

We analysis the architecture from an abstract level, ranging from the presentation layer to the persistence level, also connected to several third-party systems through provided APIs.

To take the system into real use, we need to design deployment diagram. From this, we get to know the physical linking methods and net protocols between devises. From this, we have an initial impression on how the system is distributed and worked as a whole.

In analysis model, we develop from the use case modeling designed in the first version, focus on the realization of the functions mentioned in use case diagram and consider the order and communication among particular parts. In most cases, the customer, who implements an action, would send his/her request directly through the user interface. For instance, when the customer intends to check the package or add an address, he/she will use the mobile app or web pages to interact with background system. The user interface is only able to communicate with specific controller, and to display information sent from the controller. So the controller means an important role in the bottom level of the system, which

can directly attach to the database entities or some third-party systems. With this structure, the security of our data can be well guaranteed.

As for the user interface, we adjust a few details and also design some other snapshots for the analysis model above.

1.3 Overview

The following contents mainly include analysis mechanism, architectural analysis, deployment schedule, analysis model and several updated snapshots of the user interface.

First part is analysis mechanism, consisting of different parts of mechanisms. Each mechanism has several issues to consider.

As for the architecture analysis, we design a high-level architecture with five layers and three related outside systems. To make the model more concrete, we also specifically consider one service as an example.

Then is the deployment schedule. It is realized in the form of a deployment diagram which shows how our system is dispatched into different kinds of hardware and the way the fragments communicate with others to work as a whole.

For the analysis model, we select three important use cases to realize. Each case has a realization of a class diagram, a sequence diagram and a communication to represent it in several points of view.

After is the updated user interface, we consider a particular progress that the user selects and modifies the address information from the mobile client.

At the end of the document, there are several references we study and apply in the design progress and a list of contributions made by each team member.

2 Analysis Mechanism

2.1 Persistence Mechanism

2.1.1 Granularity

User Information (ID, name, sex, phone number, fingerprint, address, designated person)

Postman Information (ID, name, sex, phone number, region, salary)

Order Information (order number, order status, create time, finish time)

Package Information (package type, package status, package position)

Signature Authorization (order number, designated person information)

2.1.2 Volume

User Information:	All
Postman Information:	All
Order Information:	All executing orders and orders that completed but dont exceed in a year
Package Information:	Package Information: packages that belong to orders that still exist in the database
Signature Authorization:	All executing orders that will be signed up by a designated person

2.1.3 Duration

User Information:	Forever
Postman Information:	Forever
Order Information:	One year after finished
Package Information:	Delete when the related order is deleted
Signature Authorization:	Delete when the order is completed

2.1.4 Retrival Mechanism

User Information:	ID is unique among all the user
Postman Information:	ID is unique among all the postman
Order Information:	Each order has a unique order number
Package Information:	Each package belongs to a unique order while the order is unique
Signature Authorization:	Each signature authorization belongs to a unique order

2.1.5 Update Frequency

User Information:	Update when users modify their information
Postman Information:	Update when postmen or management staff modify postmens information
Order Information:	Update all the time until the order is completed
Package Information:	Update following the change of the order
Signature Authorization:	When the authorization is created or the related order is completed

2.1.6 Reliability

The objects need to survive a crash of the process. When a process crash, all data that were related to this process would have to roll-back.

2.2 Inter-process Communication Mechanism

2.2.1 Latency

Normally within 100ms

If it is running under High load, the latency can be within 1.5s.

2.2.2 Synchronicity

Different service will use different synchronicity.

If some services that need to be done before another service, they will use synchronous communication while others use asynchronous communication.

2.2.3 Size of the Message

1KB ~ 1MB

Some tiny message like Boolean value with process information might be small so 1KB is enough while some message include a long string which need large space. If the size is larger than 1MB the transmission will be divided into several parts.

2.2.4 Protocol

Depends on which kind of message is transferring.

Such as login service, the message will be sent with a format of Account and password and user type in binary. There will be an interval between the information.

2.3 Task Distribution Mechanism

2.3.1 Postman Selection

A delivery will execute by a postman.

2.3.2 Maximum Package

Each postman only can deliver a certain amount of package.

2.3.3 Time Limit

Each order should be delivered before a certain time.

2.3.4 Minimum Distance

The system will arrange a route with a shortest distance for postman to deliver package.

2.4 Security Mechanism

2.4.1 Data Granularity

50 TB

2.4.2 User Granularity

100,000

2.4.3 Security Rules

High-level service is controlled by management staff. Low-level service is controlled by Super administrator.

2.4.4 Privilege Types

System > Super administrator > management staff > postman = customer

3 High Level Architecture

Our layered architecture is based on five layers: presentation, application, service, domain and persistence. This pattern presents an overview of the responsibilities of each layer and the components that compose each layer.

