SW Engineering CSC648/848
Section 01
Fall 2018

GatorTrade

Team 04 | local
Abigail Chin | chinabigail08@gmail.com
Alyssa Malunao
Dhruv Shah
Jarek Rettinghouse
Jed Ahmadia
Nikita Bajracharya
Robert Quiñones

Milestone 2

Submission Date	History
October 18, 2018	Submitted for review

1. Data Definitions V2

1. **Unregistered Users**

   A user who has no account, or has yet to log in to account. This type of user shall still be able to view content on the site, but cannot make a new posting or message a seller. This user shall be able to create an account if he/she so chooses. Access to view posts but not edit.

2. **Registered Users**

   A user who has an account on the site and is logged in. They shall be able to post new items for sale, message others, and can be either a buyer or a seller. Ability to access to view all posts but unable to edit posts created by other registered users. Can create posts that will be sent for approval, and be able to remove their own posts.

3. **Admin**

   Special credentials gives the admin access to the Admin Console. The Admin Console allows the admin to approve pending ads and upgrade other users to admins as well. Admin shall have access to view posts. Admin shall have access to remove and approve posts from registered users.

4. **Items**

   At item posted by a Registered User for sale. Items include a title, description, price, category, related images (zero or more), and a preferred meeting place.

5. **Messages**

   A form of communication between Registered Users relating an item for sale. Messages are sent from a buyer to a seller, with contact information attached so that the seller will be able to contact the buyer based on the given contact information.

6. **Images**

   Images can either be a user's profile picture, or images relating to the item being sold. Images shall be uploaded using the service Cloudinary, be no bigger than 1MB in size, and will all be converted into PNG images.

7. **Category**

   Subcategories for items posted on the site (ie. books, electronics, home goods, etc.)

8. **Meeting Place**

   Meeting places are approved locations on campus that a seller can meet a buyer to exchange goods.

9. **User Registration Records**

   Data collected from the user at registration time. This includes a valid SFSU email address, first name, last name, and optional profile picture. Once the user is added to the database, they also are given a unique user ID.

2. Functional Requirements V2

Priority 1:

Unregistered Users

    1. Unregistered Users shall be able to browse site and view all listings without creating an account.

    3. Unregistered Users shall be able to view item details.

    4. Unregistered users shall be able contact seller through the site.

    5. Unregistered Users shall be prompted to create new account for buying and selling purposes.

Registered Users

    6. Registered Users shall be able to log in and log out of the system.

    7. Registered Users shall be able to view dashboard.

        Dashboard: profile page for users. Contains messages from buyer/seller for interested items.

        Contains user settings, watchlist, and list of items posted.

    8. Registered Users shall be able to post items/services for sale.

    9. Registered Users shall be able to remove items/services posted.

    10. Registered Users shall be able to communicate with buyer/seller.

        10.1 Registered Users shall be able to message the seller for an item interested.

        10.2 Registered Users shall be able to view messages from interested buyers.

Admin

    13. Admin shall be able to login and logout.

    14. Admin shall be able to grant and deny access to users.

    15. Admin shall be able to manage contents (CRUD).

    16. Admin shall be able to create and delete account.

    17. Admin shall be able to manage items and postings.

        17.1 Admin shall be required to approve all items for sale before they go live. Items shall be posted until admin's approval.

        17.2. Admin shall be able to remove inappropriate or unfit items and message the user.

Items

    18. Item shall have product image in product detail page.

    19. Only approved items shall be posted.

Priority 2:

Unregistered Users

2. Unregistered Users shall be able to view multiple items listings on a single page.

Registered Users

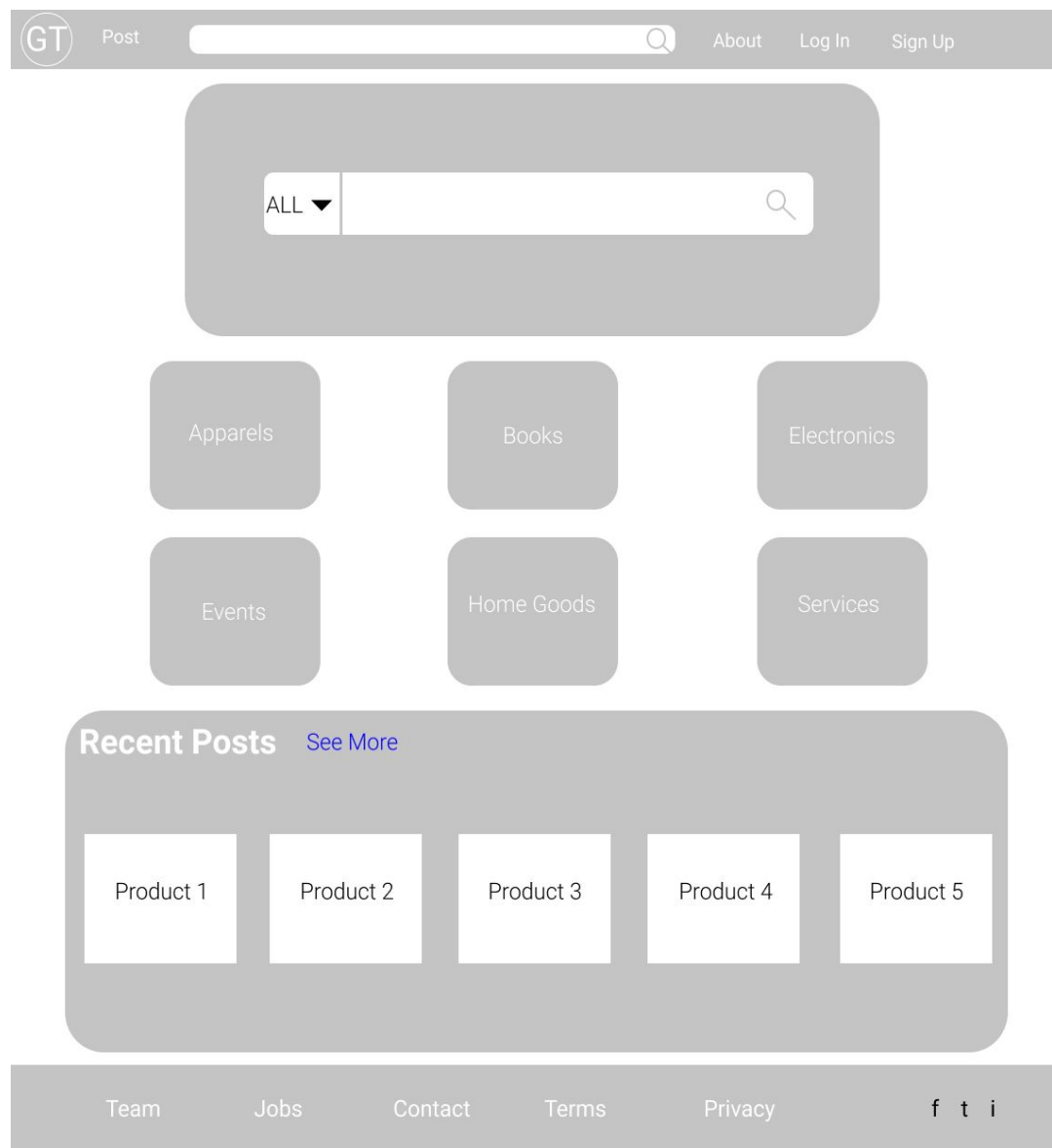11. Registered Users shall be able to mark items they have sold as 'sold'.

Priority 3:

Registered Users

12. Registered Users shall be able to leave a feedback for website.

3. UI Mockups and Storyboards

1. **Home Page(/home)**

   This is the Home Page of our website. It consists of search bar, drop down category menu for search, categories, recent post and footer. General/Unregistered users can easily browse sites and listings. The bottom panel contains information about the team, jobs, contacts, terms privacy, and social media pages.

