MyUniMarket

CS 307 Design Document

Team 24

Sakshi Choudhary

Giorgi Khmaladze

Nick Litman

Mihir Somani

Conley Utz

Table of Contents

Purpose

Online shopping has become a common activity in the last two decades. Even though there are many different services that one can choose from, college students still struggle to find the best deals on supplies they need in a centralized storefront. This has led to an increased demand for a platform that will allow college students to buy everything they may need at one convenient website.

The purpose of this project is to develop a web application for the reasons stated above. Already existing platforms like Facebook Marketplace, for example, have similar functionalities to our product, but they are inefficient and frustrating to use as they are not solely devoted to the sale of college supplies. MyUniMarket will provide a service that will solely focus on Purdue University students. This web application will allow users to buy and sell college materials that they will need throughout the semester.

Non-Functional Requirements

1. **Architecture and Performance**

   As a developer,

   a. I would like my server to handle 500 simultaneous requests
   b. I would like to develop the backend API in object-oriented and open source PHP and MySQL using a modular approach to independently test each component.
   c. I would like to have a seamless frontend for a crisp and clean web application.

2. **Security**

   As a developer,

   a. I would like to include security features to prevent against popular attacks such as MySQL Injection and Distributed Denial of Service.
   b. I would like to store passwords as hashes in the MySQL Database.
   c. I would like to allow only the authorized users to view or modify their listings.

3. **Utility**

   As a user,

   a. I would like to have an effective and efficient user experience.
   b. I would like to solely focus on selling used school products in a local domain.

4. **Hosting & Development**

   As a developer,

   a. I would like to use Heroku for the deployment of our web application as they have a simple cloud platform for development teams.
   b. I would like my frontend to involve HTML5, CSS3, and Javascript in unison to create the best user experience possible along with backend in PHP and MySQL to provide a high-uptime guarantee.

Functional Requirements

1. **User accounts**

   As a user,

a. I would like to be able to register for a MyUniMarket account with a purdue.edu email
b. I would like to be able to login to my MyUniMarket account
c. I would like to be able to sign out of my MyUniMarket account
d. I would like to be able to edit my account information
e. I would like to be able to recover my password in the event it is lost
f. I would like to be able to delete my account
g. I would like to be able to view other users' profiles

2. **Listings**

   As a buyer,

a. I would like to be able to view listings for items
b. I would like to be able to search for items using keywords
c. I would like to be able to sort items by category
d. I would like to filter items by parameters such as price
e. I would like to be able to bookmark a listing **(if time allows)**
f. I would like to be notified if one of my bookmarked posts has been sold **(if time allows)**
g. I would like to be able to see how many people have viewed a listing **(if time allows)**

   As a seller,

h. I would like to be able to list items for sale
i. I would like to be able to provide the quality of my item for sale
j. I would like to be able to set an asking price of my item for sale
k. I would like to be able to write a description of my item for sale
l. I would like to be able to edit my listings after they are posted
m. I would like to be able to take down my listings
n. I would like to be able to categorize my item for sale

o. I would like to be able to display my location on my posts
p. I would like to be able to remove my items after completing a transaction
q. I would like to see how many people have viewed my listings **(if time allows)**

3. **Communication/Misc.**

As a user,

a. I would like to be able to easily navigate the website
b. I would like to be able to utilize the MyUniMarket website across all types of devices
c. I would like to avoid seeing inappropriate words/phrases on the marketplace

As a seller,

d. I would like to be able to communicate with a potential buyer via private email

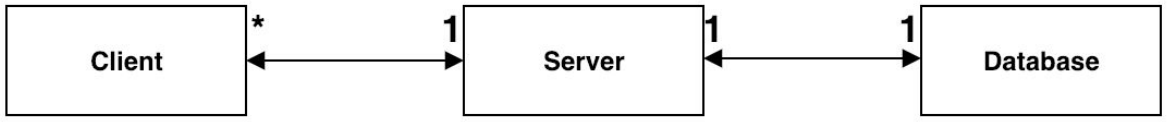
As a buyer,

e. I would like to be able to view seller ratings
f. I would like to be able to rate the seller by the quality of the item and transaction
g. I would like to be able to communicate with a seller via private email
h. I would like to know the location of the seller in order to find the most optimal listing

