Design Documentation for Grand Exchange

Ami Suzuki, Eunice Lin, George Du, Jeffrey Sun

Overview

Motivation

Description

The Grand Exchange is an item-centric centralized online marketplace for the MIT community. Users can post auction-style sell or buy offers on the application for a particular item and will be notified by the app when a match is made between a buyer and a seller. When a match is made, the offers will be taken off the marketplace and the users can meet in person to do the actual transaction.

Key purposes

1. Provide an item-centric rather than offer-centric online marketplace
   a. Consolidates all offers for the same item so that users do not have to scroll through lots of posting of the same item
   b. Provides the fairest price determined by the market
2. Centralization of buy and sell
   a. Consolidates all buy and sell offers in the same place, rather than dispersed over emails forums, Facebook groups, etc.
3. Takes the burden off of buyers and sellers by facilitating the time burden of transactions
   a. Does the match-making for the users so that buyers and sellers do not need to negotiate the offers or respond to specific offers themselves, which saves time and effort

Deficiencies of existing solutions

- eBay and Craigslist
  - Offer-centric; users have to scroll through long lists of postings searching for an item they want.
    - We want to create an item-centric model where user can just name the item and price they want without having to search through a bunch of different postings.
- APO Book Exchange
  - Only for textbooks and clickers.
    - We want to expand our application to accommodate items in addition to textbooks and clickers.
  - Not online; people have to bring books to the APO Exchange and if they are not sold they will have to bring them back.

- - - Users of our application will only have to meet in person to do the transaction if they were guaranteed a match. This will save people the time and effort of bringing their items to sell in case no one buys it.
  - Facebook groups and email forums
    - Not centralized and difficult to keep track of; users will have to post in the group individually and their post are not searchable so it might get buried by other posts. Users also have to periodically check the site for new offers and privately message the other person if they want to arrange a transaction. Often times, people would message the seller, and the item would have already been claimed.
      - We want to create a centralized platform so that the user only has to name the price and item and they will be notified by the application when a match as been made. Therefore, the user does not have to worry about their post being buried by other postings and do not have to check the website frequently.

Context Diagram

Design Model

Concepts
- Item -- The object being bought/sold and its buy and sell offers
  Motivated by:
  - Purpose 1: Each Item can have multiple offers. Users will look at the many offers of an item, rather than the single item of an offer, so we are providing an item-centric rather than offer-centric marketplace.
  - Purpose 2: Each Item has both buy and sell offers, which centralizes buy and sell
- Offer -- A user's intent to buy or sell at a particular price. Items may have multiple buy and sell Offers
  Motivated by:
  - Purpose 1: Offers exist under Items and Items may have multiple buy and sell offers, making the marketplace item-centric

- ○ Purpose 2: There are both buy and sell offers for each Item so buying and selling is centralized
- ○ Purpose 3: Offers allow the system to make a match so users can easily buy or sell
- **Transaction** -- Has exactly one buy offer and one sell offer that have been matched. The system matches the highest price buy offer with the lowest price sell offer if the sell price is less than the buy price

  Motivated by:
  - ○ Purpose 2: The system creates a Transaction when there is a match from the collection of buy and sell offers, so buy and sell are centralized
  - ○ Purpose 3: Transactions are created by the system, so buyers and sellers do not have the burden of finding matches themselves.
- **User** -- MIT community members (with MIT email addresses) that can create buy/sell offers

  Motivated by:
  - ○ Purpose 1: Users can create offers under Items so each Item may have multiple offers, making the marketplace item-centric
  - ○ Purpose 2: Users are able to create both buy and sell offers, which centralizes buy and sell