GT   Post   About   Log In   Sign Up

ALL ▼

Apparels   Books   Electronics

Events   Home Goods   Services

Recent Posts   See More

Product 1   Product 2   Product 3   Product 4   Product 5

Team   Jobs   Contact   Terms   Privacy   f t i

2. Listing Page (/search)

Any user can search and browse the items posted on the website. When browsing through the items they can see images of items. While in this page the user have many options to choose from. From the left side panel they can filter using price range, locations, and categories. User can click on each item and a dialogue box will pop up with detailed information about the item.
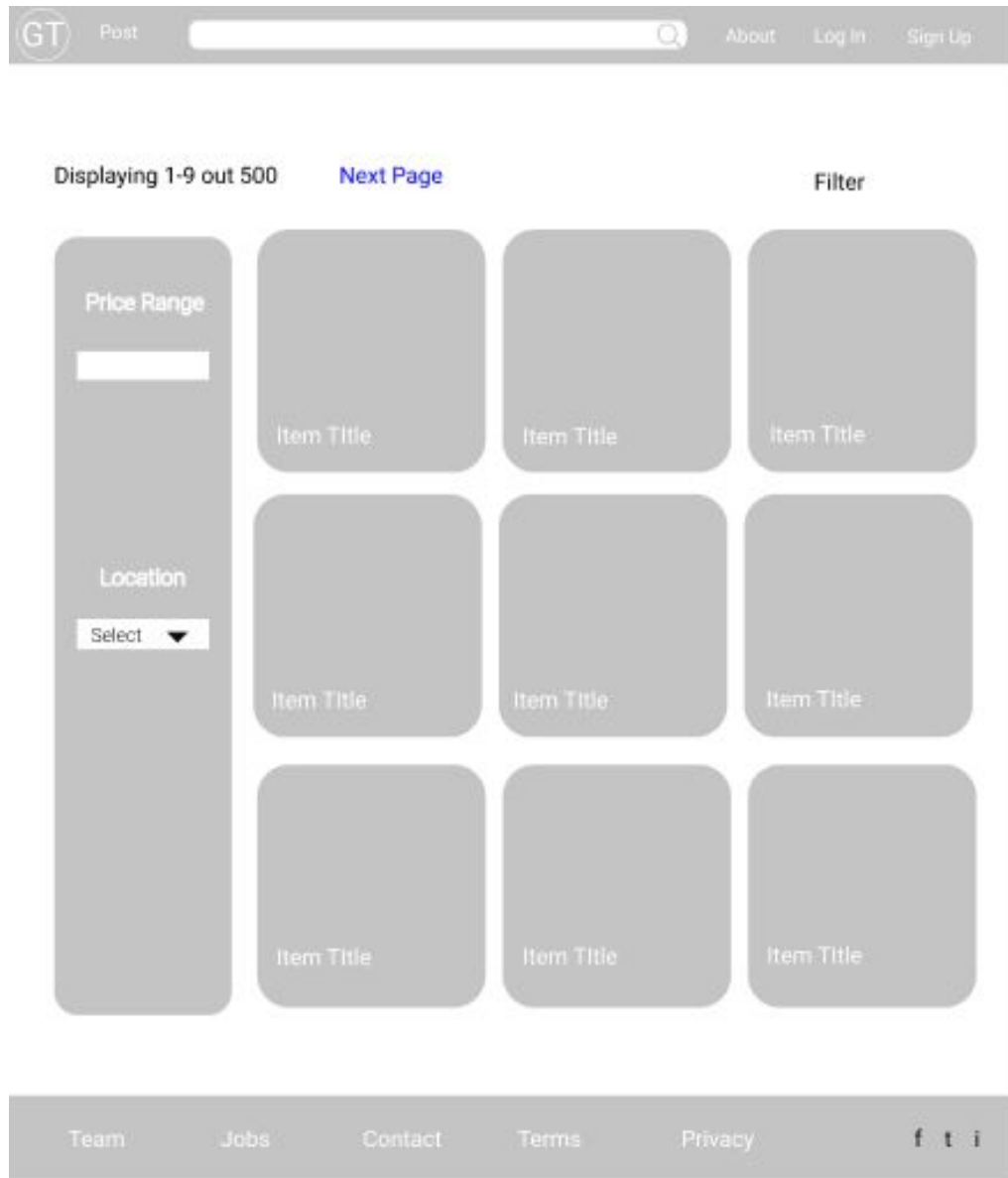
3. Item Details Page(/itemiddetails)

Buyer can click on item of interest and modal pops up showing the items title, price, location, description as well as a message box at the bottom. If user sends message and they are not signed in the website will prompt them to either sign in or create an account so they can message the seller.

