OpenBazaar Redevelopment - Requirements

The Fair Traders
Daniel Mandel - mandeldr
Shandelle Murray - murras25
Connor Sheehan - sheehacg

December 8, 2015

Abstract

This documents outlines requirements for the OpenBazaar redevelopment project.

Table of Contents

Revision History

Revision Number	Revision Date	Description of Change	Author
1	November 2nd, 2015	Created Revision History	Daniel Mandel
2	November 10 - November 20, 2015	Add non-functional requirements	All members of group
3	November 25, 2015	Section 4.1 - Kademlia and Ricardian Contract defined	Shandelle Murray
4	November 28, 2015	Sections 6.2 and 6.3 - Add context diagram and work partitioning table, create lists of figures and tables	Shandelle Murray
5	December 8, 2015	Add use case diagram, remaining non-functional requirements, and last sections	Shandelle Murray

Table 1: Table to capture the history of the document

References

We have used the Volere Template as a guide for creating this requirements document.

http://docs.openbazaar.org/ was also referenced for material regarding the OpenBazaar.

Project Drivers

1 The Purpose of the Project

1.1 The User Business or Background of the Project Effort

The modern economic era is built around e-commerce and internet trade. This is apparent from the change in the speed of stock market trades, the explosion of technology based corporations and the expansion of internet commerce services such as Alibaba and eBay. Currently, people who wish to buy and sell online are largely constrained to utilizing the services offered by the large corporations, thereby sacrificing a portion of the profit from trades. In undertaking the OpenBazaar project, we aim to benefit both online buyers and sellers by creating a platform in which internet trade can be decentralized The project will be developed as an open-source, peer-to-peer network.

1.2 Goals of the Project

The main goals of the project include:

- Creating an online marketplace that is scalable, free of intermediaries and their fees, and cannot be censored.
- Eliminating the need for centralized e-commerce services and websites.
- Reducing the overhead cost of doing business and trading over the internet by using the software which will essentially make trade free again.
- Creating a permission-less, censorship-resistant trade platform that will connect the entire world.

2 The Stakeholders

2.1 Traders

At the present time, anyone who wishes to open an online store must use a centralized service. These services often charge listing fees, subscription fees or membership fees. Traders are also forced to use centralized exchange platforms such as PayPal or be charged bank fees for direct deposits. Traders stand to benefit from the project by the elimination of both of these unnecessary expenditures. The use of BitCoin will allow for a fee-less monetary exchange and a free product listing on the OpenBazaar network.

2.2 Buyers

Buyers who shop online will benefit from this project in several ways. The overhead costs of doing trade will be lower on this platform than centralized services, and buyers should expect to see a reflection of this in the prices of products on OpenBazaar. Buyers will be free to exchange goods with anyone they can connect to on the network,

2.3 Other Stakeholders

Other stakeholders include:

- Major corporations that currently benefit from trades between buyers and sellers through the internet
- Collectively, law enforcement can be considered a stakeholder as they will be affected by this new form of online trade and will likely have to alter their tactics for detecting illegal online activity
- Members of the development team
- Computer/internet users in general may be considered stakeholders because, with a simpler and more effective manner of completing sales and trades readily available, more of these people may turn to internet trading

2.4 The Hands-On Users of the Product

The hands-on users of the product:

- Online Sellers/Traders
- Online Buyers
- Computer/internet users interested in buying and selling online

2.5 Priorities Assigned to Users

- Key Users: Online buyers and sellers
- Secondary Users: Developers and testers

2.6 User Participation

- Users acting as prospective buyers, sellers, or an anonymous, third-party mediator access the OpenBazaar network
- Users acting as sellers advertise their products on the OpenBazaar network
- Users acting as buyers browse or search for products that they would like to buy on the OpenBazaar network
- Users acting as notaries advertise their mediation services on the OpenBazaar network and serve as a third party to ensure a fair trade

2.7 Maintenance Users and Service Technicians

- Developers and Testers

Project Constraints

3 Mandated Constraints

3.1 Solution Constraints

- Description: The OpenBazaar client will run on Windows, Linux and Mac OS X.
- Rationale: These are three of the most common desktop software platforms available.
- Fit Criterion: The required framework and programming language must be installed (PyQT 4, Python 2).
- Description: In order to have full functionality a working internet connection is necessary.
- Rationale: The internet is the fastest way to connect buyers and sellers around the globe and exists in most modernized countries.
- Fit Criterion: It is required to make transactions, view markets, and discover peers on the network.
- Description: To make trade completely decentralized, as well as entice buyers and sellers to connect over the OpenBazaar marketplace, Bitcoin must be used as a currency.
- Rationale: It is the easiest as well as one of the safest ways to make transactions over the internet. it is also becoming more and more accepted in other retail and online stores.
- Fit Criterion: Users must have a Bitcoin wallet installed on their computer.