Data Model
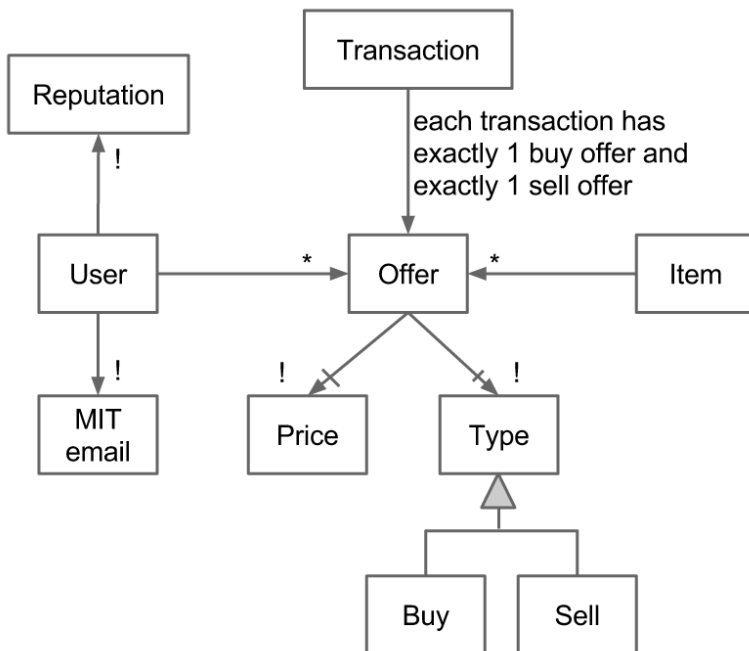
Users of the application must have a valid MIT email address. Users can post offers on the application. Each offer has a fixed price and a fixed type, either buy or sell. An item can have zero or more offers. A transaction is created when a match is made between a buy offer and a

sell offer. Each transaction consists of exactly one buy offer and one sell offer, which must be for the same item. The price of the transaction is the minimum of the two offer prices, and the sell offer price must be greater than or equal to the buy offer price. Each user has a reputation, which is a rating based on other users' feedback on them. Users can rate each other by using the up vote and down vote features.

Behavior

Security Concerns

Our basic security goal is that users should be limited to performing only allowed actions and viewing allowed data. The allowed actions are: buy/sell offers, rate users, and add items. The allowed data is: the available items, offers on those items, information about the user's own offers, and other users' reputation. This will be accomplished by using a REST API which acts as an RPC interface limiting the actions a browser can perform, even using custom-built requests. The standard attacks such as those listed in the OWASP top ten are mostly taken care of by our framework AngularJS (injection, XSS, and CSRF in particular.)

We model the attacker as an entity which can send arbitrary requests w/ arbitrary cookies such as those created in Postman. Our API should be designed so that any malicious requests (those not normally allowed for a user) will fail.

User workflow diagram
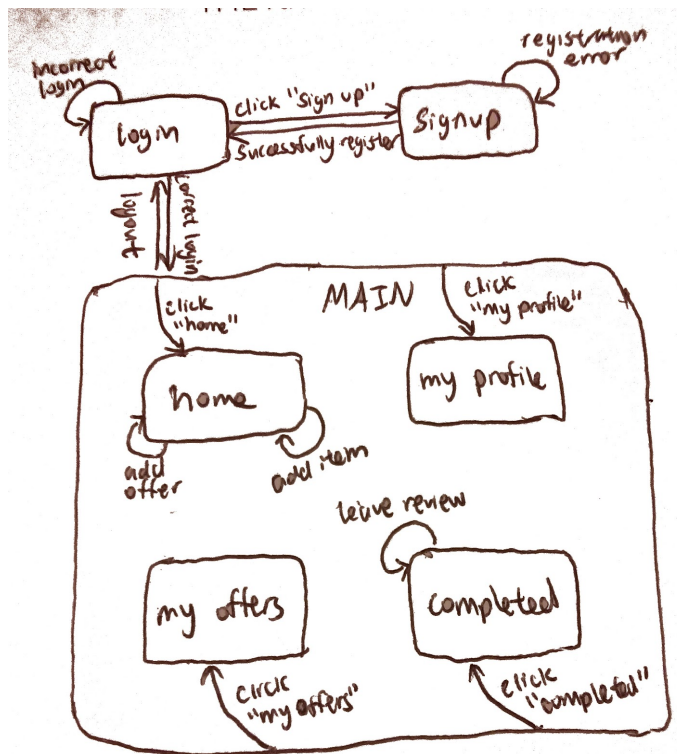
Page Wireframes

Login Page

Grand Exchange

Username

Password

Login   Signup

Signup Page

Signup

Username

Password

Retype Password

Register

Profile Page

My Profile	**Your Name Here**
Home	Reputation: x
My Offers	
Completed	Reviews
Log Out	

