Design Document For Online Bartering System

Initial Design and Proof of Concept

Sally Moen, Michele Tokuno, Timothy Cooper, Brady Hyde

MARCH 13, 2016

TABLE OF CONTENTS

OVERVIEW:

Introduction

This software design document will describe the design of a basic online barter system. This document includes both narrative and graphical descriptions of the elements of the barter system.

Purpose

The purpose of the Barter Application is to facilitate the exchange of goods and/or services between two parties without the use of currency. Parties include both individuals and businesses, including employees acting on behalf of a business.

Scope

The scope of this document is a proof of concept for a basic application that enables the creation of a trade between two users. This document includes the layout for the first iteration of this application as well as diagrams for data models that we will be using in the application. Using this application, members will be able to do the following:

1. Search for compatible trades nearby
2. Create and negotiate a trade with another user
3. Create and edit a profile that includes what they have to offer and what they are searching for

The proof of concept website will include a basic, but attractive layout and page design. The website will initially be deployed to Azure for demonstration purposes. Should this be released to the broader public, a Terms of Service document that prohibits illegal or illicit transactions will be drafted and published on the site.

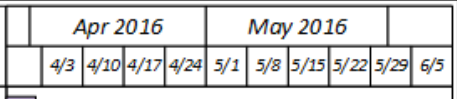
DEFINITIONS/GLOSSARY

- Barter: Exchange of goods and/or services in lieu of legal tender. Used interchangeably with transaction.
- Business: May include any organization that has more than one user account within. Including: for-profit businesses, non-profit organizations, etc.
- Haves: Collection of items (and quantities or durations) owned and up for trade by a Trader.

- Trader: Individual or Business with a trading account.
- Wants: Collection of items (and desired quantities or durations) desired by a Trader.

TIMELINE

ID	Task Name	Start	Finish	Duration
1	Create skeleton	3/28/16	4/1/16	5d
2	Start models	3/28/16	4/1/16	5d
3	Start basic controllers	4/4/16	4/8/16	5d
4	Create/edit scaffolding	4/4/16	4/8/16	5d
5	Start views	4/4/16	4/8/16	5d
6	Start implementing Login features	4/11/16	4/15/16	5d
7	Create/edit working UI	4/11/16	4/15/16	5d
8	**Milestone: Have Basic Alpha Version**	4/18/16	4/29/16	10d
9	Create Profile	4/18/16	4/22/16	5d
10	list haves/wants	4/18/16	4/22/16	5d
11	search	4/25/16	4/29/16	5d
12	create offers	4/25/16	4/29/16	5d
13	accept offers	4/25/16	4/29/16	5d
14	Messages	5/2/16	5/6/16	5d
15	Administrative users/roles	5/2/16	5/6/16	5d
16	Begin writing Unit Tests	5/2/16	5/6/16	5d
17	**Milestone: Complete Beta Version**	5/9/16	5/20/16	10d
18	Complete final UI	5/9/16	5/20/16	10d
19	Improve/refine Search	5/9/16	5/13/16	5d
20	Improve/refine Messages	5/9/16	5/13/16	5d
21	Improve/refine Offers System	5/16/16	5/20/16	5d
22	Improve/refine Profiles	5/16/16	5/20/16	5d
23	Final Testing	5/23/16	5/27/16	5d
24	Final Debugging	5/23/16	5/27/16	5d
25	Basic Usability Testing	5/23/16	5/27/16	5d
26	**Milestone: Initial Release Version**	5/30/16	6/3/16	5d
27	Deploy to Server	5/30/16	6/3/16	5d
28	Create Final Presentation	5/30/16	6/3/16	5d
29	**Milestone: Present to Client**	6/6/16	6/6/16	1d



REQUIREMENTS

Functional

- User will be able to create, edit and maintain a profile, including their Haves and Wants.
- System will authenticate and authorize users based on account credentials.
- User will be able to search for nearby Traders and offered items.
- Users will be able to create and negotiate barters with other Traders.
- Users will be able to confirm or cancel a finalized transaction.
- Users will be required to submit feedback and ratings for completed and confirmed transactions.
- Users may file complaints about other users or transactions that violate the law and/or terms of service.
- Site administrators will be able to view and respond to user complaints, and temporarily or permanently ban users in response to violations.
- Businesses may invite and enable their employees to use limited-permission trading accounts.