3.2 Implementation Environment of the Current System

- The application will be developed on Ubuntu 14, using Python 2, and PyQT as the GUI framework. The framework was chosen because of its cross-platform abilities and versatility. Ubuntu was chosen because of the compatibility it has with the existing off-the-shelf softwares (OpenBazaar Server), Git, and the other partner applications that make OpenBazaar function.

3.3 Partner or Collaborative Applications

- BitCoin will be a vital application serving as the medium of exchange on OpenBazaar.

3.4 Off-the-Shelf Software

- There is an existing off-the-shelf software, but it is in the beta development phase, with development focus on the front end (back-end complete) and testing. The front end or client side of the application will act similarly to an online classifieds system like Ebay, Amazon, Kijiji, Craigslist; however, it will be populated with only what a user would like to see.

3.5 Anticipated Workplace Environment

- This system is intended for use anywhere that there is a working internet connection. This enables users to connect from all around the world to buy and sell products and services. Virtually, the anticipated workplace environment is the entire civilized world.

3.6 Schedule Constraints

- This project should be completed and tested by November 30, 2015
- Learning should be focused on the PyQt4 framework, and creating wireframes for the GUI.
- Final documentation must be complete by December 8, 2015.

4 Naming Conventions and Terminology

4.1 Definitions of All Terms

- Python: a widely used and versatile high level programming language
- IDE: integrated development environment
- PyCharm: the chosen IDE for the project
- GUI: Graphical User Interface
- Git: source control for the project, includes features such as revision history
- Bitcoin: digital store of value and online payment system

- Ricardian contract: used for the purpose of issuing digital currency, it is a document which states the terms under which a value is redeemable. It is readable by people, parsable by programs, and prevents unauthorized changes by making use of digitally signed and checksum hashed contracts. The contract carries keys and server information, and requires digital signatures of the issuer, a holder, and possibly a third-party notary in order for a value to be redeemed.
- Kademlia DHT: a distributed hash table for decentralized peer-to-peer computer networks that is designed to minimize the number of configuration messages nodes must send to learn to about other nodes.

5 Relevant Facts and Assumptions

5.1 Assumptions

- It is assumed that any user of the application understands the Bitcoin currency.
- It is assumed that end users understand how a peer-to-peer application is different from the traditional client-server model.

5.2 Facts

- There are existing frameworks for GUIs. PyQt in particular will be used for the implementation of the client side.
- We will be focusing on developing the client side of the application due to the time constraint of this project.

6 The Scope of the Work

6.1 The Current Situation

Content A software application to connect users from all around the world to conduct trade freely is required. Users can add any peers to the network, as well as view their stores, and search for items to purchase. They will also have the ability to have their own store to sell items.

