SW Engineering CSC648/848 Fall 2019
Gator Trader - SFSU Buy and Sell Website
Section 1
Team 3 (Local)

Steve Rentschler - Team Lead (srentsch@mail.sfsu.edu)
Johnson Wong - Back-end Lead / GitHub Master
Sergei Katakhov - Front-end Lead
Ho Yin Mak (Tevis)
Tsun Ming Lee (Matthew)
Tim Chan
Osbaldo Martinez

Milestone 2
10/8/19

History Table

Date Submitted	10/17/19
Date Revised	TBD

Content and structure for Milestone 2 document for review by institutors

1. Functional Requirements - prioritized

Priority 1:

- **Unregistered user:**

    1. Shall be able to register for an account
    2. Shall be able to browse home page
    3. Shall be able to search for a product by name and/or category
    4. Shall be able to browse a product search results page
    5. Shall be able to browse a product listing page
    6. Shall be able to view the form for listing a product

- **Registered user:**

    7. Shall be able to do 1-6
    8. Shall be able to login and logout
    9. Shall be able to purchase a product
    10. Shall be able to list a product for sale
    11. Shall be able to view active listings on the dashboard page
    12. Shall be able to remove their product listing(s)

- **Administrator:**

    13. Shall be able to see listing request(s)
    14. Shall be able to approve and disapprove listing request(s)
    15. Shall be able to remove any active listing

Priority 2:

- **Unregistered user:**

    16. Shall be able to see recent listings on the home page
    17. Shall be able to see recommended products on the home page

- **Registered user:**

    18. Shall be able to do 16-17
    19. Shall be able to contact the seller of a listing
    20. Shall be able to view messages about listed products on the dashboard page

21. Shall be able to respond back to a message from a potential buyer
22. Shall be able to view purchase history on the dashboard page

- **Administrator:**

23. Shall be able to ban any account

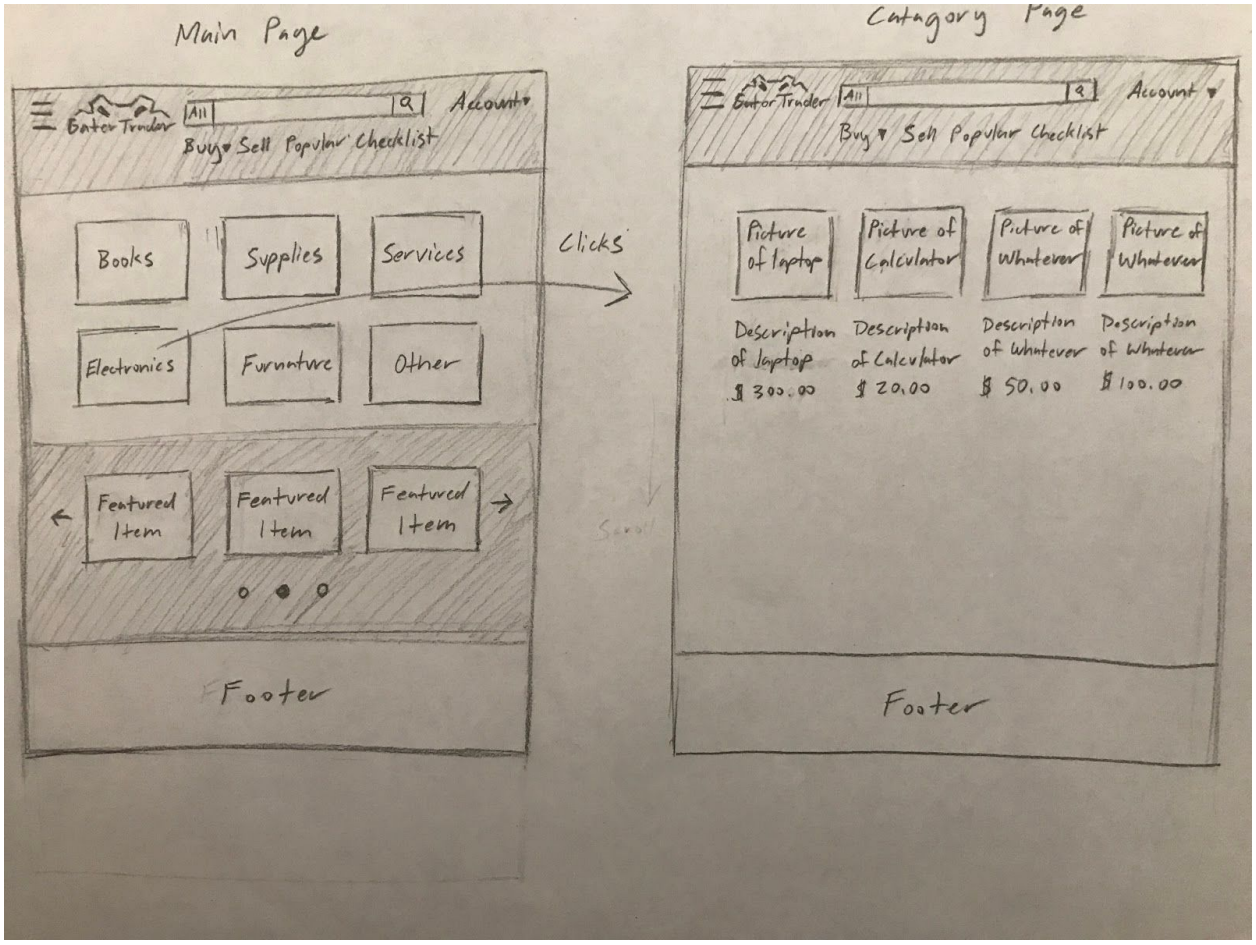
Priority 3:

- **Unregistered user:**
    24. Shall be able to search by class name and instructor's name for class materials
    25. Shall be able to see all the trending items
    26. Shall be able to use smart search to quickly find items

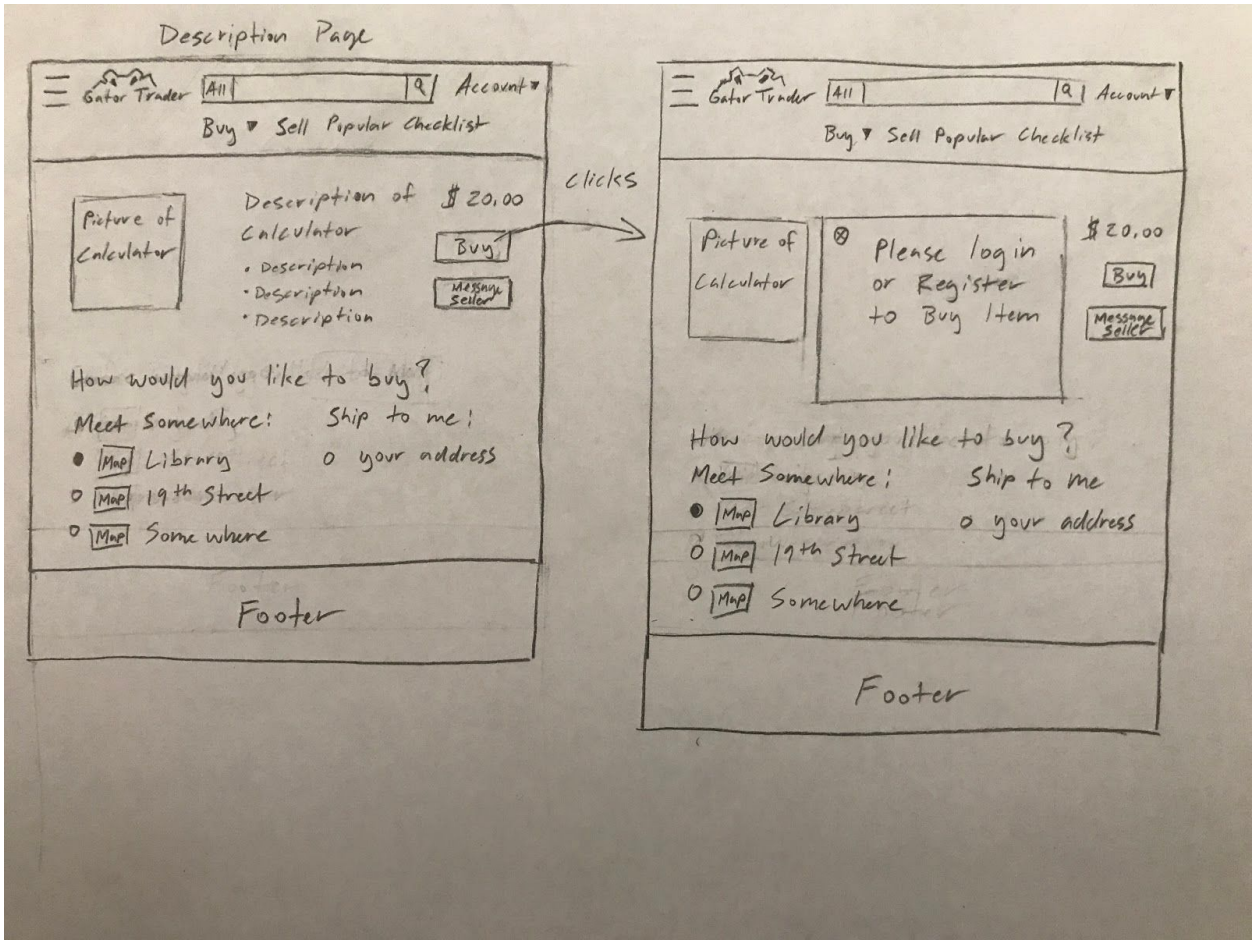
- **Register user:**

27. Shall be able to do 24-26
28. Shall be able to see a section on recommended products
29. Shall be able to locate a meet up spot
