# LottoDeal

<u>Team Leader:</u> Prateek Swain (prateeks@princeton.edu) <u>Team Members:</u>

> Lucas Irvine (<u>lirvine@princeton.edu</u>) Steven Takeshita (<u>st14@princeton.edu</u>) Antony Toron (a<u>toron@princeton.edu</u>) Dominic Whyte (<u>dwhyte@princeton.edu</u>)

# OVERVIEW

## **Elevator Pitch**

A revolutionary, new buy-sell platform where we match sellers looking to raffle off their item with hundreds of interested micro-bidders. No more indefinite waiting periods, no more hassle. An opportunity to buy everything you desire for a fraction of its price, all guilt free!

## **Investor Pitch**

LottoDeal connects buyers and sellers in a unique marketplace model. Instead of a traditional e-commerce model, the seller puts down their item as a lottery which anyone can participate in for a fraction of the original product price! Winning a sleek new \$250 Apple Watch is as simple as putting only \$2 down on our webapp for a realistic chance to be the next proud owner of the Apple Watch.

We are completely transparent about our lotteries by clearly displaying the chances of winning each item being sold. Buyers and sellers also receive complete security since items are not given away until the original target price of the seller is reached. If a goal is not reached, all users who bid on the item get their money back. Our app acts as the intermediary for payments, listing items, security, and all communication between the buyer and the seller.

We utilize the psychology behind the popular state-run lotteries and apply it to a buy-sell platform model to compete with the traditional e-commerce incumbents. There are a plethora of products which have low turnover rates and are listed for sale for long periods of time. This especially applies to luxury goods since their target customer base is much smaller. By auctioning off such items in a lottery, we include buyers who wish to bid smaller sums of money for a chance to acquire expensive items. LottoDeal thus connects all items with a much larger consumer audience and helps people acquire products they desire without breaking their bank. You don't need to eat just ramen noodles for three months to afford that vacation in Jamaica. Come put a few dollars on the platform, and with a little luck you'll be sipping pina coladas in the tropical beach by next week.

## **Requirements and Target Audiences**

### Problem

Costly products are often very desirable, but can be too expensive to justify purchasing. For example, for college students, these items can range anywhere from bicycles to dinners at upscale restaurants to even a non required reading for a class (speaking from our personal experiences).

Most products thus have a large set of potential buyers who do not wish to spend the full retail cost of it, but still would like to own the product. Because of this, a large untapped audience exists that is not being reached out to for many products. The only existing solutions for exchange of these items include micro-bidding sites and traditional marketplace models. More details about both of these alternative platforms and their associated problems are detailed below:

<u>Micro-bidding sites</u>: Platforms that have timed auctions and allow you to buy items by bidding small amounts on them (eg. quibids.com)

*Problem*: The auctions are time-sensitive and non-transparent. These sites often act as retailers who try to auction off their items for a profit by raffling them off. Unfortunately, users do not know the number of bidders, their chances of winning, and other pieces of information required to establish trust between the model and users. In extreme cases these sites scam their users.

<u>Buy-Sell Platforms</u>: Platforms, such as Craigslist or eBay, that connect buyers and sellers in a traditional online marketplace model.

*Problem*: When dealing specifically with high value items (luxury goods, niche products, most products with low turnover rates etc.), long waiting periods for sellers and very little interest from buyers lead to a poorly functioning marketplace for these items. Sellers often do not find a buyer for their product, and therefore try to either find a new marketplace or sell their product at a reduced price.

#### Intended Users

Our intended users include anyone with a desire to buy or sell an item. Ideally, this includes all geographic locations and demographic groups. However, because of its potentially broad user base, we have identified the most optimum plan of implementation as the following:

Phase 1: Princeton University Community

Phase 2: Other College Campuses

Phase 3: North America

Phase 4: Rest of the world

# Functionality

## Use Case scenarios:

## Seller:

- 1. The seller logs onto the web application using their facebook (other login methods can be added later)
- 2. The seller uploads a picture of the product they wish to sell, as well as a description, a target price, and a date by which the target price must be met. The seller should also list the estimated time and price of shipping for the product based on geographic location. Other potential options for the seller include the max bid size and the max number of bids each buyer can make.
- 3. After the seller inputs this information, the product becomes visible to buyers
- 4. There are now two possible outcomes:
  - a. If the target price is met, the raffle will take place and the seller will receive a notification that their target price has been met and a raffle for their item has occurred. The app will notify the seller and the winner of the raffle, and allow for direct communication between them so that shipping information and any concerns (e.g. the product did not arrive) can be discussed.
  - b. If the target price is not met, all money will be refunded to the bidders. The product will no longer be listed on the site, unless the seller tries to resell the item by going back to step 2.