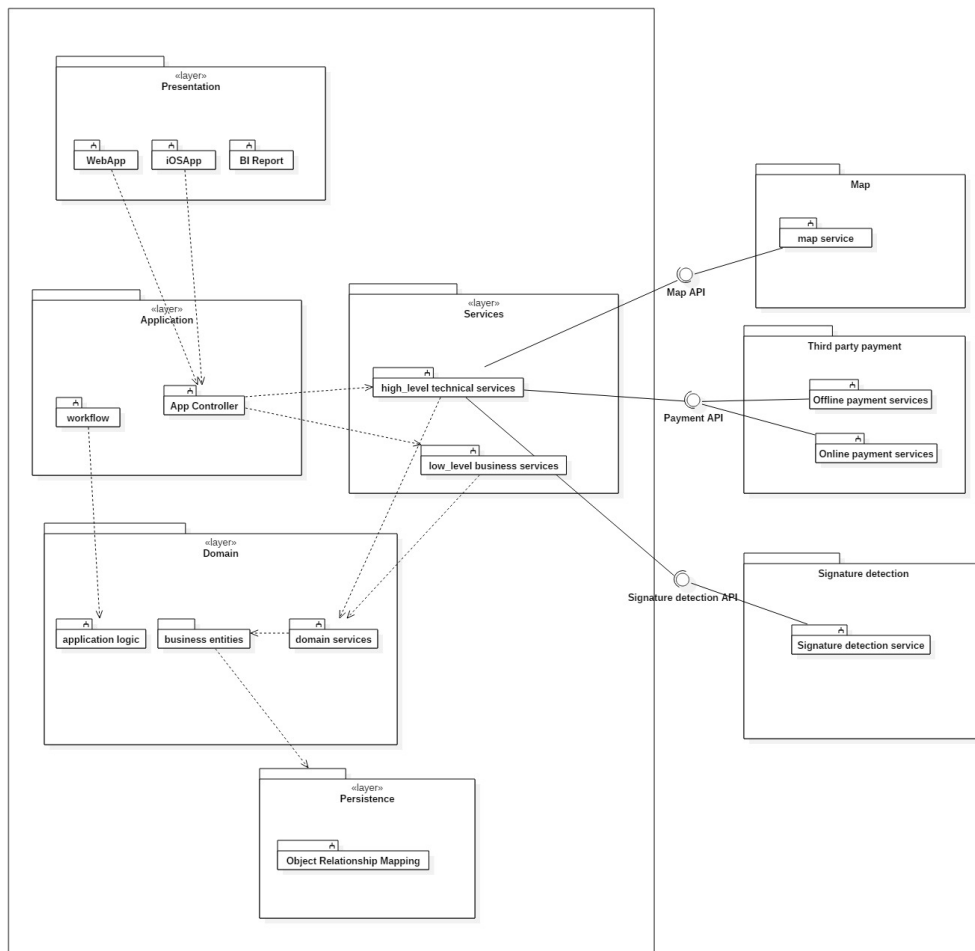


Figure 1 Five-Layered High Level Architecture

3.1 Presentation Layer

As the highest layer of the structure, the presentation layer consists of a series of forms (pages) with which the user interacts. Each form contains a number of fields that display output from lower layers and collect user input. We have three subsystems in this layer: WebApp, iOSApp and BI report. WebApp and iOSApp are mainly for app users, including customers and postmen, while BI report is designed for our management staff.

3.2 Application Layer

The application layer would handle requests from the presentation layer and encapsulate dependencies between forms and the logic associated with navigating between them. It contains two subsystems: App Controller and workflow. App controller would control the containments shown on the App forms and call relating services according to the users' requests. Business processes which reflect the macro-level activities the business performs are encapsulated by workflow subsystem that orchestrate one or more business components to implement a business process.

3.3 Service Layer

The service layer contains business components that provide high-level technical services such as delivery scheduling, package tracking and exceptions handling. Meanwhile, as our system is for package delivery, some business functionalities like GPS navigation, payment services and signature detection are achieved by API provided by outer systems. We also have a low-level business subsystem, offering services which are commonly used by different domains.

3.4 Domain Layer

This is where our business rules and logic resides, as well as business entities that are defined. Business entities are data containers. They encapsulate and hide the details of specific data representation formats. Business logic and business services components can interact with independent business entity via business service subsystem, or they can use a business entity to set their own state and then discard the business entity.

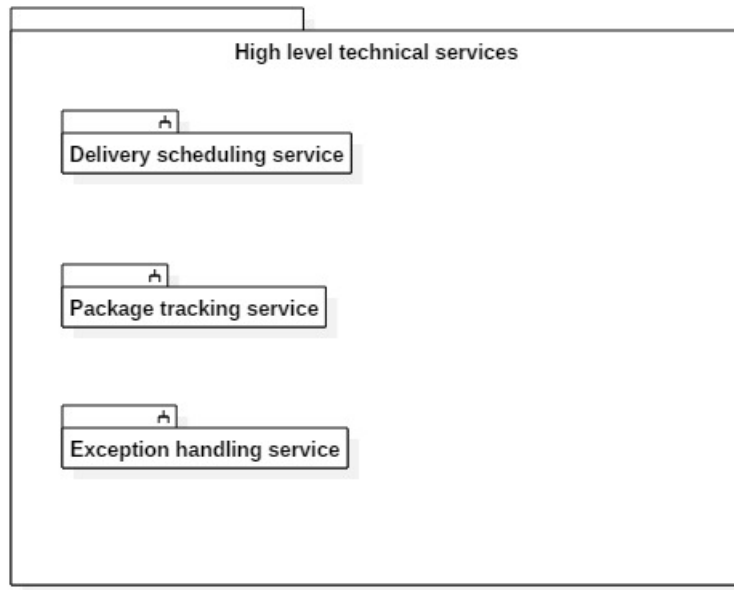


Figure 2 High Level Technical Services

3.5 Persistence Layer

As the bottom and the foundation of our architecture, persistence layer would enhance the data security of our system. We use OR(object relationship) Mapping technique to convert business entities into simpler data and then store them in our database, which isolates the business layer from the details of the specific data storage solution. As the domain layer does not interact with the database directly, we avoid most errors caused by illegal operations.