Home Page

My Profile	
Home	
My Offers	
Completed	
Log Out	

Sell: xx.xx
Image
Buy: xx.xx

Sell: xx.xx
Image
Buy: xx.xx

Sell: xx.xx
Image
Buy: xx.xx

Sell: xx.xx
Image
Buy: xx.xx

Sell: xx.xx
Image
Buy: xx.xx

Sell: xx.xx
Image
Buy: xx.xx

+

Adding an Item

My Profile
Home
My Offers
Completed
Log Out

Sell: xx.xx
Image
Buy: xx.xx

Sell: xx.xx
Image
Buy: xx.xx

Sell: xx.xx
Image
Buy: xx.xx

Sell: xx.xx
Image
Buy: xx.xx

Sell: xx.xx
Image
Buy: xx.xx

Sell: xx.xx
Image
Buy: xx.xx

New Item
Name
Description

Upload Photo +

Item Detail

My Profile
Home
My Offers
Completed
Log Out

Sell: xx.xx
Sell: xx.xx
Sell: xx.xx
Sell: xx.xx

Image

Buy: xx.xx
Buy: xx.xx
Buy: xx.xx
Buy: xx.xx
Buy: xx.xx

Description:
yada yada yada yada yada yada
yada yada yada yada

I'm willing to

Buy

Sell

for

xx.xx

My Offers

My Profile
Home
My Offers
Completed
Log Out

Price	B/S	Item
xx.xx	B	yada1
xx.xx	S	yada2
xx.xx	B	yada3
xx.xx	S	yada4
xx.xx	B	yada5
xx.xx	S	yada6

Show **25** 50 99 on page

Completed Transactions

My Profile
Home
My Offers
Completed
Log Out

Bought x from y for $z
Was the transaction completed in person? ○
Write a review...
Submit

Sold x to y for $z
Was the transaction completed in person? ○
Submit

Design Challenges

Preset items vs. dynamic items

Since our application is item-centric rather than offer-centric, one of the limitations is that we can only offer homogenous goods, since users cannot distinguish between different types or quality of the same item. Therefore, we thought about having pre-set items that our site could offer. This prevents the users from adding inappropriate or unreasonable items, such as "my old laptop" or "a dinosaur." The other option is to allow users to add new items when they want to buy or sell something that is not already offered on the site. This allows our site to be more flexible in accommodating everything that people want to buy or sell. We chose to allow users to add any item to the site and regulate inappropriate or fake items with reviews or the reputation system, such that users who make inappropriate or fake offers will receive low ratings and negative comments from other users and will eventually be banned from the site.

Transaction objects

Transactions are matched automatically when buy and sell offers intersect. In an ideal world, this would correspond to a transaction happening. Realistically, however, we have to take into account things such as fake transactions, transaction spamming, and transactions that fail to occur due to low item quality, etc. We choose to have transaction objects which can reference the offers that spawned them, thus providing a way to recreate those offers if the transaction fails. Although this feature may not be present in our MVP, our data model anticipates later challenges related to transaction verification and provides a way to mitigate them.

User Verification

Since we need users to be able to trust one another, we decided to make our app open only to MIT community members. We could either have users with MIT certificates, or users with MIT email addresses. Since there are a variety of MIT email address types (such as @mit.edu and @csail.mit.edu), using MIT email addresses would require us to use libraries that would do email verification. On the other hand, using MIT certificates has its own challenges. Since scripts.mit.edu does not work with node.js, in order to use certificates, we would need to generate and sign a fake certificate. However, this would cause the browser to question every time whether our site should be trusted. We have decided to use MIT email addresses rather than deal with the difficulties that may arise from trying to use certificates.

User Workflows

One important issue is how to structure the pages in the front-end of the application. Specifically, there are multiple ways to allow users to view an item's details and put in offers. One solution is to display item details and offer inputs on the same page that shows all the items, and have the item expand on click. While this approach speeds up the offer process and motivates the user to spend more time buying and selling, it doesn't give individual items an identity. On the other hand, putting the page detail and offer form on its own page allows individual items to be linked to from a user's offer list. This was the chosen approach, because it was decided that granularity at the item level is worth a sacrifice in speed.