Non-functional

- Clean user interface
- Mobile responsive
- Will be supported by most popular modern browsers (Chrome, Firefox, Safari, Opera, Internet Explorer (most recent version))

USE CASES

Actors

1. *Trader (or Non-Administrative User)*: A Trader is a generic public user that uses the system to conduct barter transactions. There are two types of trader users:

    a. *Business Trader*: A Business Trader may be an individual or a collection of individuals associated with a business entity.

    b. *Non-Business Trader*: A Non-Business Trader is an individual and may not be a collection of individuals.

2. *Administrative User*: An Administrative User is a user responsible for overseeing and maintaining the system and user accounts.

Brief use case descriptions

Trader Use Cases:

Use Case	Description
Create Profile	User enters personal data such as desired username and location information.
Update Profile	User views personal data and changes information as desired.
Delete Profile	User selects option to delete own profile and confirms deletion.
List Wants	User enters goods or services for which the user would like to barter.
List Haves	User enters goods or services the user would like to offer in a barter transaction.
Create Barter (Transaction)	User discovers that a second user's "want" matches the first user's "have" or vice versa, selects the relevant goods and services in the profiles, and initializes a transaction request by pressing a "Barter" button (which is only enabled once transaction goods and services are selected).
Cancel/Void Barter (Transaction)	User selects a cancel/void option on the barter transaction page, enters reason for cancellation, and confirms the transaction cancellation.
Complete Barter	User views barter transaction request details and selects a "Transaction

(Transaction)	Completed" option upon completion of the exchange of goods or services.
Rate Transaction	After completing a transaction, the user is prompted to rate the completed transaction on a scale of 1 to 5.
Browse Profiles	User views a list of all profiles.
Filter Profiles	User selects profile filters on the browse profile page and views profiles conforming to the selected filters.
Browse Wants	User views a list of all goods and services desired by other users.
Filter Wants	User selects desired "wants" filters on the "Browse Wants" page and views "wants" conforming to the selected filters.
Browse Haves	User views a list of all goods and services on offer.
Filter Haves	User selects desired "haves" filters on the "Browse Haves" page and views goods and services on offer that conform to the selected filters.
Search Profiles	User enters search term and views a list of all matching profiles.
Search Wants	User enters search term and views a list of all matching goods and services desired by other users.
Search Haves	User enters search term and views a list of all matching goods and services others are offering for barter.
Report User	User notices another user's profile that may violate the terms of service, selects the "report user" option on the other user's profile page, and enters additional information about the type of violation before submitting the report to an administrator for review.
Report Barter (Transaction)	User selects a "report transaction" option on the barter transaction page, enters additional information about violation type, and submits the report for review after encountering a possible violation of the terms of service during the course of a transaction.
Save Search	User selects a "save search" option on a search results page.
View Saved Searches	User views a list of past searches the user chose to save.

View History	User views a list of the user's own past and pending barter transactions.
View Ratings	User views a list of the barter transaction ratings given by all users for a particular user.
Create Message	User selects the "Send Message" option on a user profile page and composes a message to another user in a message composition form which may be sent or saved as a draft.
Send Message	User selects the option to submit a completed message composition form.
View Received Messages	User views a list of received messages, each of which has an option to view the full message and message details.
View Message	User views the full message and relevant details of a specific message.
View Sent Messages	User views a list of sent messages, each of which has an option to view the full message and message details.
Delete Message	User selects a message to delete, then submits and confirms deletion request.
Save Message	User selects a "Save" option on the message composition form page.

Administrator Use Cases:

Edit User Profile	Admin user views non-admin user profile data and updates information as desired.
Delete User	Admin user views non-admin user profile data, selects a "Delete User" option, and confirms deletion.
View/Browse Users	Admin user views list of non-admin user profiles.
Filter Users	Admin user selects desired non-admin user filters on the "View/Browse Users" page and views a list of only the user profiles conforming to the selected filters.
Search Users	Admin user enters search term in a search form, submits the form, and views a result set of users matching the search term.
Ban User (Time-Limited)	Admin user selects a "Ban User (Time-Limited)" option on a non-admin user's profile page, enters the ban time frame, and enters the reason for the ban.
Ban User (Forever)	Admin user selects a "Ban User (Forever)" option on a non-admin user's profile page and enters the reason for the ban.
View Reports	Admin user views list of unreviewed reports.
Review Report	Admin user selects a report not marked as reviewed, reviews the report details, and selects an appropriate action from a list of options.
View Archived Reports	Admin user views a list of reports that have already been reviewed and dealt with.
Filter Archived Reports	Admin user selects filters and views a list of reviewed reports that conform to the selected filters.
Review Barter Details	Admin user selects a barter transaction and views the transaction details.
Create New Admin	Admin user enters new admin user information.
Edit Admin	Admin user views existing admin user profile and updates information as desired.
Edit/Update Transaction	Admin user views a transaction in progress between two users, edits the desired fields, and saves the changes.