4 Deployment Diagram

The diagram above is our initial design for deployment diagram. It includes three servers: web server, application server and database server, which are connected by LAN network. The execution environment of the database server is Oracle 12C. Meanwhile, we have three clients, web client, customer client and postman client. They are all connected with the servers by HTTPS network. Postman client is a little bit different, it has two outer devices: fingerprint detection devices and GPS navigation devices. They are integrated together to help postman business. More details

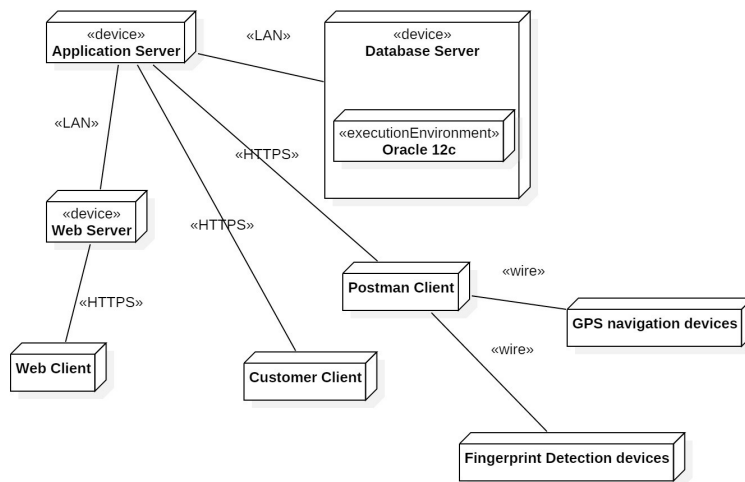


Figure 3 Deployment Diagram

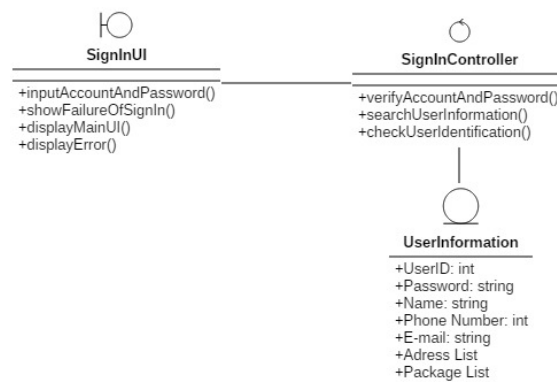
about this diagram would be added in the next assignment.

5 Analysis Model

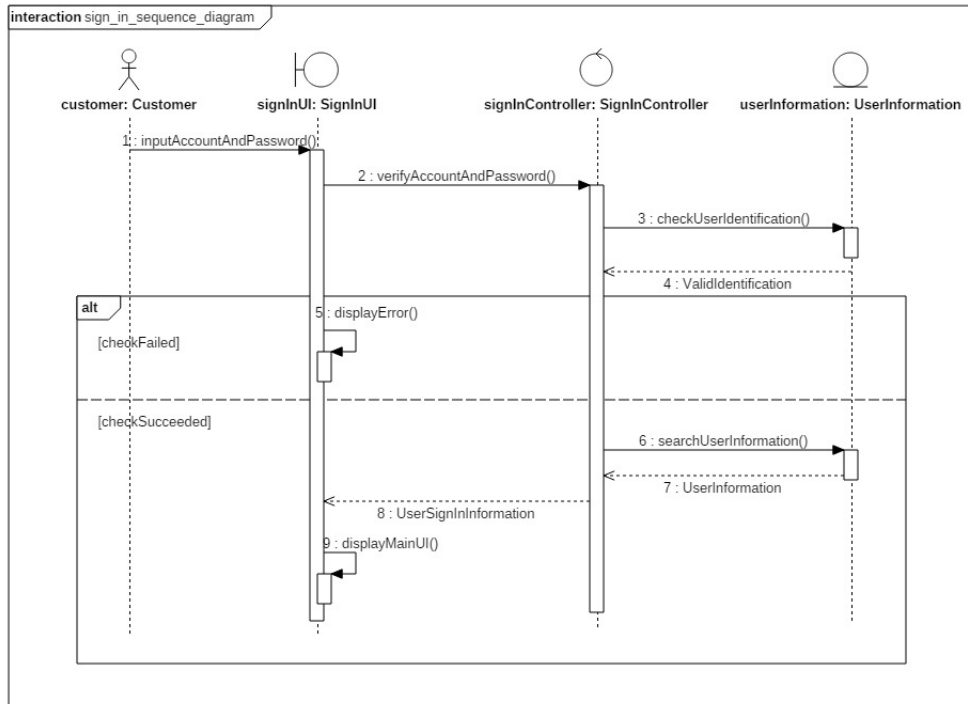
This section includes realizations of three use cases, with three diagrams for each case.

