



Onion Express

Software Requirements Specification

Yang LI
Zhongjin LUO
Guohui YANG
Yirui WANG
Xinying WU
Yiqun LIN

School of Software Engineering
Tongji University
Group 4

April 6, 2017

Contents

1	Introduction	3
1.1	Purpose	3
1.2	Definitions	3
1.3	System Overview	3
2	Use Case Modelling	5
2.1	Activity Diagram	5
2.2	Use Case Diagram	5
2.2.1	Global Use Cases	5
2.2.2	Sub Use Cases	6
2.2.3	Specification of Use Cases	6
3	Glossary of Terms	7
4	Supplementary Specification	8
4.1	Security	8
4.1.1	Access and Data Integrity	8
4.1.2	Encryption	8
4.1.3	Digital Certificates	8
4.1.4	Digital Signatures	9
4.2	Performance	9
4.3	Data Storage and Computing	10
4.4	Track the Package	10
4.5	Maintenance	10
4.6	Others	11
5	User Interface	12
5.1	Mobile Devices(iOS)	12
5.2	Website	12
6	Contributions	13

1 Introduction

Onion Express® is a system built for logistic companies, which provides them with a solution to logistics tracking, goods packing, goods distribution, after-sales management, data storage, information processing, etc.

1.1 Purpose

These days as B2C business is increasing rapidly, the growth of logistics business is also remarkable. The enormous market demand brings logistics companies opportunities as well as the challenge. Facing such kind of condition, this project is aimed at improving the efficiency of field personnel and customer satisfaction of a logistics company by building a cross-platform system.

1.2 Definitions

As Jobs has ever said, “People don’t know what I really want at all, until your products are in their eyes”. This project is specially designed for an independent logistics companies like UPS. The business scope is limited within China. To be more precise, the express is only available in Jiangsu, Zhejiang and Shanghai at the beginning. Temporarily private orders are not covered in the business scope, which means the express company corporates with e-commercial companies only with the cash-on-delivery express or normal express. The system focuses on logistics service without regard to O2O, bulk cargo or self-support e-business. Timing express might be expanded in future.

1.3 System Overview

The actors in the system are classified as *Postman*, *E-business*, *Customer service*, *Customer* and *Agent*. *Customer* and *Agent* are generalized as Receiver. The *Postman* has access to this system only on mobile devices while *Customer* has access both on browsers as well as mobile devices. *E-business* offers orders periodically. *Customer service* helps to deal with tasks cannot be done only by the system.

Web application and iOS application provide different functions for different users to enhance user experience and have some humanization design(e.g. using different colors to mark tasks as reception or delivery in postman’s app). Besides basic functions, the system also

provides some advanced functions, like printing invoices. Different offline payment methods are supported. And the customer's telephone number is hidden to protect his/her privacy. The postman is equipped with a multifunctional special device, when the customer receives his/her package, he/she can use this device to pay by card and can also press thumb on it to sign digitally, besides, the device helps collect postman's GPS location accurately. The system considers all the 8 scenarios, including sending the package, paying for the product, signing the package and so on. To integrate the system, two scenarios are added. One is creating the orders, at the beginning of the entire flow. Another is dealing the order manually, to reduce errors caused by the system and handle other unanticipated situations. That can improve the stability of the system and in consideration of the relatively small scale of users in the early stage, robot customer service is not necessary. It can be taken into consideration when the business is expanding to a certain stage. This project also designs several user interface mock-ups on the website and on mobile devices. Core functions are exhibited in these mock-ups, for example, the dispatch list interface. Nonfunctional requirements and further explanations on security, performance, data storage and computing, tracking the package, maintenance and others are detailed in supplementary Specification.

2 Use Case Modelling

2.1 Activity Diagram

2.2 Use Case Diagram

2.2.1 Global Use Cases



Onion Express

Figure 2.1: Global Use Case Diagram

2.2.2 Sub Use Cases

Payment View

Logistics Company View

2.2.3 Specification of Use Cases

Scenario 1

Scenario 2

Scenario 3

Scenario 4

Scenario 5

Scenario 6

Scenario 7

Scenario 8

3 Glossary of Terms

4 Supplementary Specification

4.1 Security

The system should avoid the database being attacked and data being taken advantage of by the wicked.

4.1.1 Access and Data Integrity

1. The authorization of access to the system of postmen, customers and customer servers should be classified and announced clearly. With certain authorization, different users have limited access to data and operation.
2. The server should use anti-virus software.
3. Firewalls and network protection are necessary, and they should be updated in time.
4. The atomic processes in the database will ensure the accuracy of the database.

4.1.2 Encryption

1. The session should not be transmitted in DNS.
2. All texts and messages should be encrypted with Encryption Algorithm such as RSA, 3DES or IDEA.
3. Two keys are used to identify a certain user. One public key is used for encryption and another private key is used for decryption. The key is a completely random mix of letters.
4. The session will record the activity of the customer, and if the customer has no operation for 5 minutes, he or she will log out the system automatically.
5. After customers log out the system, all the private information(cookies) will be cleaned.