Design Outline

High-Level Overview

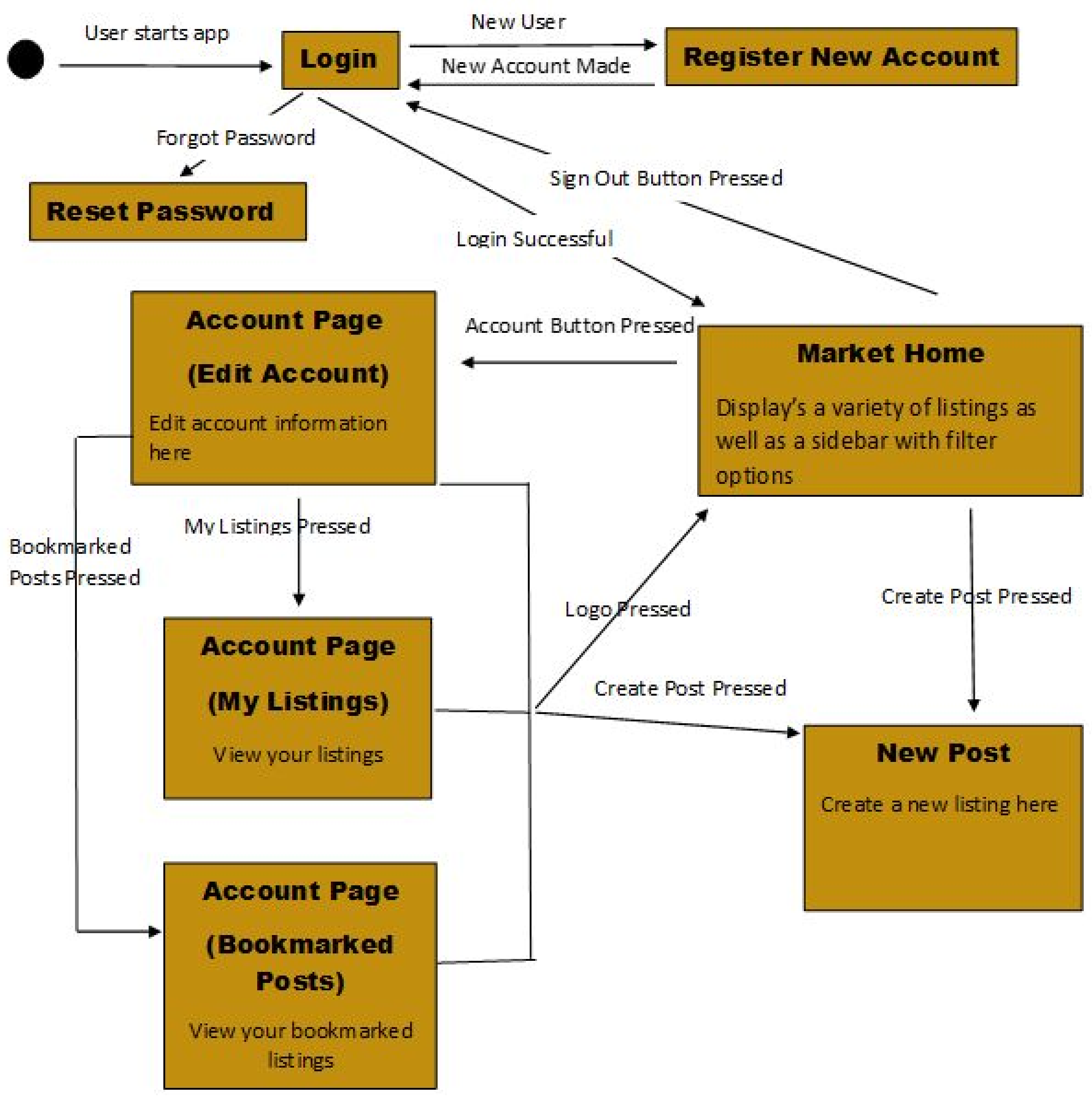
This web application provides a marketplace for sellers to post items for sale with appropriate pricing and location while providing buyers with the ability to view items available for sale. It also allows users to rate each other based on the quality and truthfulness of their transaction. The project will use a client-server model, where a server will process requests from multiple clients at a time. The server processes requests from clients, queries the database to access, store, or delete information, and responds to the client accordingly.



1. **Client**
    a. Client provides the user with an interface to the web application.
    b. Client sends requests to the server which are handled by a PHP based backend.
    c. Client receives responses to its requests from the server and renders the interface in response to them.

2. **<u>Server</u>**
    a. Server receives requests from the client and handles them.
    b. Server sends queries to the database to access, store, or delete data based on the request.
    c. Server generates responses from the to database queries and sends them to the client.
3. **<u>Database</u>**
    a. Database stores all the data used in the application such as item listings, user information, etc.
    b. Database receives queries from the server and performs operations such as requesting, modifying or deleting data.
    c. Database sends the response of the query back to the server for feedback.

Sequence of Events Flowchart



Sequence of Events Description

The design of our flow presents how simple and straightforward it is to access and use the website. The user is prompted to log in when the first page is requested. If the user does not have an account yet they are redirected to a page where they can create a new account. If creation is successful they are redirected back to the login page. If the user forgot the password to their account they are redirected to the reset password page. When the login is successful, the user is redirected to the market homepage. From the homepage, there are three ways to go. If the user clicks the “create post” button they will be redirected to a page where they can do so. If the user chooses to press the account button they are redirected to the account page where they can edit their account and see the content they have chosen to save and post. Pressing the sign out button will redirect the user to the login page. Pressing the logo on all of the pages, except login, register, and forgot password pages, takes the user to the market home page.

Design Issues

Functional Issues

1. **What information do we require for signup?**

   Option 1: Name, Email, and Password

   Option 2: Name, Email, Password and Retype Password

   Option 3: Name, Email, Password, Retype Password, Mobile Number

   Our Choice: Option 2.

   Justification: During the signup process, email and password are required fields since they're used to uniquely identify every user and provides a mechanism for login functionality. Retyping a password field ensures that the user doesn't make any typos in their password during the signup process. The name field will be useful in giving every user a unique username. We decided to not include a field for a mobile phone number since our web application will provide an interface for communication between users via email thus making mobile number an unnecessary field.

2. **What filter options do we provide for Items?**

   Option 1: Category, Price

   Option 2: Category, Price, Seller Ratings

   Option 3: Category, Price, Quality

   Our Choice: Option 3.

   Justification: Sorting items by category and price (Lowest to Highest or Highest to Lowest) are essential filters needed for any item. We decided on using quality as our third filter since most of the items on the marketplace will be second-hand and therefore the quality of the item will be important to help users make well-informed purchasing decisions. However, seller ratings wouldn't be as useful to a buyer since we will only allow Purdue students to use the web application, which will give the buyer the guarantee that the seller will be a Purdue student.