4. Registration Page(/signup)

An unregistered user can create a new 'GatorTrade' account by clicking on 'Sign Up' link on the top right corner of page. When clicked, a registration page will open where user has to enter his/her information.
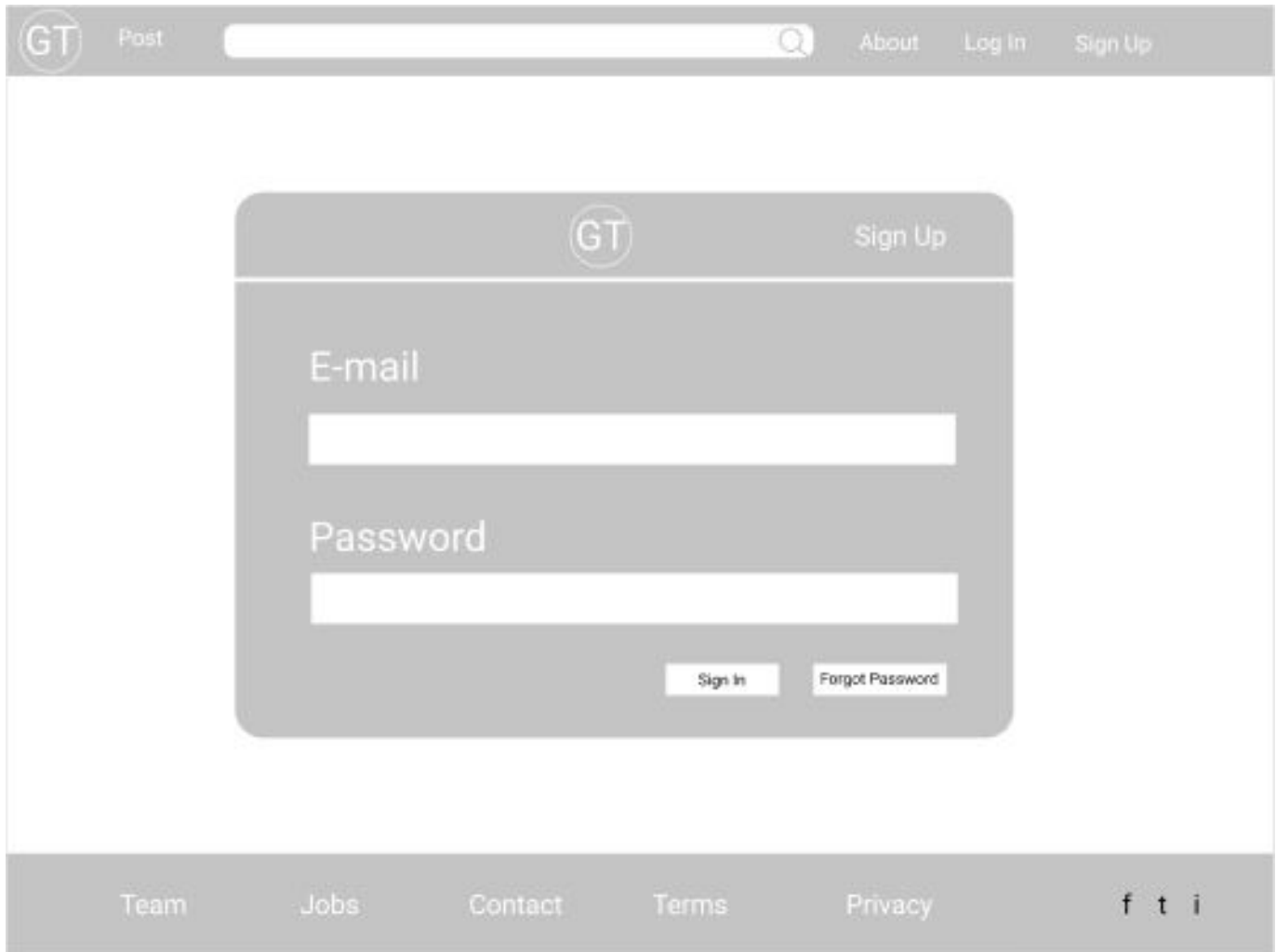
5. Sign In Page (/signin)

When the user wants to sign in they are prompted to enter their email and password with a forgot password option at the bottom and a sign up option at the top right corner for those who have not already created an account.

6. Post Ad Page (/post)

Registered Users can sell their product by clicking on 'Post' link from the Top Panel. Upon clicking, it navigates to the Post page, where user can add product details and upload images.

