Jazzy Ostriches

Alessandro Cartegni, Brian Leung, Dasha Shifrina, Joyce Wu

Softdev pd. 7

# Lost in New York

A lost and found web application that allows users to report and claim lost items in New York City.

---

## Functionality

- Google Maps API : adds markers on map of New York City for location of found items
  - User may zoom in on certain areas to see more markers
  - User can input the location and API will identify the coordinates of the address
- Lost and Found API : preloads a list of lost and found items from lostandfound.com
- Mailjet API : allows for email communication between users
  - If person wants to claim found item, he/she can contact the owner of the post, who will then ask a security question (up to users' discreton)
  - Messenger will maintain communication between users
- Accounts created so users can add, delete, and keep track of posts
- Listing of lost and found items
  - Can be categorized by location, type of item, date object was found/lost
  - User can click on individual postings to get more information and contact person
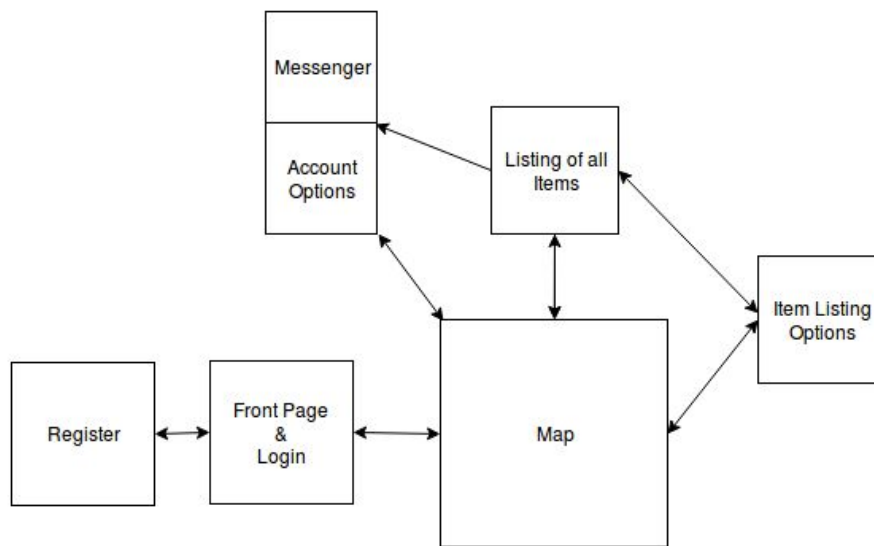
---

**Team Roles**

- Alessandro (PM) : help out with front end and where needed
- Joyce : front-end integration
  - Will use Google Maps API for integration with maps
  - Bootstrap and JavaScript
- Dasha : item postings
  - User profile system
  - Main flask app
  - SQLite item database
- Brian : account system and messenger feature
  - Will use email API to help maintain correspondences between users
  - Flask app for accounts

---

**Database Schemas**

- Accounts: ID | username | e-mail | hashed password | list of found item IDs
- Items: ID | item | description | location | account | status
  - Every time a post is created, the item is assigned a unique ID in numerical order
  - Account identifies the person who created the post
  - Status will be a true/false boolean that identifies if item is a found or lost item
    - True = found
    - False = lost

---

**Site Map**

## Component Map



**app.py**
-root route directs user to login page
- map route lets user interact with map
- chat route allows users to chat using a messenger box

**map.py**
uses google maps api to allow users to navigate through map and search for lost items.

**Google Maps API**

**account.py**
manages user registration, changing account info, editing lost and found items, and maintaining connections to other users in the messenger system

**users.db**

**items.db**

**chat.py**
uses sendbird api to allow users to interact with one another to link lost& found items. connects users using username.

**SendBird API**

**login.html**
prompts user to either login using username and password, or create an account using the same fields along with an email

**account.html**
allows user to edit and add/remove lost and found items.

**chat.html**
shows a very similar interface to messenger, except the only people user can talk to are the ones they've connected to either through a lost or a found item

**index.html**
displays the map with lost and found items (using javascript)