3. **How do we want our Buyers and Sellers to communicate?**

   Option 1: Chat Room

   Option 2: Email

   Option 3: Text Messages

   Our Choice: Option 2.

   Justification: We decided on providing an email API for communication between buyers and sellers since we're already using purdue.edu emails for signup and login functionality. Using email as the platform for communication will ensure that any communication related to the web application will be in one convenient place - email.

Non-Functional Issues

1. **What backend web service are we going to use?**

   Option 1: Heroku

   Option 2: Amazon Web Services

   Option 3: Digital Ocean

   Our Choice: Option 1.

   Justification: Heroku is a free hosting service (with a student email) which has seamless GitHub integration where we can easily auto-deploy our website. Heroku has many options and tutorials for setting up PHP applications. Digital Ocean can be expensive, and our group is unfamiliar with Amazon Web Services making Heroku the best option.

2. **What backend language are we going to use?**

   Option 1: Node.js

   Option 2: PHP

   Option 3: Django

   Our Choice: Option 2.

   Justification: PHP is an open source object-oriented server-side scripting language which provides flexible integration with HTML, CSS, and JavaScript which we'll be using for our front-end. Moreover, our MySQL database has open-source connectors for PHP along with plenty of documentation.

3. **<u>What type of Database are we going to use?</u>**

   Option 1: MySQL

   Option 2: MongoDB

   Our Choice: Option 1.

   Justification: MySQL is an open-source relational database management system(RDBMS). Since MySQL libraries and queries are well-documented and provide smooth integration with PHP, MySQL was an easy winner for our back-end architecture.

4. **<u>How are we going to implement our frontend?</u>**

   Option 1: HTML, CSS

   Option 2: HTML, CSS, Bootstrap

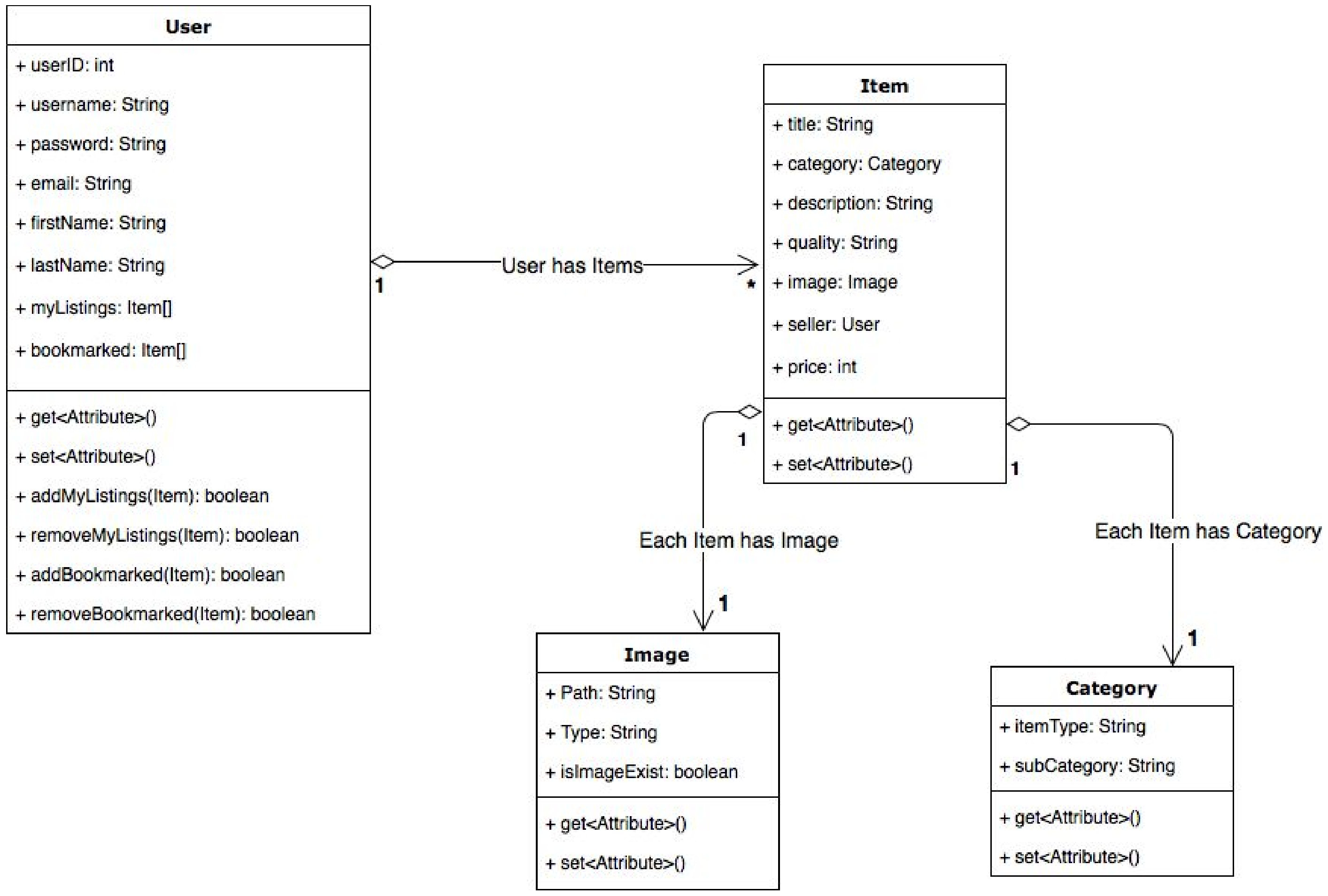
   Option 3: HTML, CSS, Bootstrap, JavaScript

   Our Choice: Option 3.