7. User Dashboard Page (/dashboard)

Registered users when selling items can check their dashboards, which contains the different messages they received as well as all of their prior posts. Registered users can then edit by clicking 'view/edit' to edit the post (which might again take 12 - 24 hours for the administrator to approve the post, if the user chooses to edit) and to view messages click 'view'.

GT   Post   About   User|Sign Out

Dash Board

My Messages

ID	Date	
		View
		View
		View
		View
		View
		View
		View

My Posts

ID	Date	
		View/Edit
		View/Edit
		View/Edit
		View/Edit
		View/Edit
		View/Edit
		View/Edit

8. Administration Page (/admin)

This is the Administrator page which consists of a user's post. The administrator shall have the ability to view their image, title, description, price, and location. If the administrator finds spam/inappropriate posts, they can click the deny checkbox for a user's post and notify the user. Otherwise, the administrator can click the approve checkbox. Approximately it will take 12 - 24 hours for the administrator to approve the post.
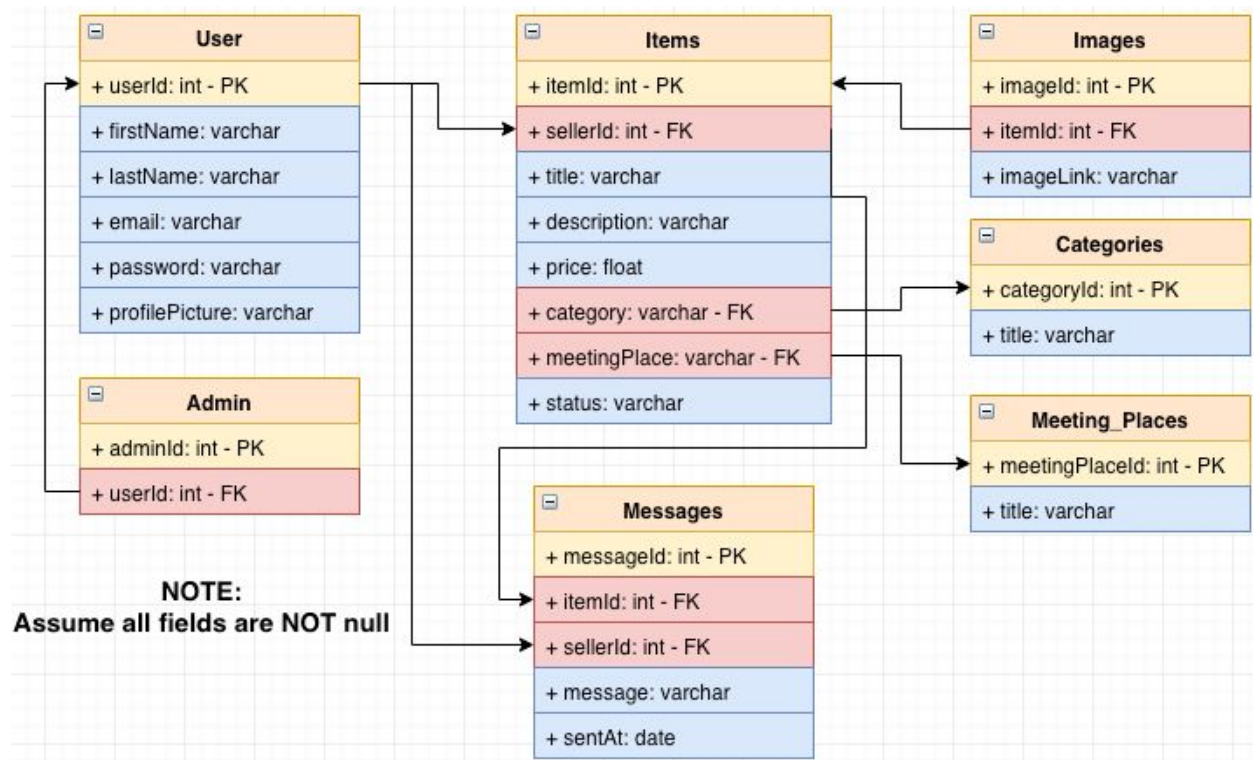
4. High Level Architecture, Database Organization

Database Organization:

Media Storage :