6.2 The Context of the Work

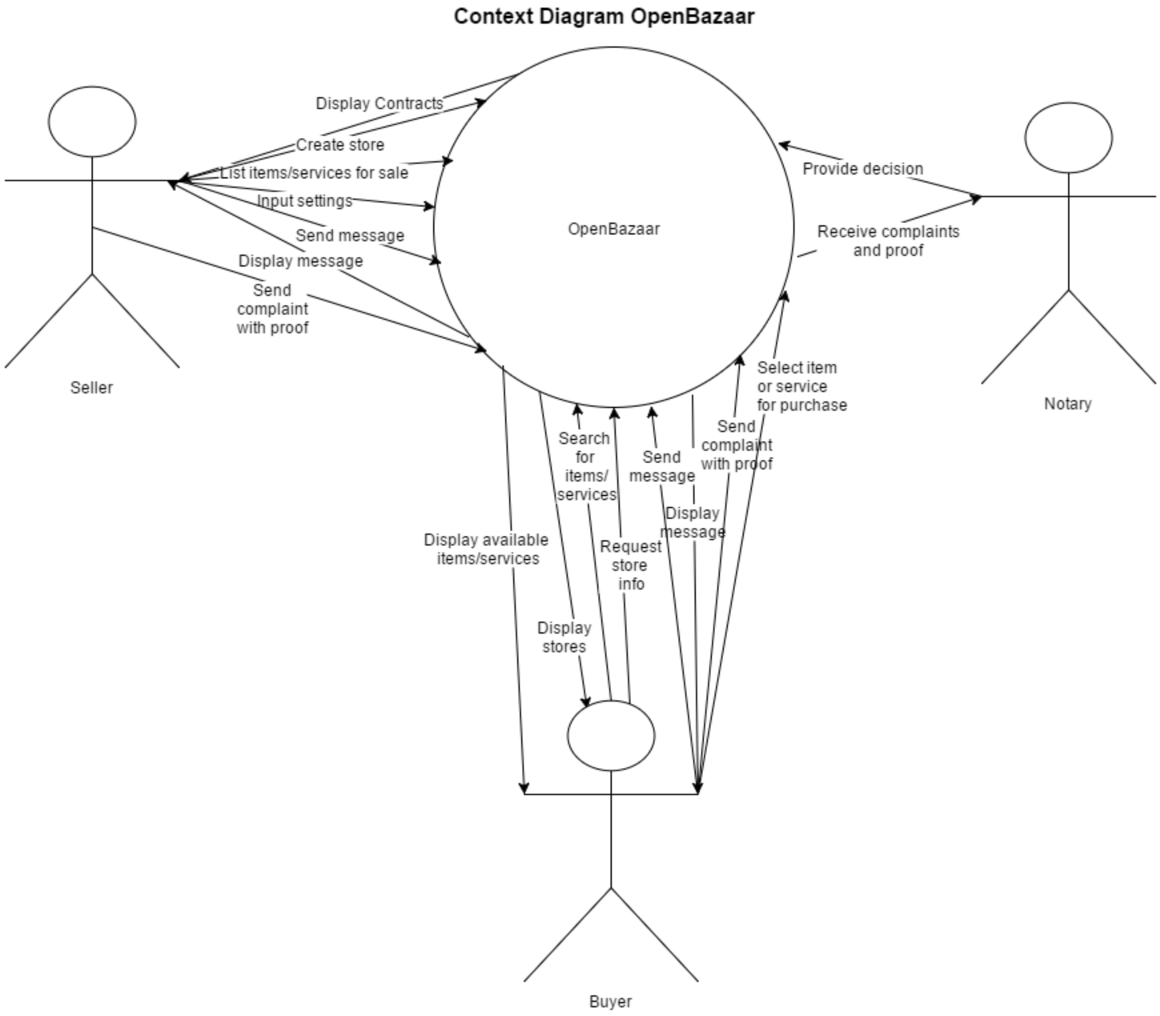
Context Diagram OpenBazaar
Display Contracts
Create store
List items/services for sale
Input settings
Send message
Display message
Send complaint with proof
Seller
OpenBazaar
Provide decision
Receive complaints and proof
Notary
Select item or service for purchase
Send complaint with proof
Search for items/ services
Send message
Display message
Display available items/services
Request store info
Display stores
Buyer

Figure 1: Context Diagram OpenBazaar

6.3 Work Partitioning

Event Name	**Input and Output**	**Summary**
1.User inputs store information	Store Information (IN)	Seller saves store information
2.User inputs new contract information	Contract (IN), Contract (OUT)	Seller lists new item for sale and it is listed in their store
3.User inputs settings	Settings (IN)	User specifies personal or store settings
4.User sends complaint with proof	Complaint (IN), Complaint (OUT)	Buyer or seller generates complaint and notary receives it
5.User sends messsage	Message (OUT), Message (IN)	User sends message and another user receives message
6.User searches for listings	Search Criteria (IN), Available Listings (OUT)	User searches for avaiable listings and application displays them
7.User searches for store	Store ID (IN), Store (OUT)	User searches for store and application displays store
8.User requests to purchase listing	Purchase Request (IN), Updated Contract (OUT)	User requests to purchase item or service and application advances contract to next step
9.User provides decision about complaint	Decision (IN), Decision (OUT)	Notary makes ruling in favor of eithe party and application proceeds to complete the transaction according to the contract