## <u>Buyer:</u>

- 1. The buyer logs onto the web application using their facebook (other login methods can be added later)
- 2. The buyer then sees a list of categories of products. The buyer can search through these categories, as well as by price of bid, chance of winning, and product name.
- 3. The buyer will then see a list of pictures of items that fit the given criteria. Above the picture will be the product's name, and below the picture will be the time left to bid, and price of each bid and associated chance of winning the product. By clicking a "bid button," the buyer pays the listed amount and receives a notification that their bid has been accepted or rejected.
- 4. By clicking on the picture of a product, the buyer will be able to see more detailed information including a description of the product, estimated shipping times, the condition of the product (for used items), etc.
- 5. After the target goal is reached or the product deadline is reached, either a winner is chosen or all money is refunded to the bidders (as described in the seller's use case).
- 6. If the buyer has won a given product, they will receive a notification that they won.

#### General:

Our page will also have the following content:

- 1. An about page describing all 5 of us and a way to contact the company
- 2. A recent winners page listing the products that were recently raffled off as well as the winners of those products

#### Design

The app will be focused on having a clear distinction of the three-tier system separation, and incorporation of the MEAN stack.

#### First-tier: Front-end

The first layer will be the client side or front-facing web application. The goal is to incorporate AngularJS onto the web application, allowing for easily updated DOM elements, and dynamically changing data displayed to the users. AngularJS's strengths lie in its ability to allow instant updates of the DOM without refreshing the page, allowing for a smooth UI and minimal effort on the user side in terms of viewing the information presented (as well as easy integration with server-side technologies, as will be described below). A simple use case of this can be seen where users post a new item to be sold, the user will instantly see this new post regardless of having to refresh the page. They will also be able to see real-time purchases of raffle tickets on their items - a useful tool for any user on our application.

Other potential libraries/frameworks in this layer will be the following technologies: Bootstrap, Jquery, Font libraries. Though this is a web-based application, another goal will be to make this mobile-friendly to allow people to view the site on any device (which Bootstrap can help do). Jquery is another library that is very useful to accomplish similar things that AngularJS does. Jquery is light, and it allows for quick coding in terms of selecting elements in the DOM, attaching listeners to certain things on your page, and also Ajax integration. Using this on top of AngularJS is useful, because many things that AngularJS does are abstracted away from the user, making it difficult to debug certain important features. This is when JQuery can become useful to provide more control over the DOM and particular features due to its nature of being a library as opposed to a framework. Both Jquery and AngularJS have widespread support, as well, making the two together a good combination for making sure that all features will be working smoothly. The last category - Font libraries - is a blanket term to describe all of the different fonts and styling we aim to find to create an appealing UI and graphical interface for users to work with on the website.

#### Second-tier: Server

The next abstracted layer is the middle-tier, server-side. We plan to program our server in NodeJS and ExpressJS/Express, because these technologies provide a simple and easy way

to set up a web-service as opposed to building something in a more hefty software like Java. There is also very handy integration of NodeJS into many modern databases (hence the incorporation of MongoDB into MEAN). By using NodeJS, this frees up more time to focus on front-end development, because of the speed that a NodeJS application can be created with. The idea with this layer is to create a very modular way to access information from the database to send back to the frontend. Example use: a user is trying to put up a new item to sell, so they create the request on the front-end, which then conveniently in the javascript (due to to the AngularJS integration with Ajax) sends an HTTP request to this running server that has a modular method for accepting certain types of data, without having to talk to the database directly.

Another important part of this layer will be Stripe. Stripe is an api that allows for easier payment and transaction processing for apps. It has an extensive API reference and existing infrastructure to be used in NodeJS. This is another reason why NodeJS is useful in this layer: there are multiple libraries and apis supported in NodeJS, which allows for quick integration of things like Stripe.

#### Third-tier: Database

The final layer is the database, which we plan using MongoDB for. The reasoning behind this is that NodeJS is well adapted to working with Mongo servers, allowing for quicker development and creation of the product. Additionally, if we think about the data that will be used for this application, it can be seen that many of the items we wish to save may not be structured (a consequence of using a REST api in Node and JSON objects in that layer). Some of this type of data includes posts: one person raffling off an item might want to have a different configuration of settings for that item than another user, which might become too complicated working with MySQL or other non-relational databases. During the creation of this project and in the long term, however, we will see what type of database is best for the type of data that is decided upon in the final product (which could result in use of a MySQL-type database).