For GatorTrade, the only supported forms of media will be PNG Images. These images will be stored using Cloudinary, an cloud based management system for web and mobile applications. Cloudinary provides us with an API and API Key to be used within our Node app. The API allows us to take images passed to our Node server, convert them in both size and file type, and upload it to the Cloudinary server tied to our API Key. If the upload is successful, Cloudinary returns a JSON Object that includes a public link to our image, which can be stored in our database as a string. Similar to this to the screenshot posted below.

```
{
  public_id: 'cr4mxeqx5zb8rlakpfkg',
  version: 1372275963,
  signature: '63bfbca643baa9c86b7d2921d776628ac83a1b6e',
  width: 864,
  height: 576,
  format: 'jpg',
  resource_type: 'image',
  created_at: '2017-06-26T19:46:03Z',
  bytes: 120253,
  type: 'upload',
  url:
'https://res.cloudinary.com/demo/image/upload/v1372275963/cr4m
xeqx5zb8rlakpfkg.jpg',
  secure_url:
'https://res.cloudinary.com/demo/image/upload/v1372275963/cr4m
xeqx5zb8rlakpfkg.jpg'
}
```

Cloudinary also gives us the option to return our image in different sizes and resolutions, which makes thumbnails a breeze. While theoretically the links to the images will be public, Cloudinary URL randomization will make the links almost impossible to guess.

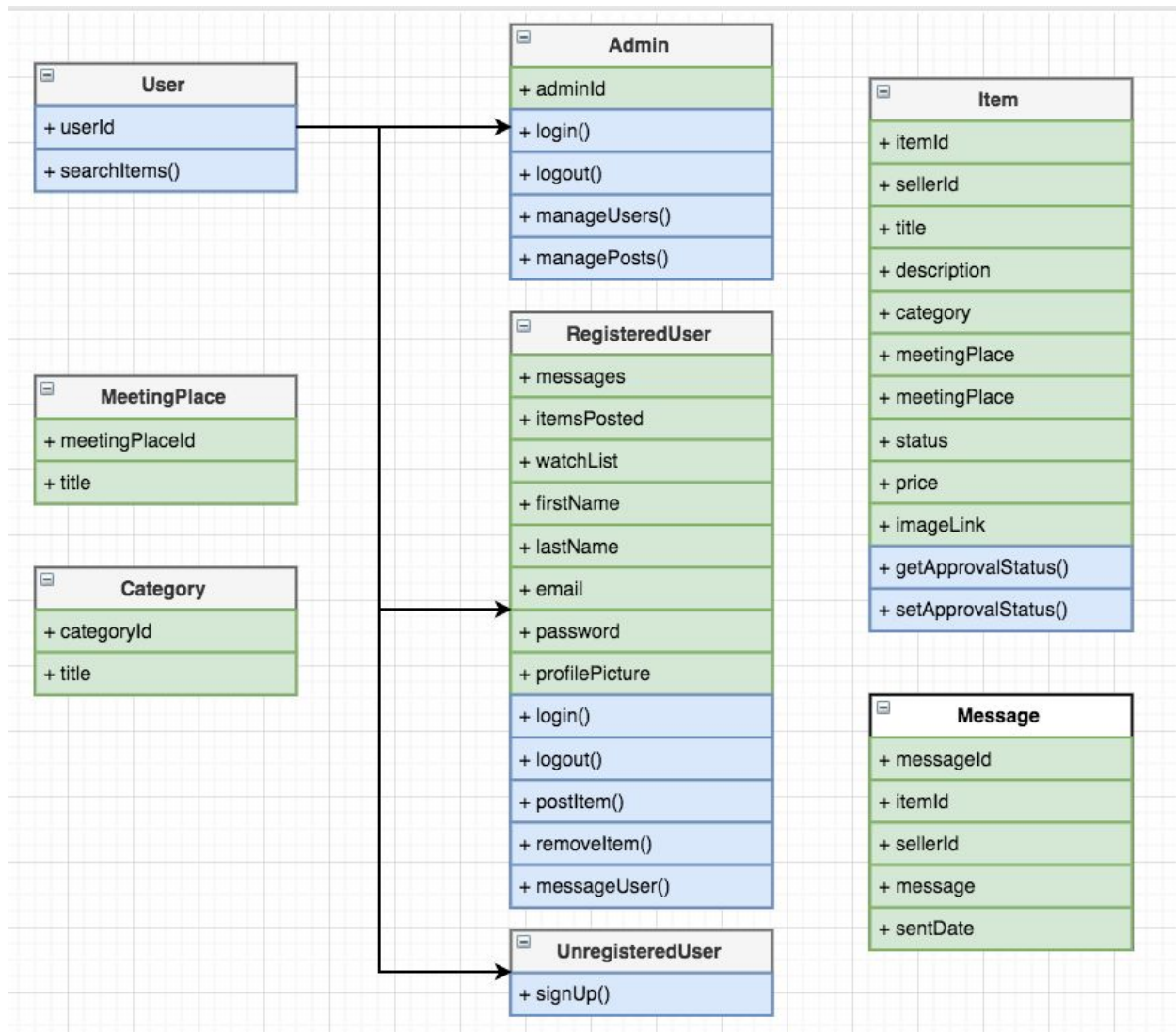**Search / Filter Architecture and Implementation:**