30. Shall be able to see a map that confirms their safe pickup location
31. Shall be able to change their background/theme
32. Shall be able to have a reminder/checklist notification
33. Shall be able to use smart search to quickly find items
34. Shall be able to access their spending analysis graph
35. Shall be able to see featured reviews about an item

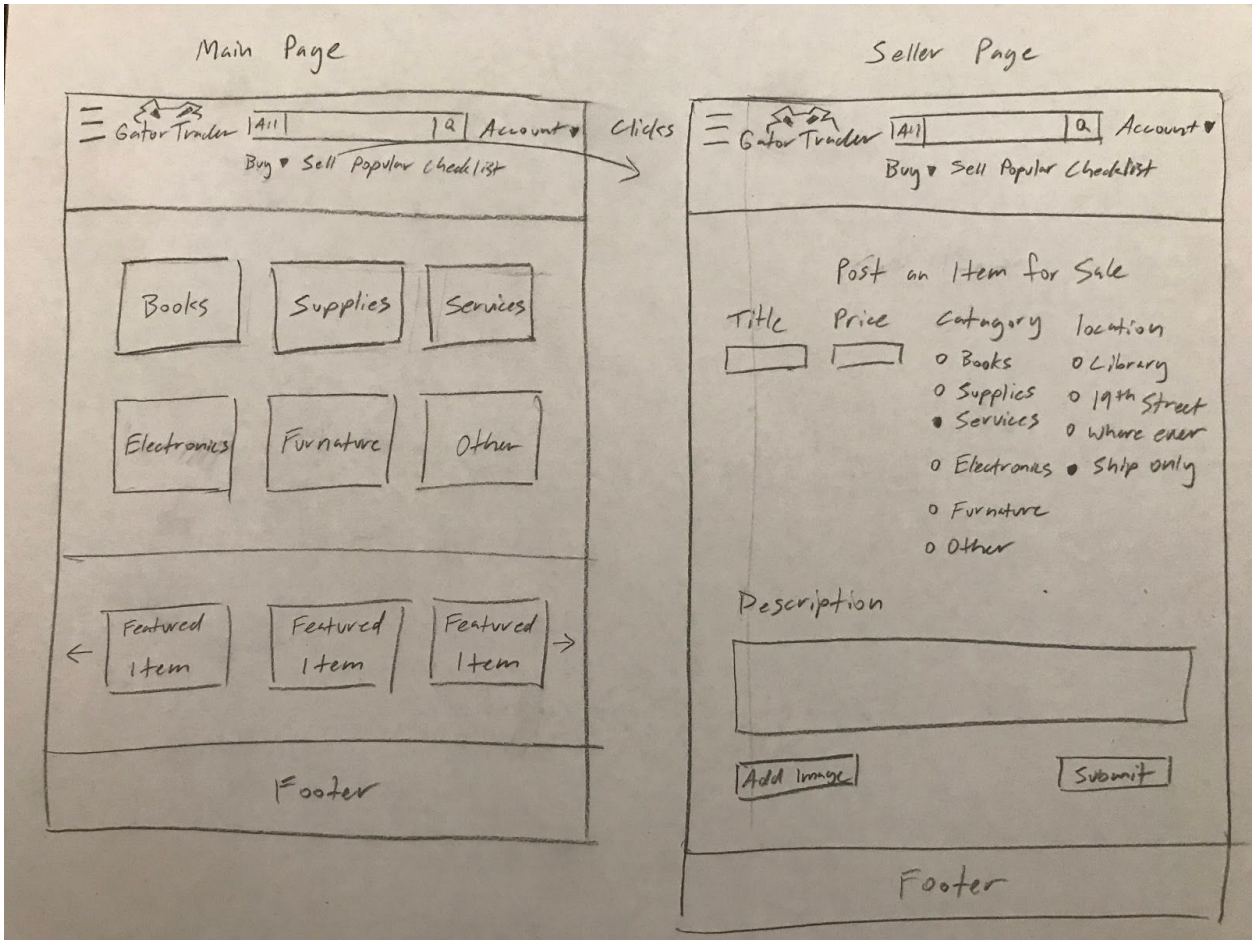
2. UI Mockups and Storyboards (high level only)