5.1 Sign In

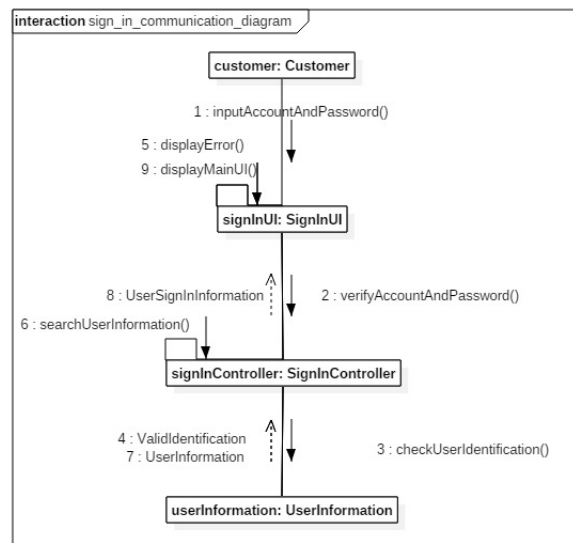
5.1.1 Class Diagram



5.1.2 Sequence Diagram

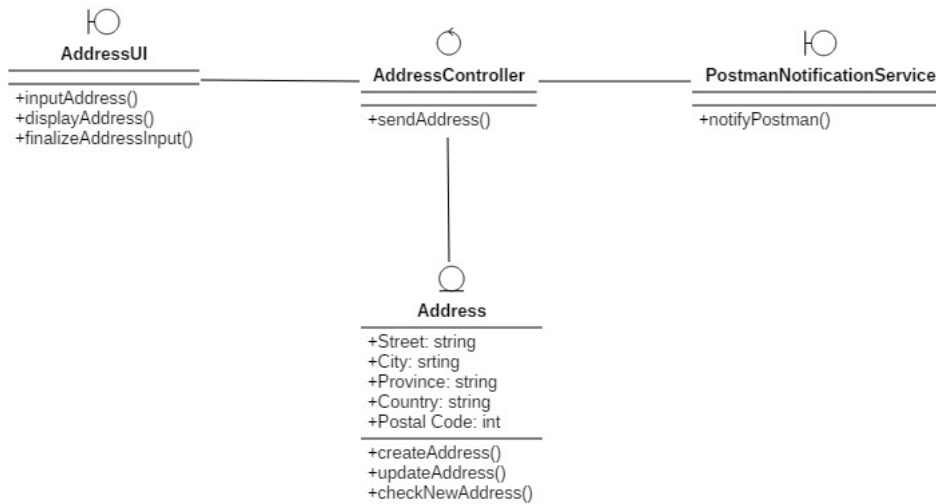


5.1.3 Communication Diagram

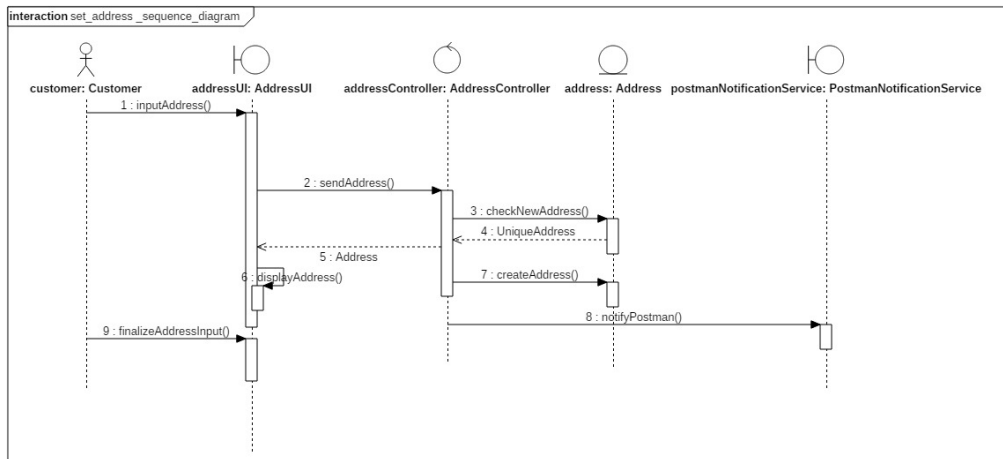


5.2 Set Address

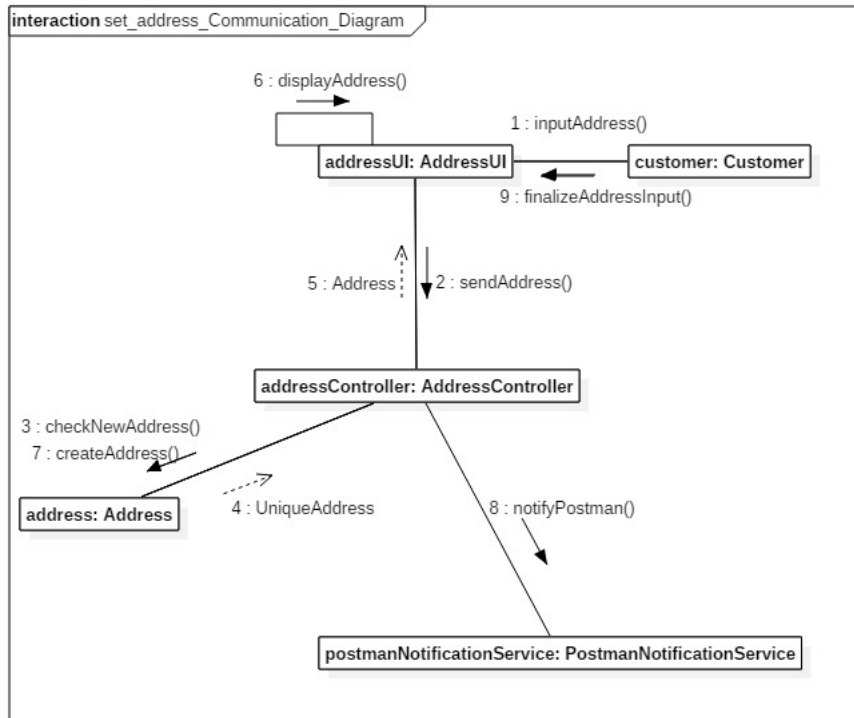
5.2.1 Class Diagram



5.2.2 Sequence Diagram

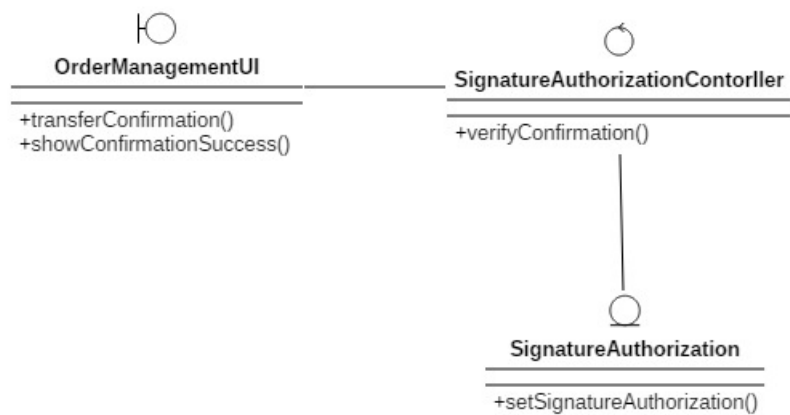


5.2.3 Communication Diagram

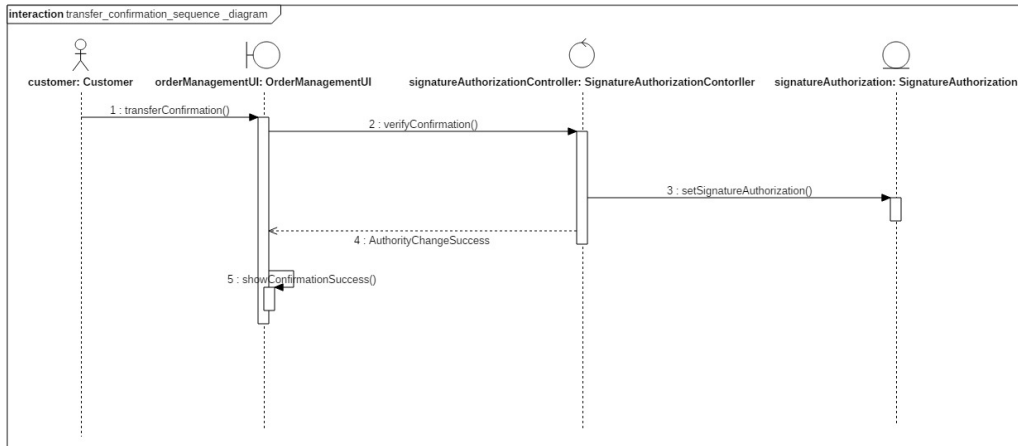