   Justification: HTML and CSS are essential for creating any basic website. Bootstrap (a CSS library) will help us focus on the implementation of our client-server architecture and performance rather than the look and feel of the UI. Since, HTML, CSS, and Bootstrap only allow us to create static websites, we decided to include JavaScript to our front-end stack as it will enable us to do client-side processing on the web application proving a dynamic, modern, and seamless user interface.

Design Details

Class Diagram



Class Interaction and Description

Since PHP is object-oriented, the classes of our web-application are based on each object. Each object has fields, including fields of other instances of classes as well as methods which perform computations on the fields of another object.

1) <u>User</u>

a) A user is created when a user is logged in.
b) A user object defines a buyer/seller.
c) To uniquely identify each other, each user is assigned a unique user ID.
d) To smoothly perform login functionality, the user object contains fields for username and password.
e) Information for the profile of the user will require fields like first name and last name which are included in the object's fields.
f) Each user has an array of listings they posted for sale.
g) Each user has an array of listings they bookmarked for future reference.
h) The User object contains methods which allow to add or remove listings posted by the user for sale.
i) The User object contains methods which allow adding or removing bookmarked listings saved by the user for future reference.

2) **<u>Item</u>**

   a) An Item defines a listing posted by a seller and viewable by a buyer.
   b) Each Item contains a title of the listing.
   c) Each Item contains a category field to group multiple items of the same category together.
   d) Each Item contains a description field for more-detailed information about the item.
   e) Each Item contains an image field which represents the thumbnail of the item.
   f) Each Item contains a seller field which links this Item to its seller.
   g) Each Item contains a price field which denotes the cost of the item.

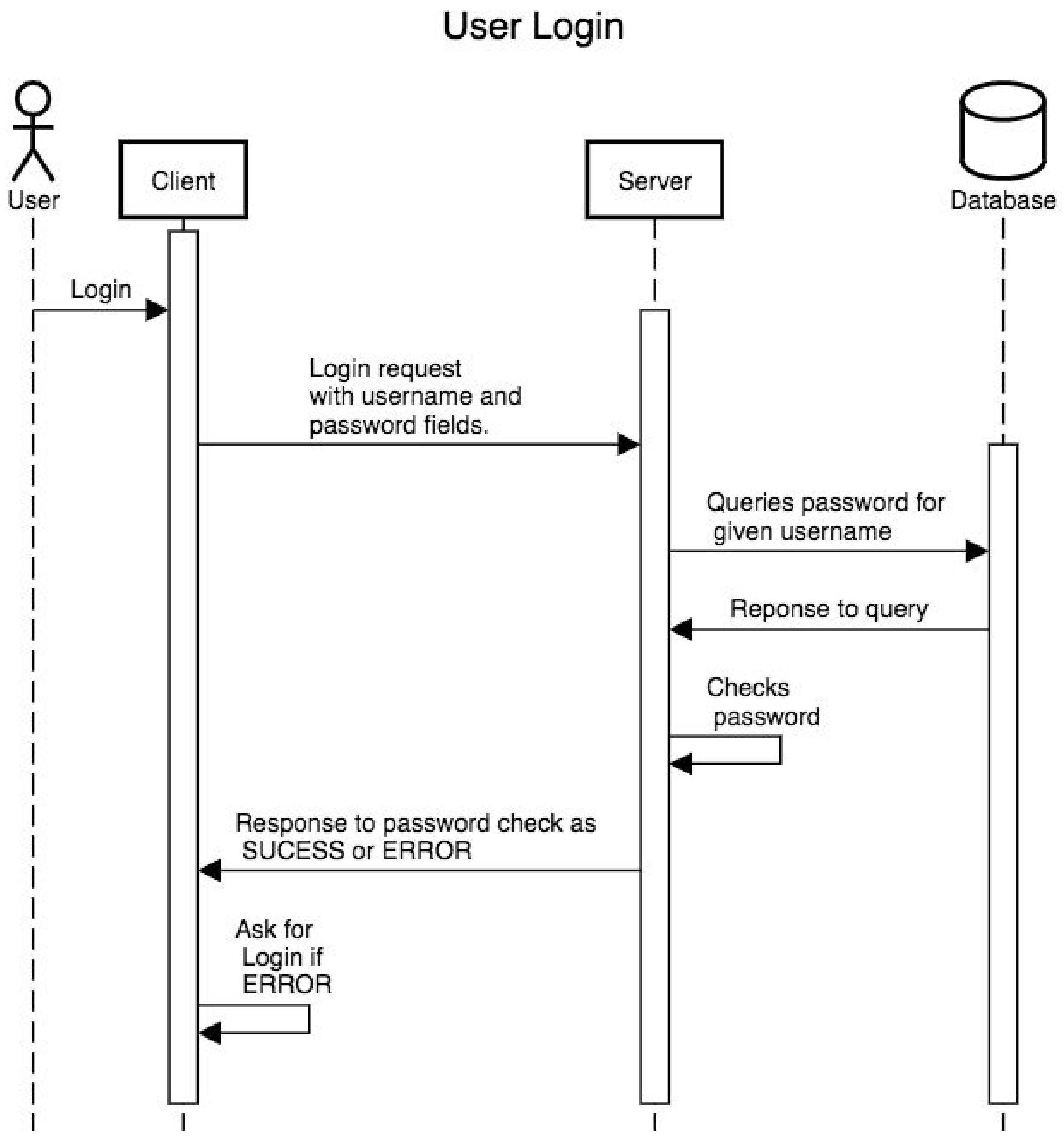
3) **<u>Image</u>**

   a) The Image object represents the thumbnail of an Item listing.
   b) It contains a field for the Path which represents where the image is stored locally on the server.
   c) It contains a boolean field which might be used when the seller decides not to upload an image for his item.