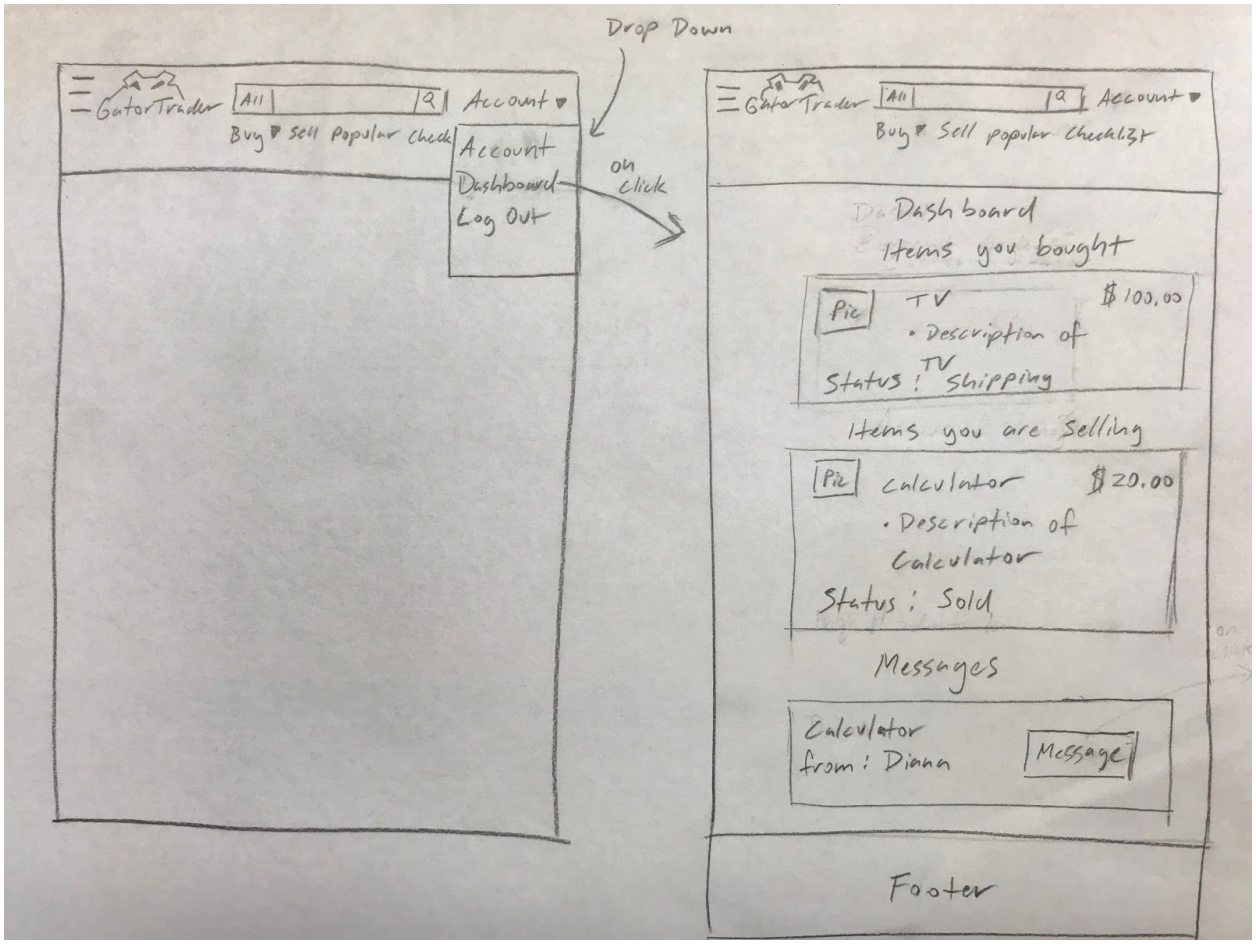
1. Diana uses her phone to navigate to the website. Upon loading the website she starts to browse through the items to see if something catches her eye. Remembering she needs a calculator, she searches for a calculator.