Using built in mysql search query options, we can use keywords passed through an input field to query our database for matches to return to the user. Using the %like mysql query will give us flexibility on how we can obtain useful information from our database in a simple and meaningful way. Since the project and databases are small in scale, we can take advantage of %like functionality without worrying about a slow response speed. If database grows more significant in size, then we will explore other options to improve performance speed.

New Framework Descriptions:

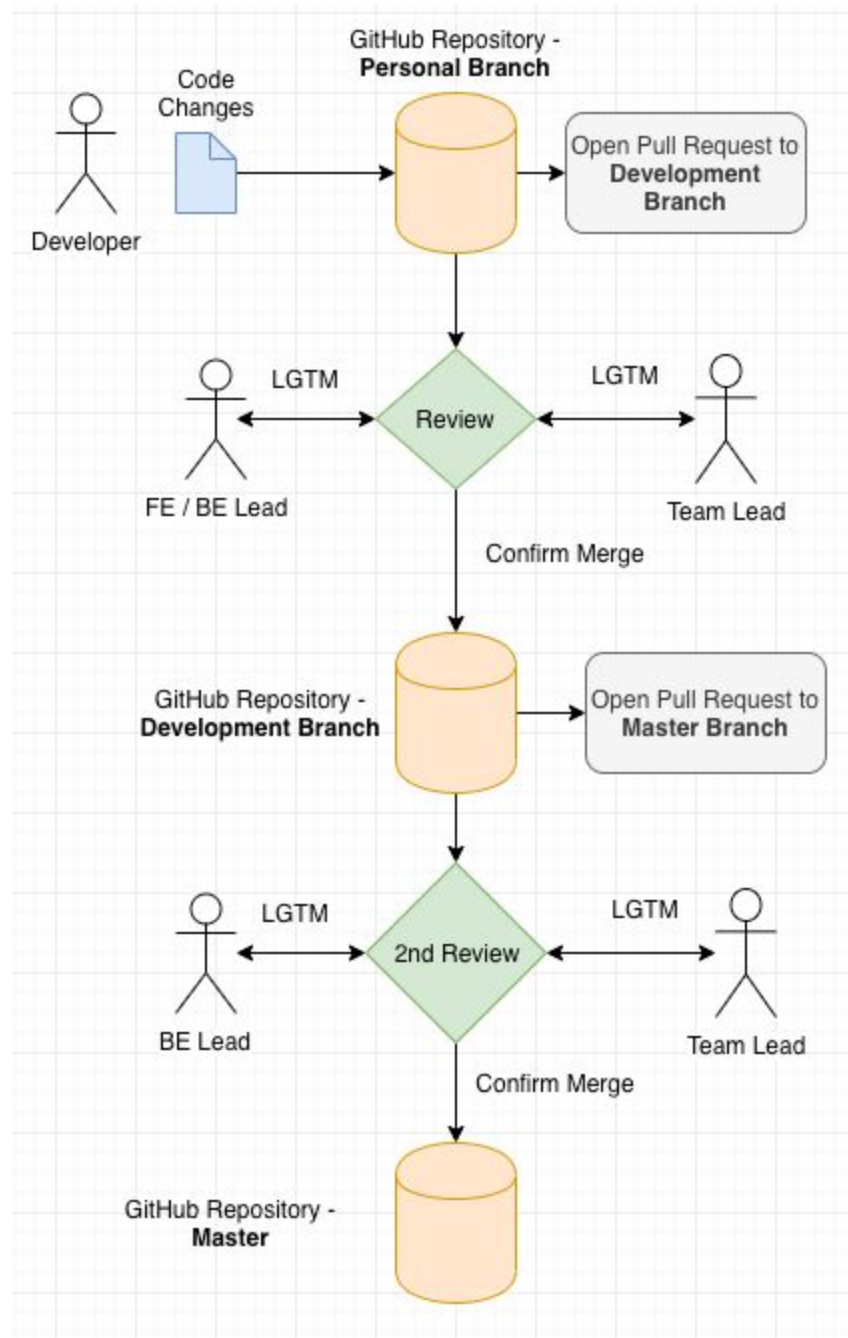**Cloudinary --** End-to-End image management solution for web and mobile applications.