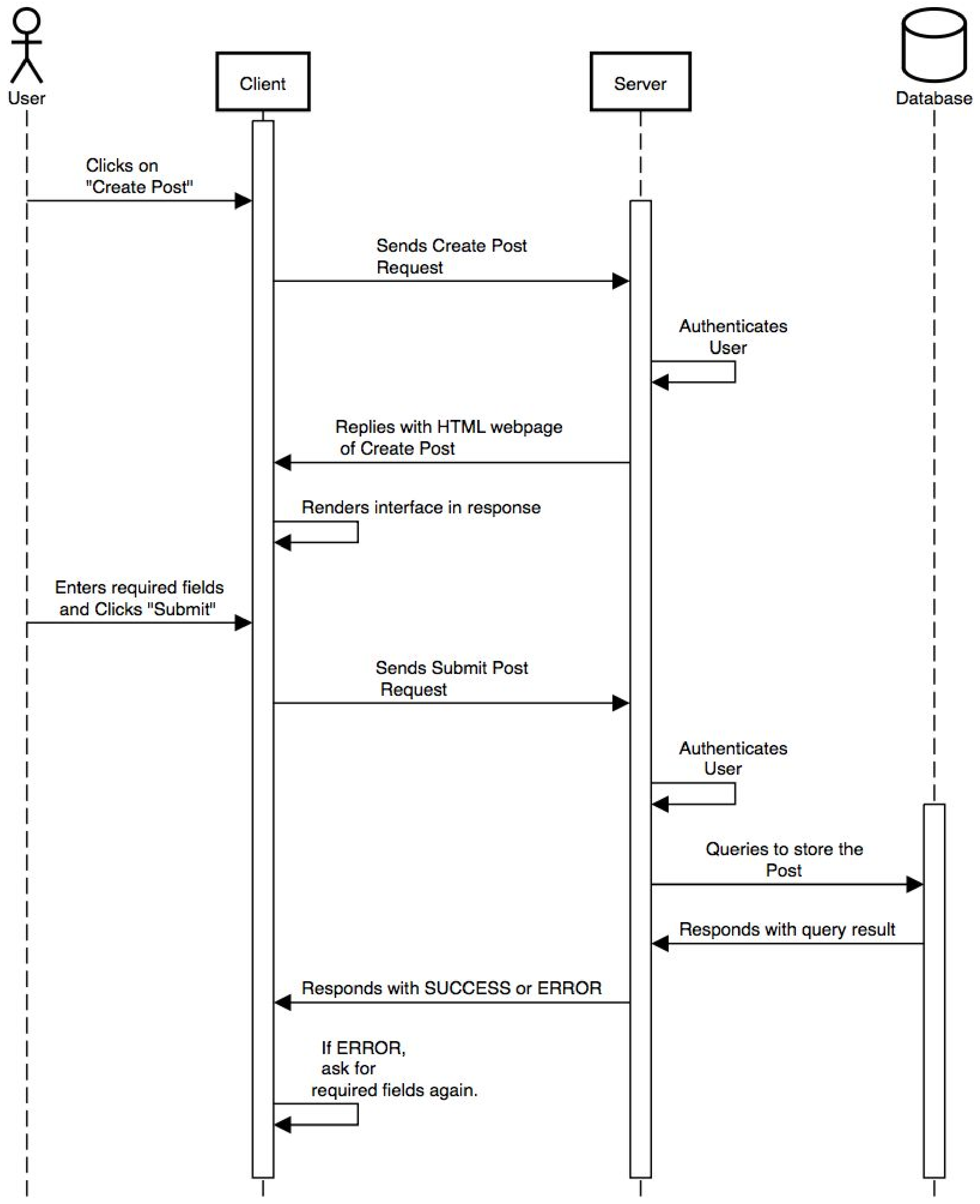
4) **<u>Category</u>**

   a) The Category object represents the category of an Item
   b) It contains a field which defines if the item is a textbook, iClicker, etc.
   c) It contains another field which defines the subcategory of the item to help with navigation and searching by category.

