



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
3	Glossary of Terms	6
4	Supplementary Specification	7
4.1	Security	7
4.2	Performance	7
4.3	Data Storage and Computing	7
4.4	Track the Package	7
4.5	Maintenance	7
4.6	Others	7
5	User Interface	8
5.1	Mobile Devices(iOS)	8
5.2	Website	8
6	Contributions	9

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

3 Glossary of Terms

4 Supplementary Specification

4.1 Security

4.2 Performance

4.3 Data Storage and Computing

4.4 Track the Package

4.5 Maintenance

4.6 Others

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.