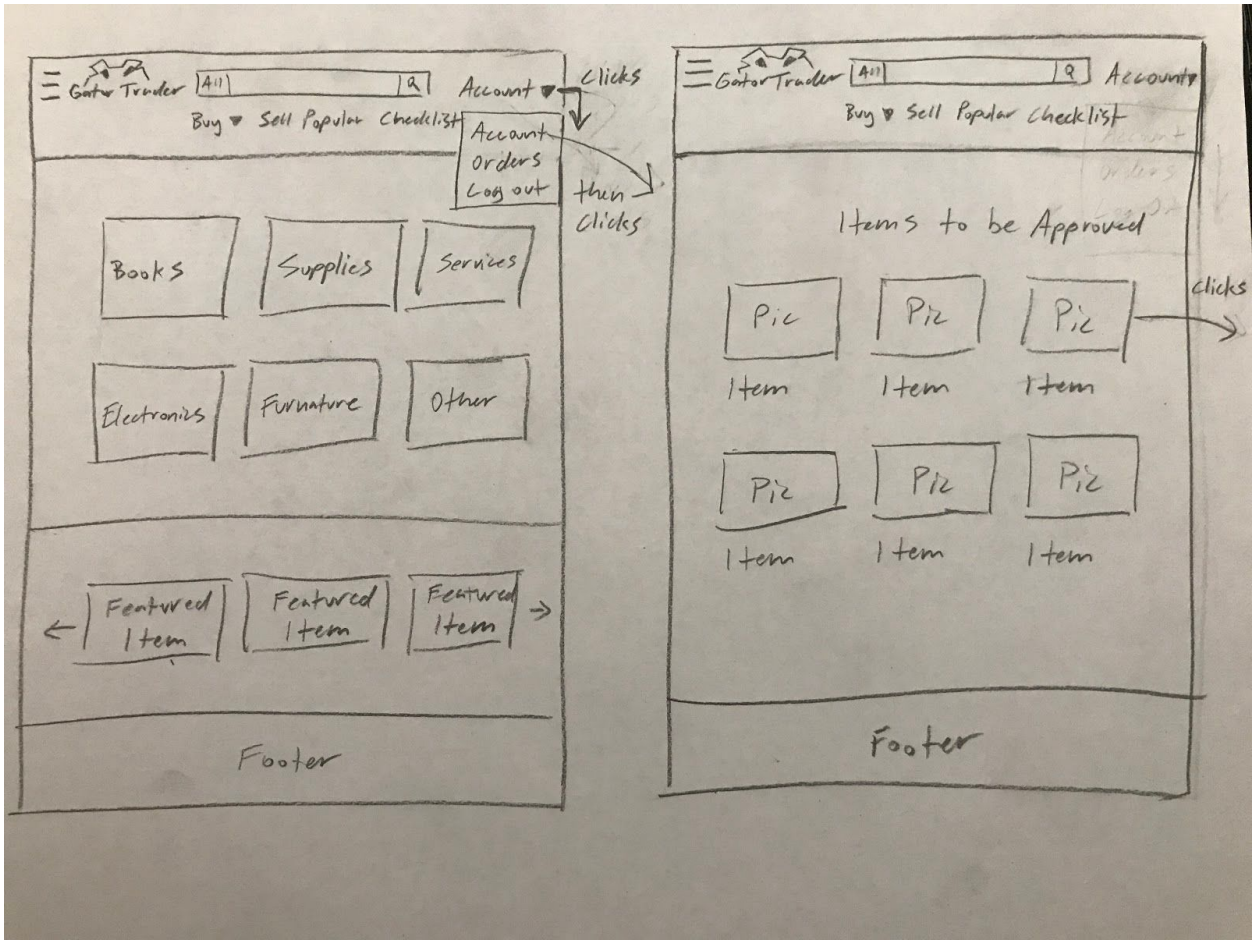
After finding the right calculator for a reasonable price, she tries to buy the calculator and she is prompted to sign up/sign in.