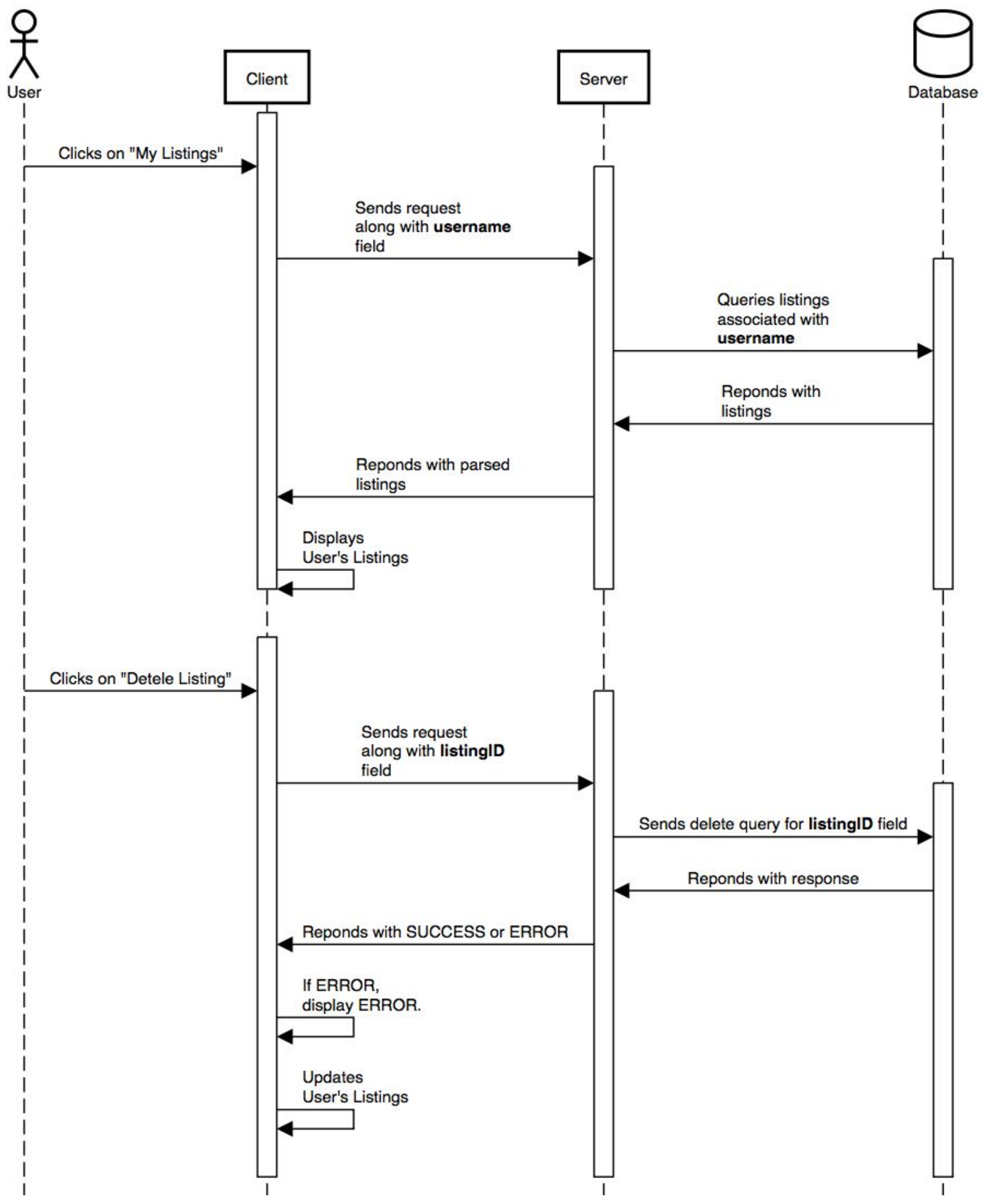
Sequence Diagrams


User Login
User
Client
Server
Database
Login
Login request with username and password fields.
Queries password for given username
Reponse to query
Checks password
Response to password check as SUCESS or ERROR
Ask for Login if ERROR