5.3 Transfer Confirmation

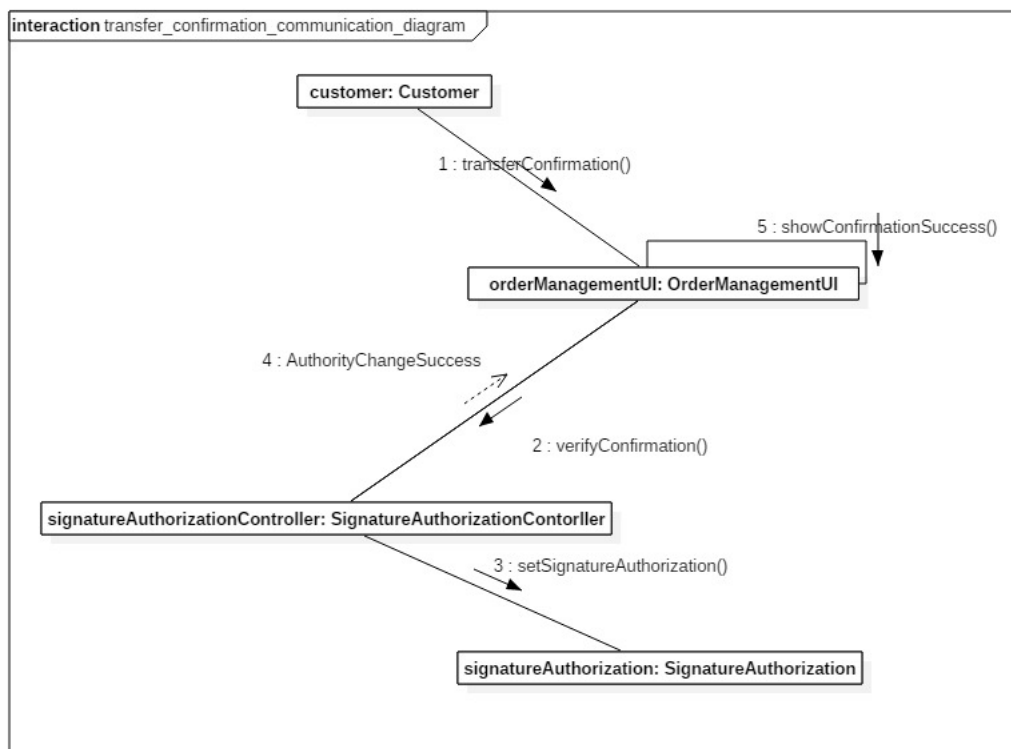
5.3.1 Class Diagram



5.3.2 Sequence Diagram



5.3.3 Communication Diagram



6 Initial Snapshots of the System's User Interface

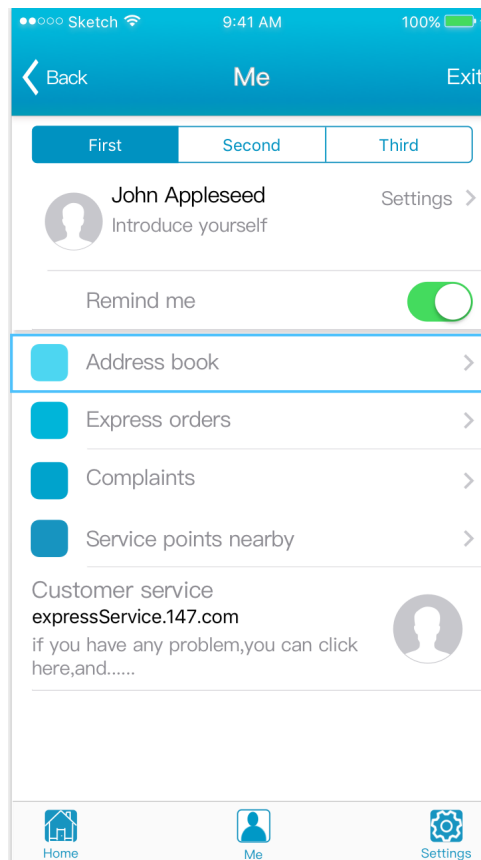
Our group have made eight user interface, altogether four pictures belong to the customer, two belong to the Courier, and others are about background data management.