Table 2: Table to capture the inputs and outputs of an event

7 The Scope of the Product

7.1 Product Use Case

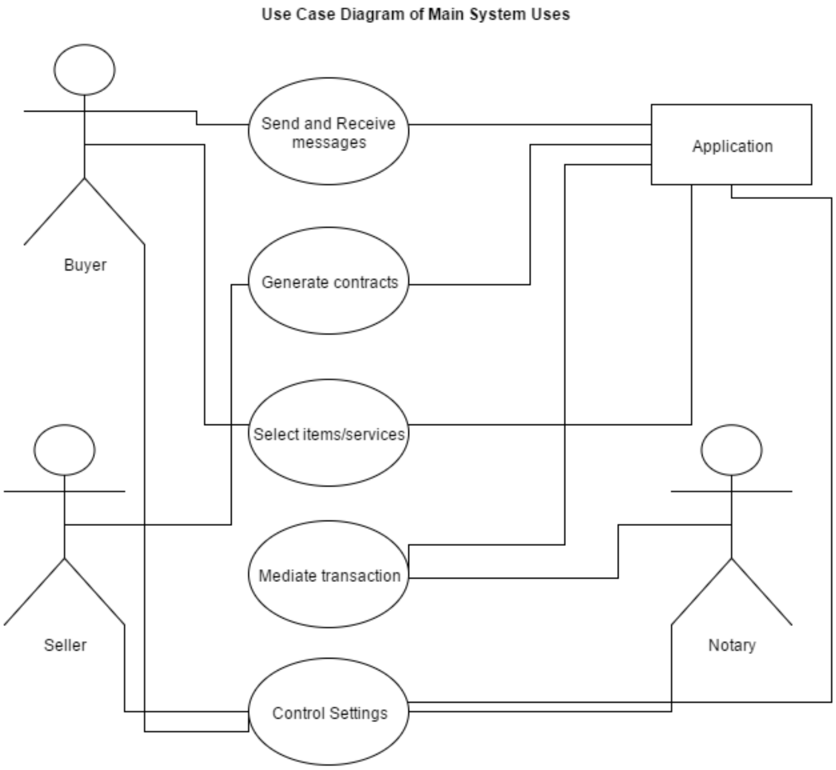

Figure 2: Use Case Diagram

Functional Requirements

R1 The application shall simulate trade between two people.

R2 Users shall have the ability to list an item for sale on the application.

R3 Users shall have the ability to find and search through items for sale on the application.

R4 Users shall have the ability to access a seller's store.

R5 Users shall have the ability to formally define trade contract terms through a Ricardian contract.

R6 Users shall have the ability to list any good or service they are in a position to offer for trade.

R7 Buyers and sellers shall have the ability to find and choose notaries to oversee the fairness and completeness of transactions.

R8 Users shall have the ability to select an avatar.

R9 Users shall have the ability to upload images along with the item they wish to sell.

R10 Users shall have the ability to view existing notaries and merchants.

R11 Users shall have the ability to attach keywords to a contract so that it can be found with search.

R12 Users shall have the ability to view a notary's store.

Non-functional Requirements

8 Look and Feel Requirements

8.1 Appearance Requirements

The look and feel of the application should be on par with existing internet commerce services, such as Alibaba and eBay.

8.2 Style Requirements

- The layout of the application should be organized logically.

9 Usability and Humanity Requirements

- The platform should be deployable by a person with little to no technical computer knowledge.
- The platform should be accessible by any person who could access a marketplace in the real world. For instance, anyone should be able to access the platform by their mid-teens.

9.1 Personalization and Internationalization Requirements

- Sellers should be able to customise the theme of their store to fit their products, personal preference or any other design choice they make.
- All text displayed on the interface should be translatable into multiple different languages for deployment to different geographical regions.
- The interface should be resizeable to allow all users a comfortable buying and selling experience.

9.2 Learning Requirements

- The interface should have little to no learning curve for full use of the platform.
- Sellers should be able to easily create a store and list products.
- Buyers should be able to find specific products easily.
- Notaries should be easy to find by both buyers and sellers.

9.3 Accessibility Requirements

- Text on the interface should be readable by all persons, including the colour-blind and persons with imperfect vision.
- The interface should be traversable with the tab button, to allow persons with hand tremors or other disabilities to access the marketplace.

10 Performance Requirements

10.1 Speed and Latency Requirements

- The application should be able to locate other nodes on the network within a reasonable time frame such as () seconds.

10.2 Precision or Accuracy Requirements

- Prices attached to listings should be precise to the equivalent of no smaller than 1 satoshi which is 0.000 000 01 of a bitcoin.