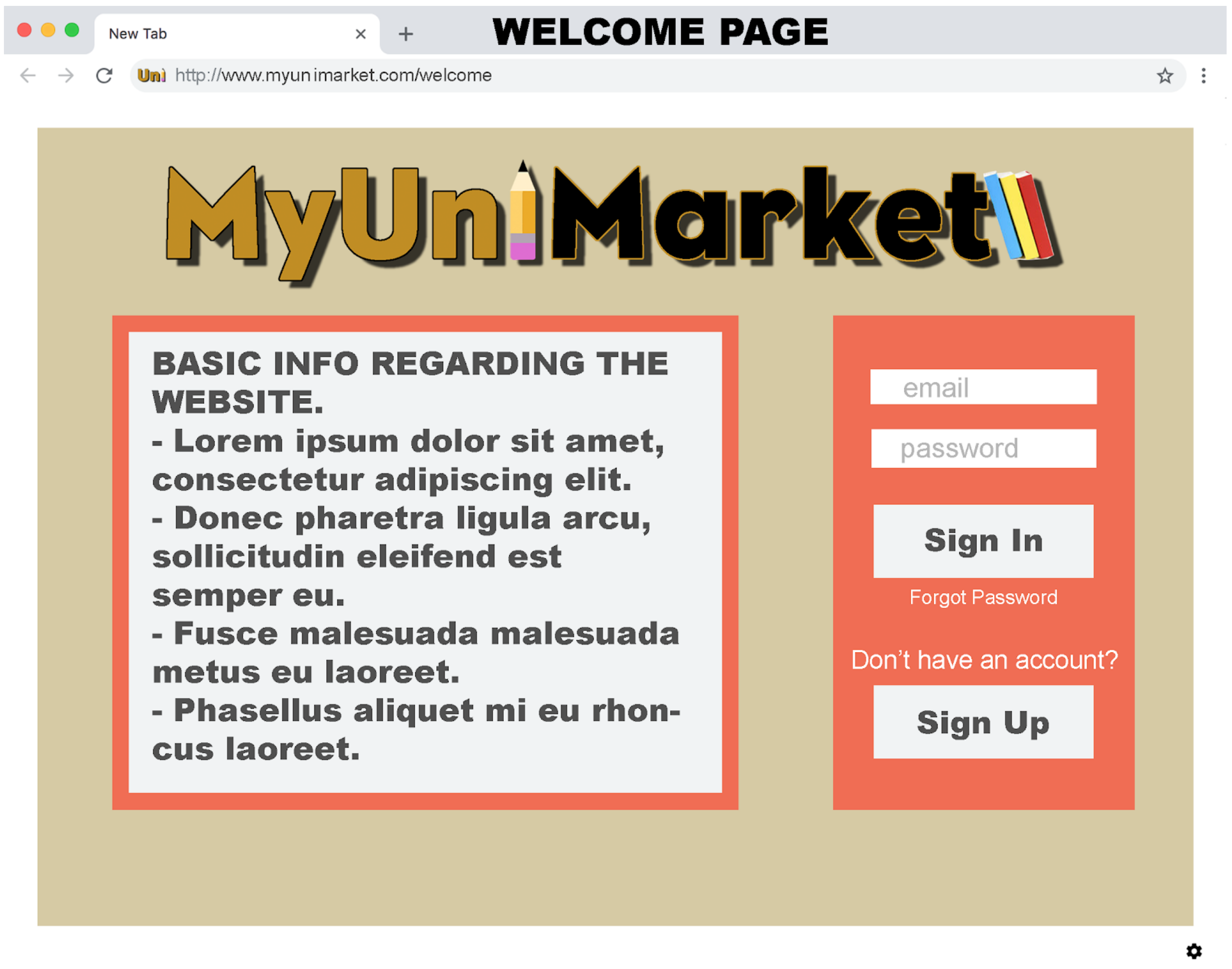
Posting an Item


User
Client
Server
Database
Clicks on "Create Post"
Sends Create Post Request
Authenticates User
Replies with HTML webpage of Create Post
Renders interface in response
Enters required fields and Clicks "Submit"
Sends Submit Post Request
Authenticates User
Queries to store the Post
Responds with query result
Responds with SUCCESS or ERROR
If ERROR, ask for required fields again.

My Listings

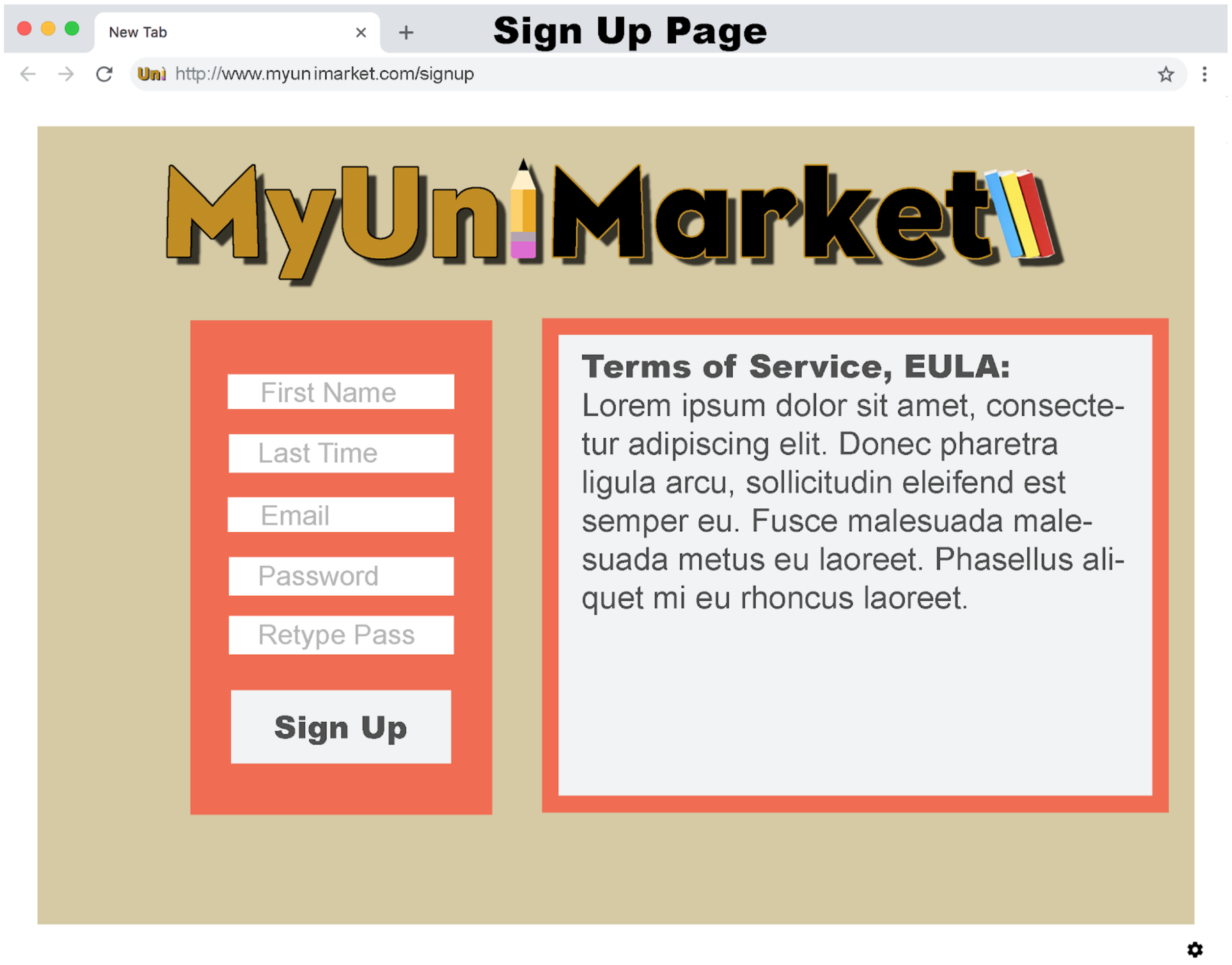

User
Client
Server
Database
Clicks on "My Listings"
Sends request along with **username** field
Queries listings associated with **username**
Reponds with listings
Reponds with parsed listings
Displays User's Listings
Clicks on "Detele Listing"
Sends request along with **listingID** field
Sends delete query for **listingID** field
Reponds with response
Reponds with SUCCESS or ERROR
If ERROR, display ERROR.
Updates User's Listings