And this time, we update our user interface by adding two picture in the end of this section.

6.1 Customer

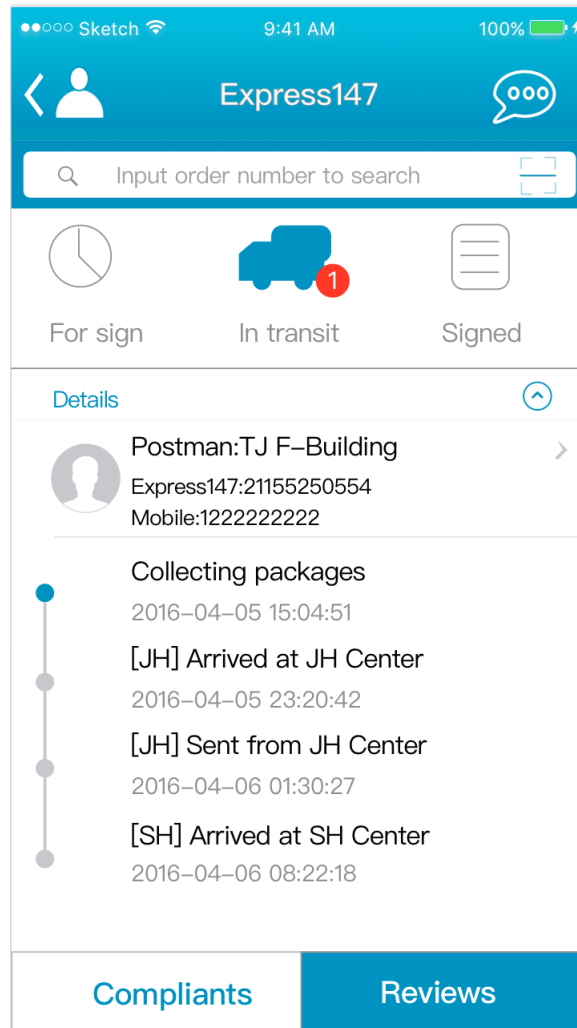
6.1.1 Main

The first figure is Lord of the user interface, the customer can implement some functions via these buttons, such as: to remind the receipt, address management, view the express orders, complaints, see nearby service points, and online customer service.