Detailed Use Case Diagrams

Create Profile Detailed User Case Description:

Use case name:	*Create profile.*	
Triggering event:	Trader wishes to start using the barter application.	
Brief description:	Trader enters personal data such as desired username and location information.	
Actor(s):	Non-Business Trader	
Related use case(s):	This use case includes the “list haves” and “list wants” use cases.	
Stakeholder(s):	Non-Business Trader, Business Trader	
Precondition(s):	Trader must be willing to divulge basic personal information. Trader must have goods and/or services to offer other users as well as goods and/or services that the user desires.	
Postcondition(s):	Trader profile must be successfully created. Trader profile information (including initial wants and haves) must be accurate and saved to the database.	
Flow of activities:	Actor (Non-Business Trader)	System
	1. Trader selects a “Create Profile” option and enters necessary profile information into profile creation form (including initial wants and haves). 2. Trader submits profile creation form.	1.1 System validates trader input. 1.2 System checks that the trader does not already have a profile. 2.1 System creates a Trader object that contains the entered trader input.

Exception condition(s):	1.1 Information entered by trader is not valid. 1.2 Trader already has a profile.

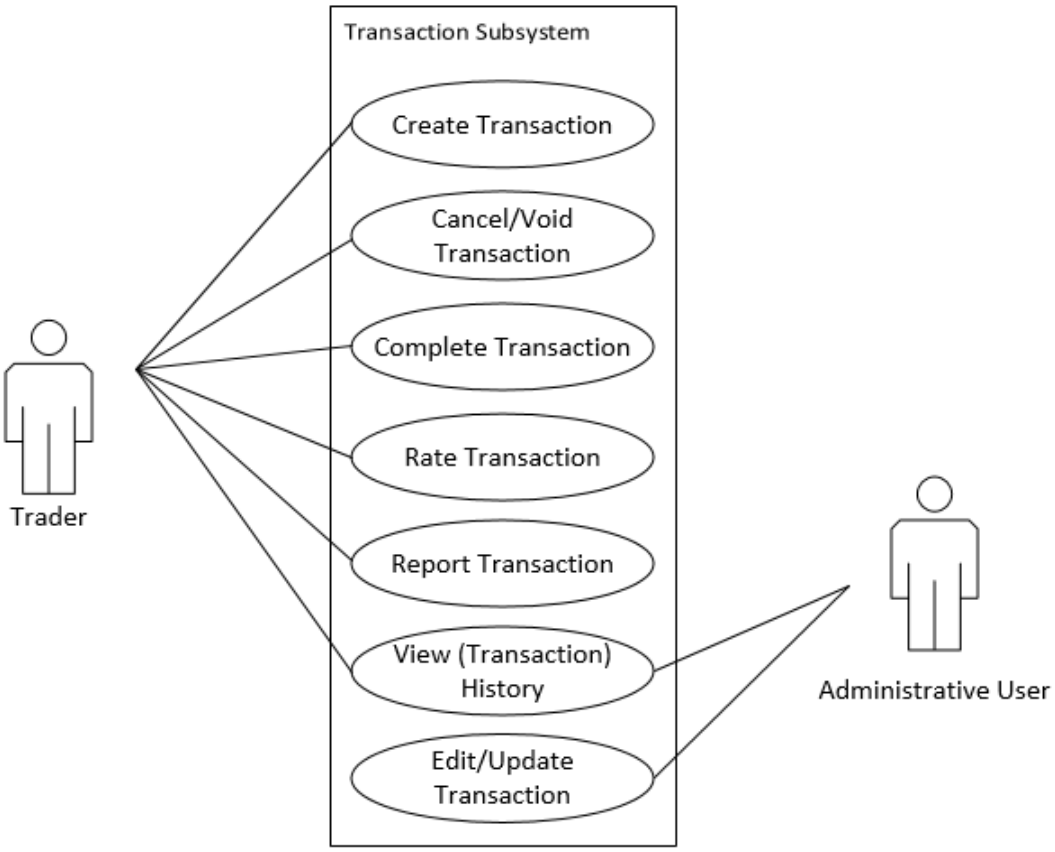
Create Barter (Transaction) Detailed Use Case Description:

Use case name:	*Create barter transaction.*	
Triggering event:	Trader discovers that a good or service on offer matches one of the trader's "wants" or the trader discovers a good or service another trader wants that matches something that the trader is offering.	
Brief description:	Trader selects the goods and/or services that the trader is offering and for which the trader is bartering from the appropriate profiles before pressing a "Barter" button (which is only enabled once transaction goods and services are selected).	
Actor(s):	Non-Business or Business Trader,a different Non-Business or Business Trader	
Stakeholder(s):	Non-Business or Business Trader (x2)	
Precondition(s):	Traders must have goods or services on offer. Users must have desired goods or services listed. One user must have goods and/or services on offer that match up with the other user's desired goods and/or services "Wants" and "Haves" listing information must be correct and up-to-date.	
Postcondition(s):	Barter transaction request must be successfully completed and sent to the second party.	
Flow of activities:	Actor (Trader)	System
	1. Trader selects the goods and/or services that will be involved in the transaction.	1.1 System sends the trader to a barter transaction creation form. 1.2 System prompts trader to enter additional information necessary for successful barter transaction (e.g. transaction location, transaction time)

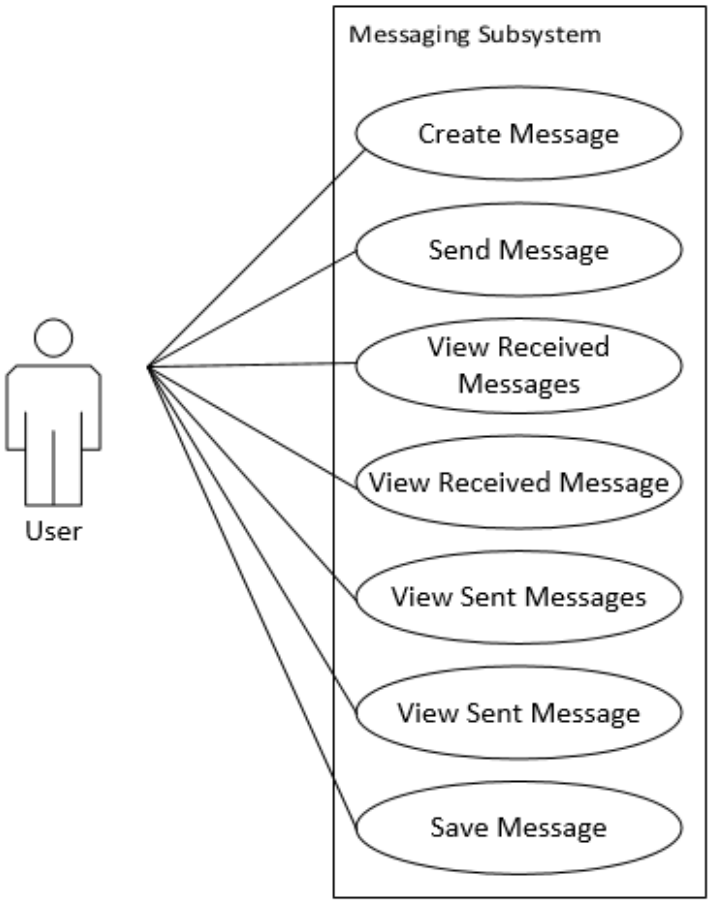
	2. Trader enters the rest of the barter transaction information and submits transaction request.	2.1 System validates trader input. 2.2 System sends transaction request to the second party involved in the barter transaction.
Exception condition(s):	1.1 Trader internet connection drops out before the user can access the transaction creation form. 2.1 Trader input is invalid. 2.2 Second party listings inaccurate or out-of-date. 2.2 Second party no longer uses barter application. 2.2 Second party has decided to give up all modern conveniences and go live in a yurt in the forest.	