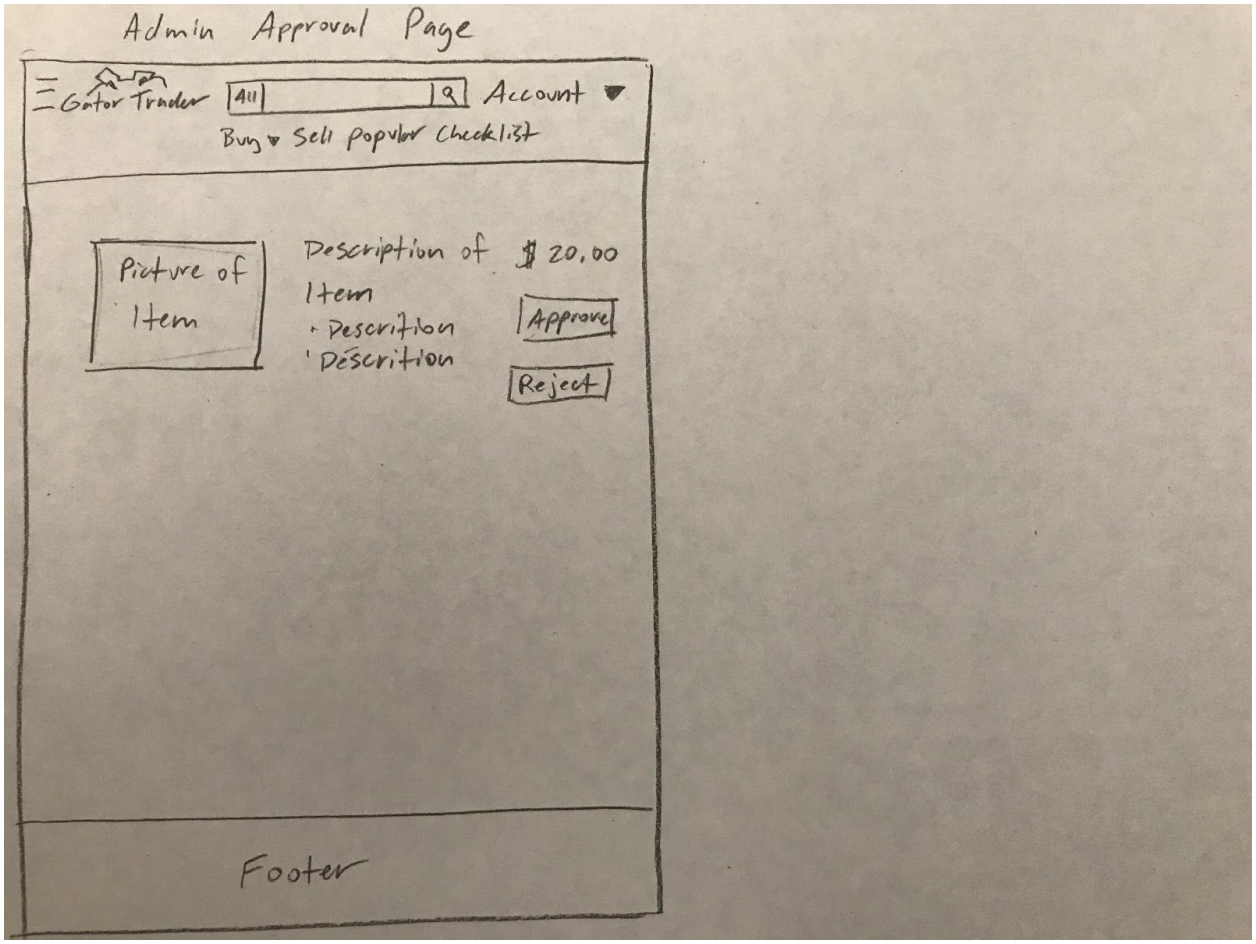
2. Mark -- knowing that he is a good note taker plus he has beautiful handwriting -- decides to make a listing for the digital version of his class notes of Calculus 3.



3. Bob goes to his dashboard to see if anything was bought. He sees that his calculator was bought and that the TV he bought is now shipping.



4. Alexander logs into his admin account for the website. He sees a bunch of items that still needs approval or rejection.