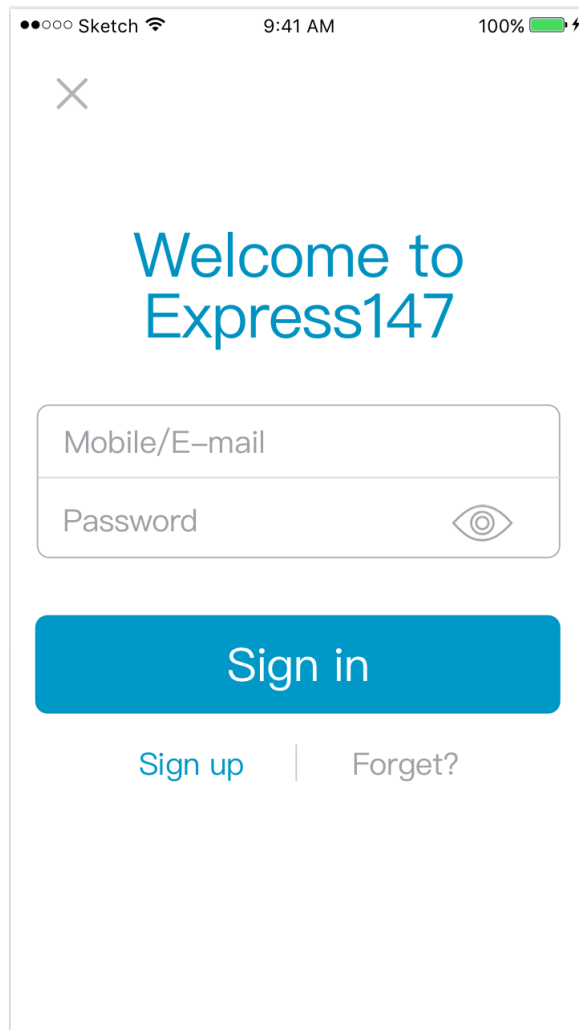
6.1.2 TransitInformation

The second picture shows that: customers can use the express waybill number that logistics companies provide to track the package, and then make complaints or reviews.



6.1.3 Sign In

The third picture is a sign in interface, and the user can log in or sign up for an account.



6.1.4 Add Address

Customer's last picture: users edit shipping address, and you can set the default shipping address.

•••• Sketch 9:41 AM 100%

< Add adress Save

Receiver

Tel

Region Select >

Street Select >

Details

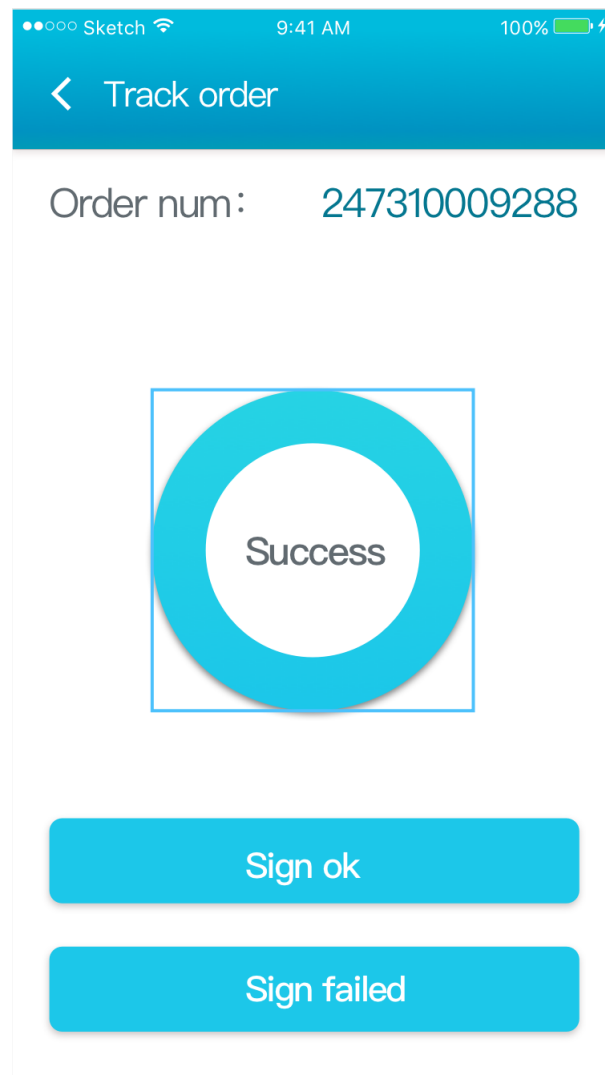
Set as default adress

6.2 Postman

These two user interface belongs to postman. These two user interface belongs to postman.