#### Expected Roles:

Web-development:

HTML and CSS focus (AngularJS, Bootstrap, other styling libraries)

Javascript Focus (AngularJS, JQuery, Ajax)

Combination of the above two

Design (planning of UI/UX, and creation of logos and branding - Photoshop, Illustrator) Server-side:

Setup of REST Api to pick up requests from the front-end (NodeJS, Express)

Setup of Stripe API to process the transactions (NodeJS, Express)

Hosting of the server on either Princeton servers, or AWS, or some other hosting service Database:

Concrete contract development on the type of data that will be processed and stored, and consequently the correct database to use.

NodeJS queries and pushes to the database (in the case that a non-relational database is used as expected, like Mongo).

## Timeline

## March 3, 2017

Survey classmates to measure product demand and interest.

## March 7, 2012

Meeting with Professor Kernighan, obtain project approval.

## March 24, 2017

Develop a structured contract between the different tiers of the application (API, data processed in database, etc.) and assign roles to each member of the team, based on interest and expertise.

## March 31, 2017

Have a working prototype of the payment system (likely using Stripe), to eliminate the uncertainties associated with implementing delayed transactions. Decide on database layout and begin implementation of MVP API functions. On the front end, have a full drawn-out layout of the MVP design and begin working on implementation.

#### April 7, 2017

Test working server, which should implement the functions necessary for adding an item for sale, bidding on an item, and performing the lottery (including working database). Front end should be functional on large screens, work now on mobile optimization.

#### April 14, 2017

Project Prototype. Minimal Viable Product prototype test - By the 14th, have an initial working version which can handle the baseline functionality, ie., add an item for sale, bid on an item, and perform the lottery (including thorough documentation). Frontend must allow for this functionality. Determine bugs, areas for improvement in UI and decide what areas we need to focus on most before the April 24th launch.

## April 18, 2017

Heavily polish the MVP by the 18th, and begin adding non-baseline features, such as displaying lottery odds to users and newsfeed, subject to pivot.

#### April 24th

Launch. Release the latest version for student use, by marketing on campus and posting to listservers. Goal will be to auction off 5 items to students in the first 10 days. In the next week, work on adding any key supplementary features and on bug fixes based on student feedback.

#### April 28, 2017

Alpha Test. Finalize adding supplementary features, and do stress as well as corner testing.

### May 4-5, 2017

Beta test. No more feature adding - spend these days rigorously testing the existing product.

# May 8-10, 2017

Demo day.

# May 14, 2017 (midnight)

Final submission for Dean's Date.

#### **Risks and Outcomes**

- 1. Buyer/Seller Security. One key area of potential risk is performing transactions after the item lottery price has been reached, ie. several days after a user inputs their payment information for the sale. We have experience using the Stripe API for immediate transactions, however there is a lot more uncertainty in storing the payment tokens for use at a later date. For example, there are edge cases such as: What if a user's payment information becomes invalid in the days before the transaction goes through? How would the selling price of an item be affected by a user providing payment information has been invalidated? If instead we perform the transactions immediately, how can we reliable return the funds to users if the lottery doesn't reach its goal? (and not be subject to the 2.9% fee that Stripe imposes?).
- 2. Customer Satisfaction. We need to develop a system for transferring purchased products from the seller to the buyer, in the case of a successful lottery. This includes risks such as: What if the item being sold was unjustly represented in the description (eg. wear and tear)?
- 3. Lack of clarity on positioning items as lotteries: We don't have any data on what the best way to break down the price of an item into bids for users. Should bids solely depend on no. of bids being circulated (ie should a bid price be broken down as a fixed multiple of the total price to keep chances of winning constant) or should there be a range of values which would be optimum for get more users to bid (we assume it would range between \$0.1 to \$10). Not knowing how to break down an item into a lottery could lead to items not being auctioned off and thus lack of any benefits.
- 4. Legal issues: We don't expect this to be an issue, since we will be launching solely at Princeton and dealing with items of small economic value, however in the long term we will have to consider the legal ramifications of selling items in this style, and the laws that apply to a business in this new business. Since this is a relatively uncharted area, a scaling business based on this model would likely face similar challenges to Airbnb and Uber/Lyft, pioneers in their respective industries.

5. Learning New Programming Languages: To learn the most from this experience, we will task each member with a part of the project that they have never worked with before. Therefore, we will all be able to learn about new technologies and develop our skills quickly. However, the problem is that we will definitely run into bugs all the time as we try to learn. This should not be a big problem though as Antony Toron has much more experience than us in Web Technologies and will be able to walk us through when we get stuck. Also, our tight timeline will allow for plenty of time to pivot in case we are not finding it feasible to reach our initial goals.