He looks at the items to make sure they are appropriate. He doesn't see any problems with the first few items, so he approves them. He stumbles upon an item with blurry images, so he can't make out the item. He rejects item and lets user know why his item was denied. After some time, he sees a listing with an inappropriate image and he rejects the item.

3. High level Architecture, Database Organization

- *DB organization*: Describe the main database schema/organization (high level), e.g. list main DB tables (e.g. their titles) and items in each DB table (check instructors' suggestions and class slides on architecture). Make sure the titles and var. names are in easy to understand plain English and consistent with data definitions in Section 1 above.

    - **RegisteredUser**(sid: int, email: string, password: string, rating: double, createdAt: timestamp, lastLoggedIn: timestamp)
    - **Administrator**(aid: int, username: string, password: string)
    - ~~**Category**(cid: int, name: string)~~
    - **Product**(pid: string, category: string, name: string, description: string, quantity: int, picture: blob, deliveryMethod: string, status: string, views: int, submittedAt: timestamp, lastModAt: timestamp)
    - **Order**(oid: int, buyer: int, soldAt: timestamp, status: string)
    - **Inquiry**(iid: int, product: string, regUser1: int, regUser2: int)
    - **Message**(inquiryId: int, message: string, time: timestamp)
    - ~~**Includes**(orderId: int, productId: string)~~
    - **Reviews**(admin: id, product: string, action: string, comment: string)

- *Media storage*: Decide if images and video/audio will be kept in file systems or in DB BLOBs (decision on file vs. BLOBs must be made by the end of M2). Describe any other special data format requirements like for video/audio/GPS etc.

    - Will be using BLOBs for media storage

- *Search/filter architecture and implementation*: what will be the alg/SW for search; how will you organize search items from the user; what DB terms will be searched, how it will be coded and organized in the DB (check instructors' suggestions in the class. OK to use SQL and %like).

    - For search we will be using SQL queries and %like to search items that best match the keywords the user entered. The search results will be based on the best results of the item's title and/or description.

- Your own APIs (if any): Describe and define at high level any major APIs that you will create other than standard ones provided by tools and frameworks you use