Use Case Diagrams

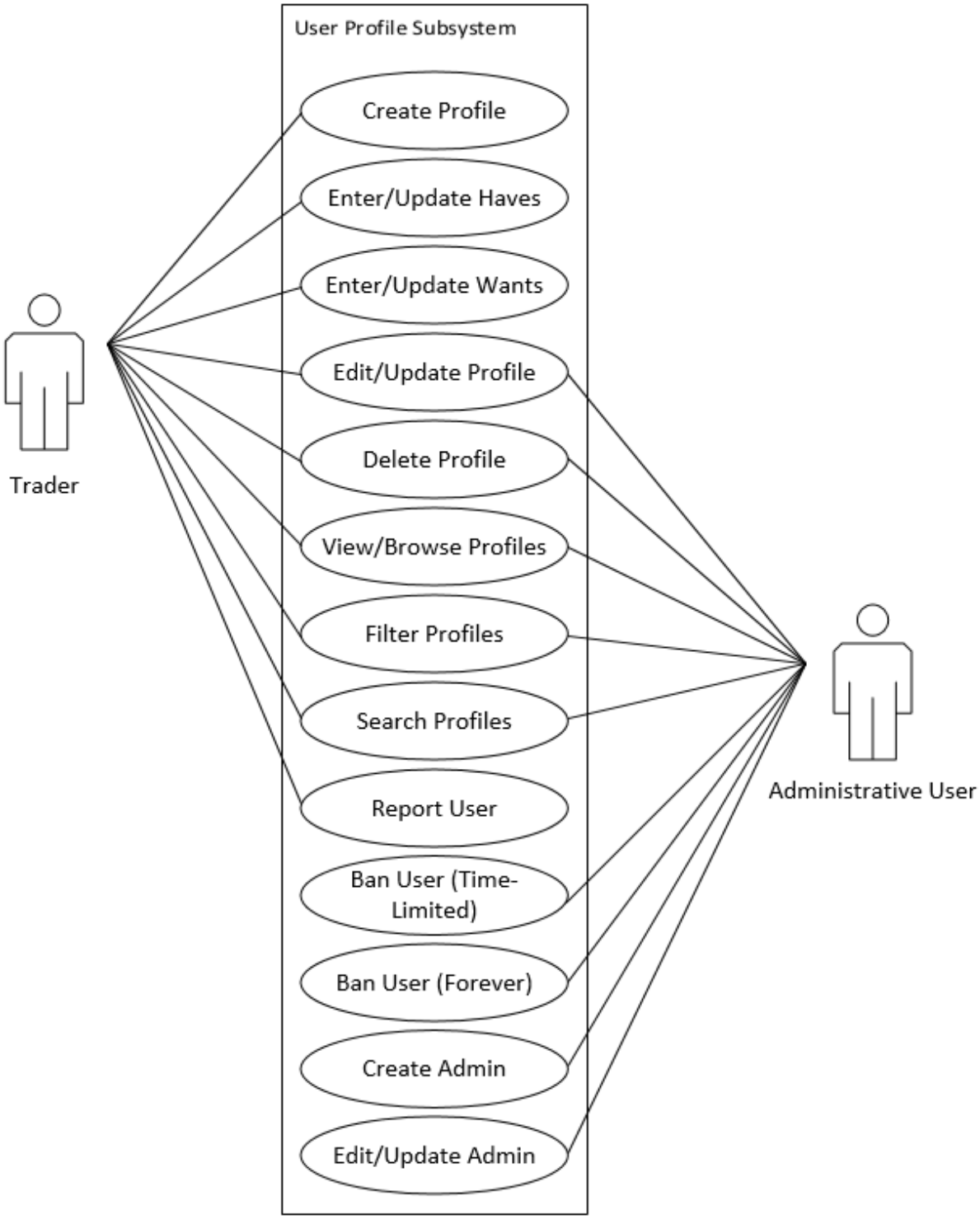
Transaction Use Case Diagram:



Messaging Use Case Diagram:

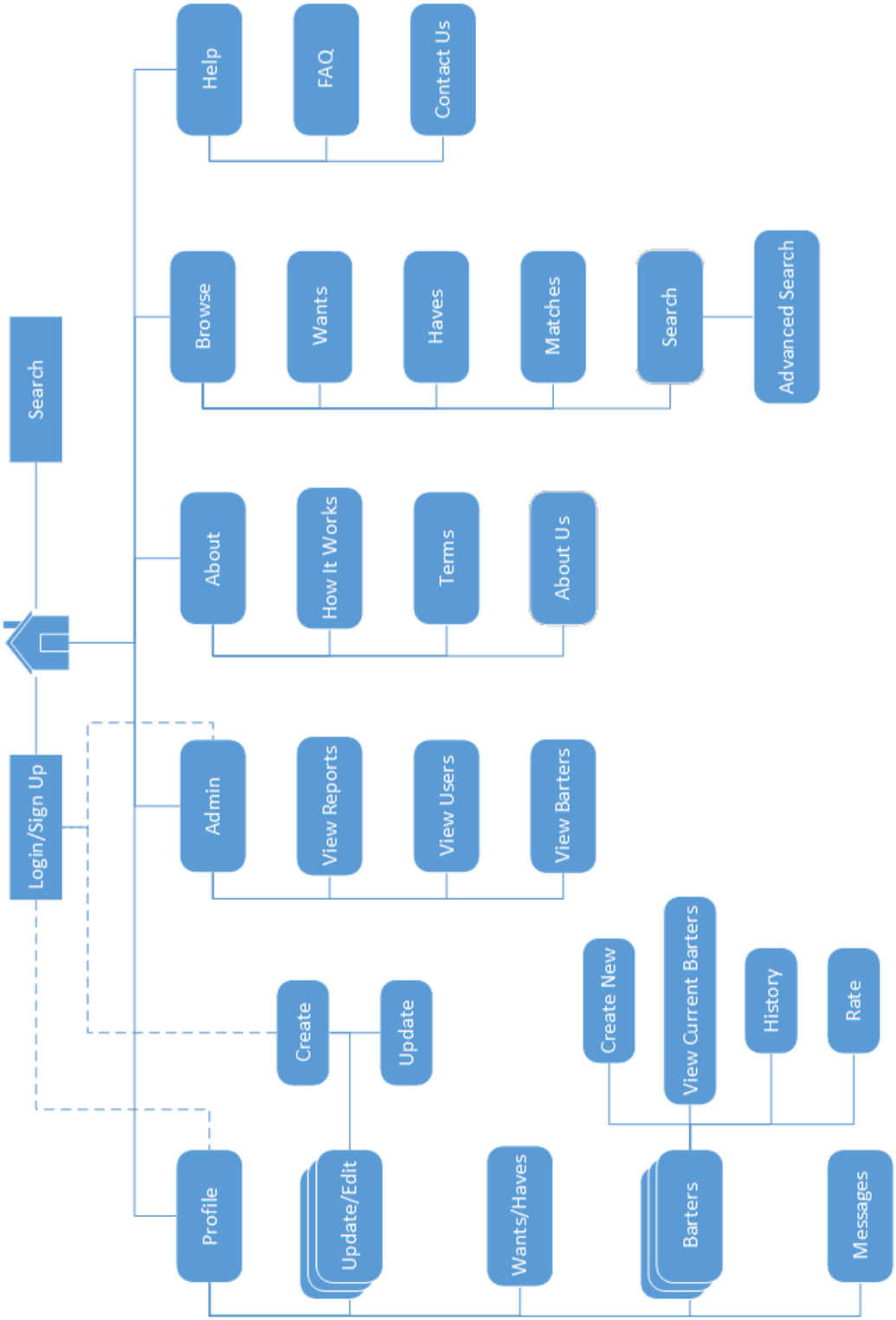


User Profile Use Case Diagram:



LAYOUT

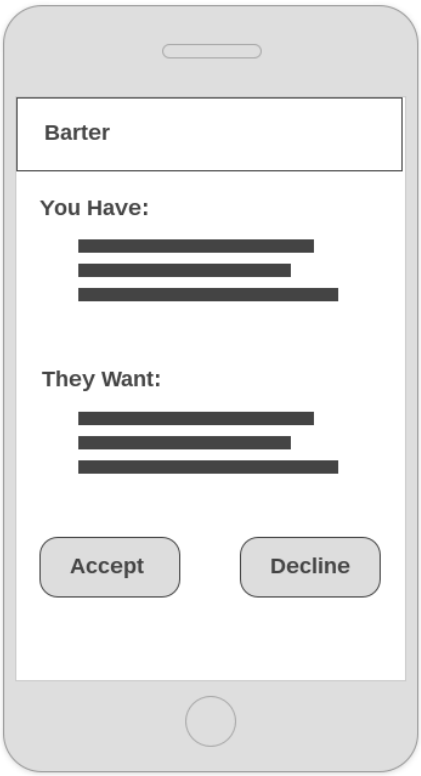
Site Map

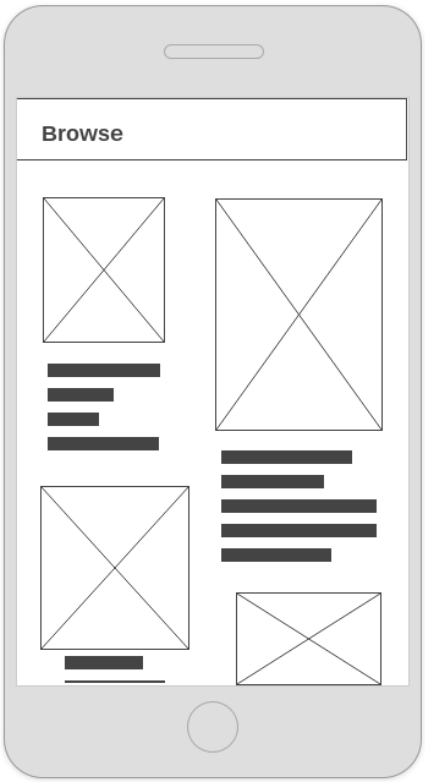


User Interface

Basic Profile View (Desktop):

Barter and Browse Pages (Mobile):





SYSTEM DESIGN

Architecture (Physical System)

The system will be designed as a responsive website that will specifically target mobile devices (smartphones, tablets). It will expand gracefully from a mobile layout to a tablet layout and, if necessary, a desktop layout.

Technologies

- *Programming Languages*:
  The web application will be created using standard web technologies such as HTML5 and CSS as well as C# and ASP.NET MVC.

- *IDE (Interactive Development Environment)*:
  The web application will be developed using Visual Studio Ultimate 2013. It is an IDE that we are familiar with and have been using to develop our projects for class assignments at LCC.

- *Version Control*:
  Version control will be done on Git via GitHub. Each developer will take a branch to develop and test a section of the code, and the team will review the branch before changes are merged into the master branch.

- *Unit Testing*:
  To keep the workflow consistent between production and testing, unit testing will be done with Visual Studio 2013. Controller methods will be tested for viability and stability to catch potential errors as soon as possible in the development process.

- *Deployment*:
  The web application will be deployed to and managed on Microsoft's Azure cloud service during development.

DATA DESIGN

Objects:

Admin : User
The Admin class is identical to, but separate from the User class. The Admin is separated in case there is a need to add any extra attributes or functionality to the class.

(Abstract) BasicMessage
A BasicMessage is the foundation that all other message classes are built on.
● *Sender:* User who created the message. ● *Recipient:* User to whom the message is sent to. ● *Subject:* Subject of the message. ● *Body:* Message body text. ● *DateSent:* Date and Time that the message was sent.

Feedback : Message
Feedback is a specialized message for rating and giving feedback on completed Trades.
● *Rating:* Numeric rating, 1-5. ● *Trade*: Trade that the feedback is for.

Item
An Item describes a product or service, possessed or desired.
● *Name:* The Item's name. ● *Description:* Item's description. ● *Tags:* List of ItemTags used to describe this Item. ● *Quality:* Numeric description (1-5) of item quality. ● *Images:* List of images of Item supplied.
● *<int>Compare(Item):* Compare this item to another to determine degree of similarity.

Message : BasicMessage
Messages can reply to other messages, and be replied to.
● *ParentMessage:* The message this message is responding to. ● *ChildMessage:* The Message that is responding to this one.

Product : Stock
A Product is a Stock described in terms of quantity or amount.
● *Quantity:* Integer that describes amount of product available for, or desired in, trade. If set to -1, quantity is unlimited. ● *Unit:* Units that quantity will be in.

Report : BasicMessage
A Report is created and submitted to document and report another User's violation of the law and/or TOS.
● *Target:* Url or offending User or Listing. ● *Admin:* Admin handling case. ● *Resolved:* whether reported issue has been resolved.