5. High Level UML Diagrams
- Core Functionalities

User
+ userId
+ searchItems()

MeetingPlace
+ meetingPlaceId
+ title

Category
+ categoryId
+ title

Admin
+ adminId
+ login()
+ logout()
+ manageUsers()
+ managePosts()

RegisteredUser
+ messages
+ itemsPosted
+ watchList
+ firstName
+ lastName
+ email
+ password
+ profilePicture
+ login()
+ logout()
+ postItem()
+ removeItem()
+ messageUser()

UnregisteredUser
+ signUp()

Item
+ itemId
+ sellerId
+ title
+ description
+ category
+ meetingPlace
+ meetingPlace
+ status
+ price
+ imageLink
+ getApprovalStatus()
+ setApprovalStatus()

Message
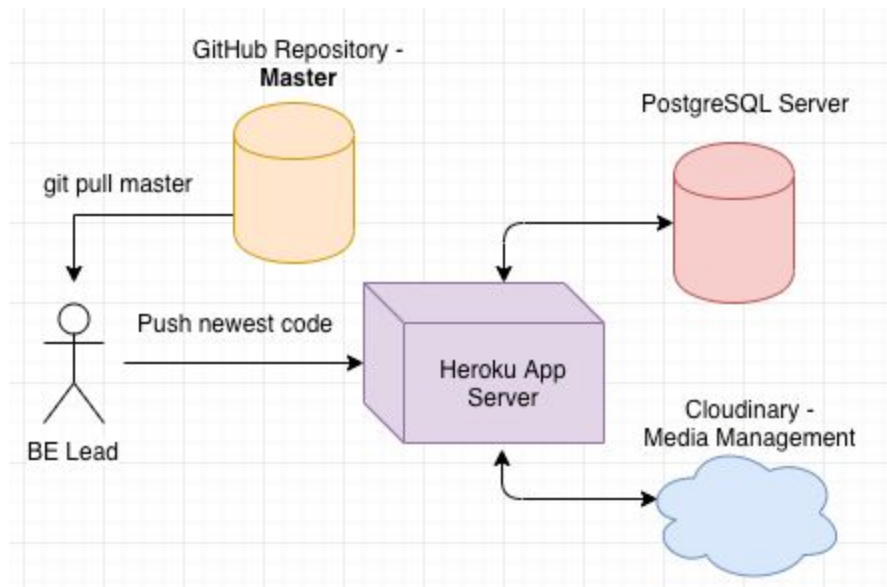+ messageId
+ itemId
+ sellerId
+ message
+ sentDate

5. High Level UML Diagrams cont.
  - **GitHub Workflow**

5. High Level UML Diagrams cont.
- **Deployment Workflow**

6. Identifying Key Risks

Skills Risks
- Front end team lacking in knowledge of bootstrap and css

Schedule Risks
- Managing other classes as well as this one

Technical Risks
- Lack of knowledge of implementing the Cloudinary API for image upload and retrieval
- Decision on which Node modules to use for querying the database

Teamwork Risks
- Commuters not always on campus to meetup

Legal/Content risks
- No foreseeable risks at this time

How do we resolve each risk
- Doing tutorials in Cloudinary, creating quick sample apps (technical risks)
- Review different node modules that allow us to interact with our database, and vote on the best approach (technical risks)
- Front End team is watching tutorials to catch up on bootstrap and css skills (skills risks)
- Google hangouts for off-campus meetings (teamwork risks)