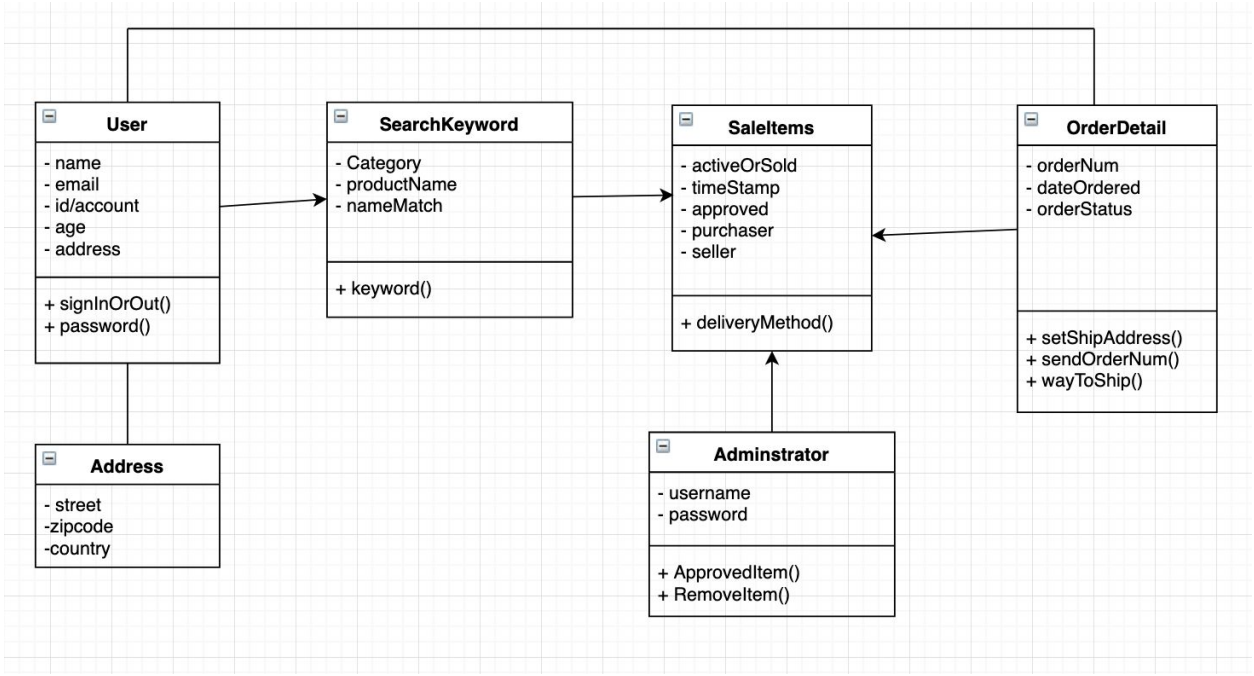
    - Create user: creates a new user

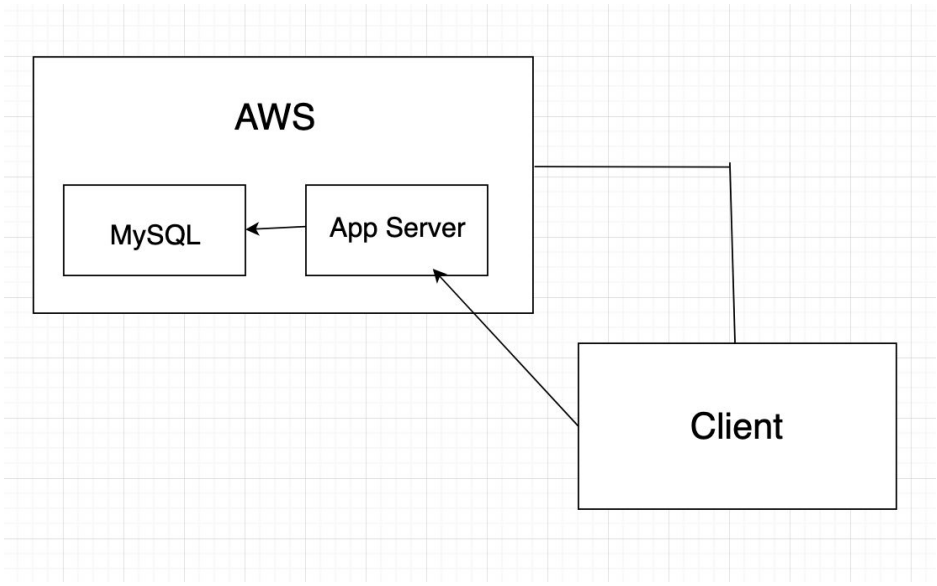
    - Create new item: creates a new item posting

- Describe any significant non-trivial algorithm or process if any (like rating, ranking, automatic prioritizing of items etc.)

    - Ratings will be calculated by taking an average of all the ratings.
    - Ranking will be calculated by taking into account the average rating and the number of ratings.
    - Items will be automatically prioritized by taking into account the newest postings and pricing.

4. High Level UML Diagrams





5. Identify *actual* key risks for your project at this time

a. Skills Risks:

For this project – a web application – our team has just a few students who have experience in web development. The rest of the team will need to spend extra time to do research on what they are required to learn in order to finish the project.

Our strategy to resolve these risks involves seeking help online, getting a consultation from our CEO – Professor Petkovic – and CTO - Anthony Souza –, receiving assistance from teammates, and reading relevant materials.

b. Schedule Risks:

All team members are full-time students and, as a result, we are able to meet only once or twice per week (besides our regular hours in class). However, we are highly motivated and committed to finish the project on time with the quality in mind.

In order to achieve that, we will be utilizing all possible methods of communication in our disposal such as emails, Discord, text messages, and Trello. This will help us to be more productive and efficient even with a shortage of in-person meetings.

c. Technical Risks:

We will be implementing a popup-window messaging system and spending analysis (which will be represented in a form of graphs) as our signature application features. Both of these are still under research from our team since we have never implemented something like this before. We will be allocating some extra time to make sure that these features are working properly and – at the same time – improving users' experience.

d. Teamwork Risks:

Having a relatively large team, we might face different types of conflicts such as work commitments, communication issues, workload allocation, or differences in opinion. During the implementation and especially at the end of the semester (when the stress usually builds up), we will try to be patient, respectful, supportive, and understanding to our teammates, no matter what happens.

e. Legal/Content Risks

The content that users will be uploading might contain inappropriate or illegal materials that should not be posted. In order to avoid that issue, all new listings will be sent to an Admin. He or she will be checking if these listings have proper content and will be giving permissions to post it on the website.