10.3 Reliability and Availability Requirements

- The application should be available at all times as long as the user has access to an uncensored internet connection.

10.4 Robustness or Fault-Tolerance Requirements

- Data should not be lost by the application once a process has reached a point of completion at any phase (seller has created contract, buyer has made a motion to buy item listed, buyer has agreed that the product has been received, etc).

10.5 Capacity Requirements

- The peer-to-peer network should support use by millions of peers concurrently.

10.6 Scalability or Extensibility Requirements

- The application will be implemented as an easily scalable network of nodes that run the trade protocol. In the early days of the application the network will have few known nodes, but as more nodes are added the network will be able to expand.

10.7 Longevity Requirements

- Any listings should be available as long as the node that posted it is online. If the node goes offline and returns online, the information should still be retained and the listing once again available on the network.

- Settings and personalization information should be available each time the node reconnects to the network.

11 Operational and Environmental Requirements

- The application is currently expected to run on the Linux operating system; however, in the future it will be expected to run on Windows and Mac operating systems as well.

11.1 Expected Physical Environment

- The application is expected to run at any location in the world where there is a computer that can make an uncensored internet connection.

11.2 Requirements for Interfacing with Adjacent Systems

An Application Programming Interface (API) will be provided publicly for easy integration of stores and listings into existing websites, such as classifieds and personals. This API will be implemented in a common web environment (Javascript, PHP etc) and return HTML5 frames for up-to-date display of data. For example, if the listing is displayed on a personal via the API, when the listing is taken off the network the personal ad should reflect this.

11.3 Production Requirements

- Future releases of the application shall have more functionality related to the networking components.

- The application should be easily installable by a person with enough technical knowledge to list or by items/services in an online marketplace.

12 Maintainability and Support Requirements

12.1 Maintenance Requirements

- An update system will need to be implemented to ensure that security bugs, vulnerabilities, and attack vectors are patched quickly on all network nodes.

12.2 Supportability Requirements

- The final application shall not be supported after the completion of the project; however, the source code and will be available for viewing and editing.

12.3 Adaptability Requirements

- The application is expected to run on Linux Ubuntu 14 currently, but will eventually be required to run on Windows and Mac operating systems as well.

13 Security Requirements

13.1 Access Requirements

- Users should only have access to view and edit items for which they are authorized. For example, a user should not be able to list items in another user's store.

13.2 Integrity Requirements

- The application should be secure against unauthorized attempts to alter or fabricate data such as unauthorized attempts to change contract information or attempts to fake network identities.

13.3 Privacy Requirements

- Users should be permitted to add as much or as little personal information to the application as they would like. All information added by the user should have a privacy setting which ensures the data is not released to any node that does not have permission to view it.

13.4 Audit Requirements

Not applicable

13.5 Immunity Requirements

Not applicable

14 Cultural Requirements

14.1 Cultural Requirements

Not applicable

15 Cultural Requirements

15.1 Cultural Requirements

- The application itself should not display any culturally degrading images or phrases. This does not pertain to any information that users are able to input as there are no restrictions in place to control listings or descriptions.
- The application should eventually be available in all languages with which users may consider their official language.

16 Legal Requirements

- The application shall accomodate legal intervention that law enforcement deems necessary such as in the case of illegal activity taking place through the network.

16.1 Compliance Requirements

Not applicable

16.2 Standards Requirements

Not applicable

Project Issues

17 Open Issues

Currently, the networking component of the application has not yet been implemented. Due to the timeframe of the project, the scope has been narrowed to focus on the GUI and the personal/local portions of the application with mock data added in for testing and demonstration purposes. Due to the constraints of the library being used for peer to peer network, this is difficult to implement.

Additionally, the application currently only runs on Linux Ubuntu 14 which will have to be modified in the future in order to increase the portability of the application.

18 Off-the-Shelf Solutions

There is an open sourced version OpenBazaar currently being developed which is set to be realeased by late 2015 or early 2016.

19 New Problems

Not applicable

20 Tasks

- Implement networking component
- Make executable on multiple operating systems

21 Migration to the New Product

Not applicable

22 Risks

There are security risks involved in online trade and transactions that will have to be further tested. Additional mechanisms may have to be put into place and verified.

23 Costs

Not applicable

24 User Documentation and Training

There is not currently a user training manual; however, there is a text file available outlining the proper installation procedures.

25 Waiting Room

Not applicable

26 Ideas for Solutions

Not applicable

List of Figures

List of Tables