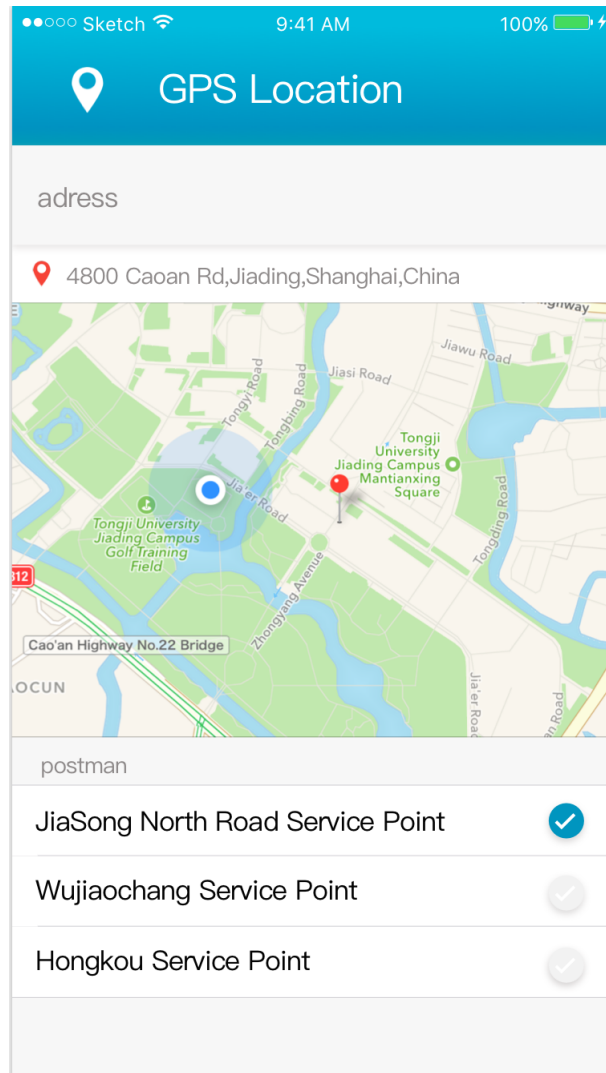
6.2.1 Track Order

First Photo: courier can track packages information, whenever he succeeded in delivering a package, the information will be inserted into the system, and after the customer received the package, he can update the package status with sign success or failure. If you fail to sign, the staff will re-enact a sign program.



6.2.2 GPS Location

The second one: When a customer orders a product and the address of the customer has been not stored in the database, the delivery person will go to the address, and collect the GPS location for the customers address. And the system will choose a near service point for customer.

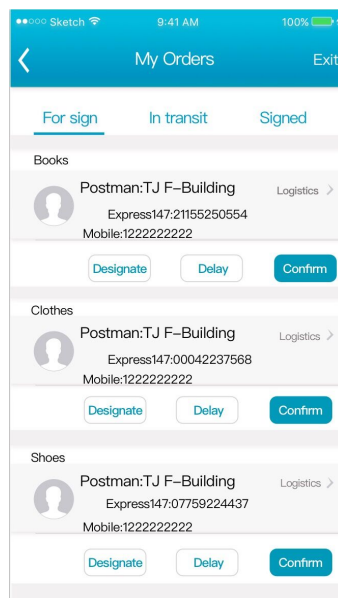


6.3 Management Staff

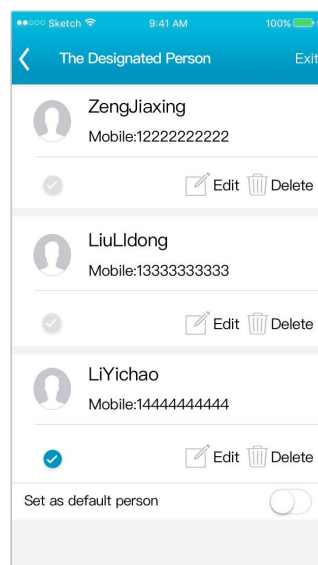
These two picture are about database management. The staff can ensure and manage orders and evaluate recent transactions. And also they can communicate with customers and deal with their complaints.

as the designated person when they are not at home or busy. If the customer chooses one person, his or her information will be updated in the system, and then the postman can identify the designated person.

6.4.1 Orders For Sign



6.4.2 Designated Person



7 List of References

7.1 System design and implementation of third party logistics management based on electronic commerce

In view of the current confusion fusion problem of e-commerce and logistics management system, puts forward a conception of integrated B/S mode of the third party logistics management system based on J2EE, and taking as architecture to carry on the detailed design to the system, the realization of distribution management, including scheduling management, function. [1]

7.2 Design and realization of logistics management system based on uml modeling technology

This paper briefly introduces the UML modeling technology, describes the design and the modeling process of logistics management system and applies the B/S pattern and the.NET technology, SQL sever database, it also discusses the realization method of logistics management system based on the environment of the Internet. [2]

References

- [1] LI Min. System design and implementation of third party logistics management based on electronic commerce. *Electronic Design Engineering*, (19):107–109, 2015.
- [2] D. U. Wen-Jing. Design and realization of logistics management system based on uml modeling technology. *Journal of Shenyang Normal University*, 2011.

8 Contributions of team members

- Zeng Jiaying: Diagrams organized with use case realization, analysis mechanism
- Li Yichao: Diagrams organized with use case realization, introduction, type setting
- Liu Lidong: Diagrams organized with use case realization, high-level architecture

- Yi Siqu: Diagrams organized with use case realization, updated user interface

Every member in our group makes equal contribution.