SearchAlert
Contains a search-string to periodically execute. When a new item is found, notify the Trader
● *SearchString:* String with search syntax to execute. ● *LastAlert:* Date and Time an alert was last sent. ● *Trader:* Who to send the alert to.
● *Execute():* Perform the saved search. If there are any new items (based on the posting date of the items, and the date and time of the last alert), send an email alert to the Trader.

Service : Stock
A Service is a Stock that described in terms of duration.
● *Duration:* Number of minutes available or desired in trade.

(Abstract)Stock
A Stock describes a quantity of an Item
● *Item:* The Item described. ● *DateUpdated:* Date Stock was created or updated.
● *<int>GetQuantity():* Get quantity of Item. ● *SetQuantity(int):* Set quantity of Item. ● *<string>GetUnit:* Get Item measurement unit.

SubTrader : Trader
A SubTrader is a Trader who has their account linked to a parent Trader. This is intended to allow businesses to permit their employees to do account activities (specific activities described by User Roles). A SubTrader synchronizes many of their fields with their parent Trader, but their trades can be tracked internally by their parent Trader.
● *ParentTrader:* The account to which the SubTrader belongs to. The SubTrader shares many of its values with its parent, though a specific SubTrader might not always have full access to all of the features of its parent Trader.

TempRole
The TempRole class describes a temporary authorization classification. The most typical use would be for temporary bans.
● *Role:* Role to be assigned. ● *Expiration:* Date and time that the Role will expire.
● *<bool> IsExpired():* Checks current date and time to determine if the Role has expired.

Trade
A Trade describes an accepted offer and a promise by both parties to complete the trade.
● *FinalOffer:* Accepted and agreed-upon offer. ● *DateFinished:* Date and Time Trade created. ● *SenderConfirmed:* Whether the Sender (on FinalOffer) has verified the Trade. ● *SenderCancelled:* Whether the Sender has opted to cancel the Trade. ● *ReceiverConfirmed:* Whether the Recipient has verified the Trade. ● *ReceiverCancelled:* Whether the Recipient has opted to cancel the Trade. ● *SenderFeedback:* Feedback from Sender. ● *ReceiverFeedback:* Feedback from Recipient.

TradeMessage : Message
TradeMessages have an offer (or counter-offer) attached.
● *Offer:* TradeOffer to display as part of the message.

TradeOffer
A TradeOffer represents a potential trade as offered by one Trader to another.
● *Sender:* Trader making the offer. ● *Recipient:* Trader getting the offer. ● *SenderItems:* List of Stocks offered by Sender. ● *ReceiverItems:* List of recipient's requested Stocks. ● *DateMade:* Date and Time offer was made. ● *CounterOffer:* Offer made by in response to this one. Usually by Recipient, but sometimes by Sender. ● *Open:* Whether the offer is open, and can be accepted.
● *<Trade>Accept():* Accept current offer, and create and store a Trade object. ● *Reject():* Reject or cancel offer, closing the offer and ending the trading dialog. (counter-offers not accepted)

Trader : User
The Trader class describes any trading member of the website. The class stores detailed data about the user.
● *Wants:* List of Stock objects describing collection of products and/or services that the Trader wants to trade for. ● *Haves:* List of Stock objects describing collection of products and/or services that the Trader can offer. ● *Trades:* List of Trades that the Trader has completed. ● *ProfileText:* HTML-formatted text displayed in the Trader's profile. ● *FavoriteUsers:* List of other Traders that this Trader has selected for fast access. ● *FavoriteSearches:* List of saved search strings (string which defines a search, whether simple or advanced, using special syntax) that the Trader has selected for frequent use. ● *SearchAlerts:* List of SearchAlert objects that the Trader has created to alert them whenever a new item appears in a defined search. ● *AllowRandomTrades:* Boolean value that determines whether the Trader has opted to allow other Traders to make offers to them of items not on their "Wants" list.
● *<decimal> GetRating():* Calculate aggregate rating based on the ratings on all Trades performed by the Trader.

(Abstract) User : IdentityUser
The User class is the base class for all account-using users of the application. User inherits most of its important functionality and properties from the IdentityUser class.
● *ProfilePicture:* Image url for displaying the user's profile picture and messaging avatar. ● *DateCreated:* DateTime object recording when the user created an account. ● *TempRoles:* List of TempRole objects that enumerates any temporary Roles. ● *Messages:* List of BasicMessages written, sent, or received by the User.

Relationships (UML)