4.1.3 Digital Certificates

1. We use digital certificates as a replacement of user names and passwords, for example, SSL Certificates. It will be used automatically with the permission of users.

2. The IP address or location where users log in the system will be recorded and when the account is used beyond their regular locations, the user will get alarmed.

4.1.4 Digital Signatures

1. Users should log in the system with a password. Our system will test its complexity. If it is too simple, the system will remind the users to complicate it. That involves cryptography.
2. We use a message digest to ensure the integrality of the data.
3. If necessary, we can extend our fingerprint system to login system.

4.2 Performance

1. The information of the package, including the real-time position, Order-ID, the postman etc., should be checked by customers in 3 seconds with at most 0.1% error rate.
2. The payment should be confirmed in 2 seconds by the system from the moment when the third party trade agent sends the message or the postmen report the payment.
3. The order created by customers should be processed in 15 minutes.
4. The orders obtained from e-business should be processed every hour(about 5,000 orders).
5. Information of the delivery such as the phone number, the address, the receiver and others should be updated and checked by postman in 1 min.
6. This system allows the e-business to create batch orders which can be sent at regular time.
7. The estimate of delivery time should be accurate with the max uncertainty in 2 days.
8. The expectation should be sent to custom service in 2 min from the time a postman reports it.
9. This system's unavailable time should be controlled in 20 minutes in a year.
10. To offer the best user experience, a content delivery network should be used by this system.

4.3 Data Storage and Computing

1. To store a huge amount of data, distributed database should be used. And it should use Homogeneous Distributed Databases Management System.
2. Considering that there may be an enormous number of visitors and inquiries at the same time, the system must implement cloud computing service.
3. The system can support as many as 1500 times of visits per second.
4. There must be a copy of the database, including device entity, software, data and even employees, in order to prevent some unpredictable disasters.
5. If the database is destroyed, the copy should be enabled in 3 hours.
6. The data can be in English, Chinese, Japanese, French and Korean.

4.4 Track the Package

1. In order to track the package, the GIS system should be applied, with the help of the GPS system. The system gets geographic information from a third party system, and get the position of postmen who deliver the package through the system of postmen. And this system should match both kinds of the information and show it to users of the system.
2. The system for postmen should upload the position of the postman automatically every 2 hours, through 3G, 4G or WLAN network.
3. If the locations of postmen are missing for 4 hours, the system should inform the custom servers, and custom servers will contact with postmen.

4.5 Maintenance

1. The distributed database should be maintained by the employees of our own company including the employees of the standby database every day when the visiting traffic is not heavy.
2. The software for custom service, customer and postmen and the system itself should be maintained by our employees.
3. The geographic information source should be multiple, in case that one of the sources is unavailable.
4. The engineers from the company offered DBMS will maintain our system every year.
5. An integrated scheme to deal accidents, for example, the crash of database, is necessary.

4.6 Others

1. The architectures of the postman APP and the customer APP are B/S and C/S, but that of custom service is C/S for safety.
2. Our system can be used in iOS and Android on mobile devices and in a normal browser on PC(Windows/macOS/Linux).
3. Anticipated development time is two months.

5 User Interface

5.1 Mobile Devices(iOS)

5.2 Website

6 Contributions

Visit more on [GitHub](#)

1452559 Yang LI	iOS UI, Document	17%
1453645 Zhongjin LUO	Use Case	17%
1451229 Guohui YANG	Use Case	17%
1552651 Yirui WANG	Use Case, Activity Diagram, Review	17%
1552677 Xinying WU	Web UI	17%
1552705 Yiqun LIN	Use Case	17%

Bibliography

- [1] 830-1998, *IEEE Recommended Practice for Software Requirements Specifications*, IEEE, Oct 1998.
- [2] 29148-2011, *Systems and software engineering – Life cycle processes –Requirements engineering*, ISO/IEC/IEEE International Standard, Dec 2011.
- [3] Russ Miles, Kim Hamilton, *Learning UML 2.0*, O'REILLY, 1st edition, April 2006.
- [4] Jim Arlow, *UML 2.0 and the Unified Process: Practical Object-oriented Analysis and Design*, ADDISON WESLEY, 2nd edition, 2005.
- [5] Karl Eugene Wiegers, Joy Beatty, *Software Requirements*, Microsoft Press, 3rd edition, 2013.
- [6] Craig Larman, *Applying UML and Patterns*, Pearson Education International, 3rd edition, 2005.
- [7] Simon J. Bennett, Steve McRobb, Ray Farmer, *Object-oriented Systems Analysis and Design Using UML*, McGraw-Hill Education, 2nd edition, Dec 2001.
- [8] Yunjie TAN, *Thinking in UML*, China Water Conservancy Hydropower, 2nd edition, March 2012.