User Interface Mockups



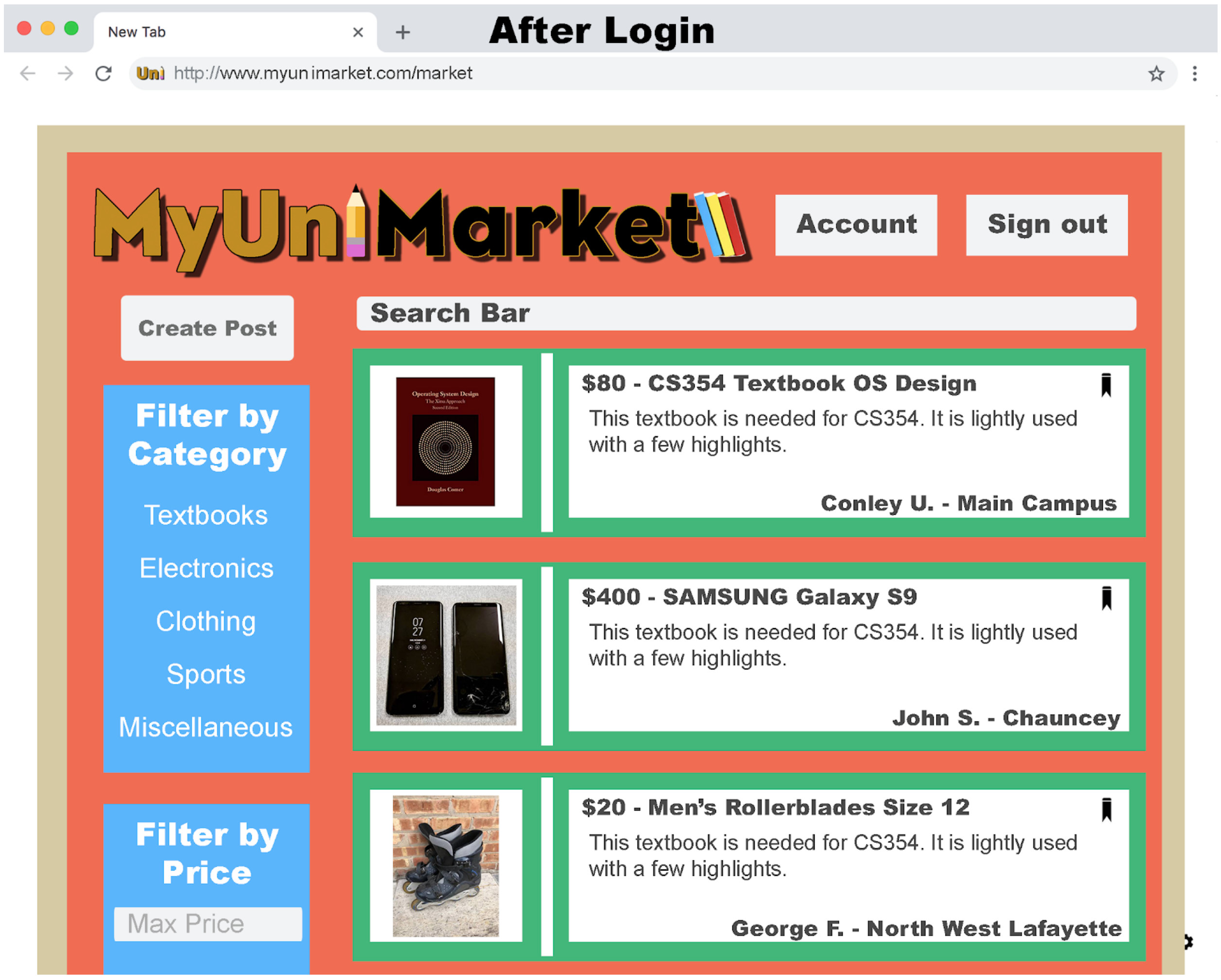
BASIC INFO REGARDING THE WEBSITE.
- Lorem ipsum dolor sit amet, consectetur adipiscing elit.
- Donec pharetra ligula arcu, sollicitudin eleifend est semper eu.
- Fusce malesuada malesuada metus eu laoreet.
- Phasellus aliquet mi eu rhoncus laoreet.

This is the index page which also serves as the sign in. There are fields to input the sign in information. There is a button for signing in, as well as a button to reset an account's password in the event that it was lost. We also have a button for new account registration, leading to the registration page. On the left, there is basic information on how the website operates.



Terms of Service, EULA:
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec pharetra ligula arcu, sollicitudin eleifend est semper eu. Fusce malesuada malesuada metus eu laoreet. Phasellus aliquet mi eu rhoncus laoreet.

This is the Sign-Up page. On the left, it has fields which a new user will fill out in order to create their MyUniMarket account. Under the fields, there is a button that confirms the signing up of the user, granted all the required fields have the appropriate content needed. On the right half of the page, we have included the EULA/TOS that user automatically accepts by clicking the signup button.

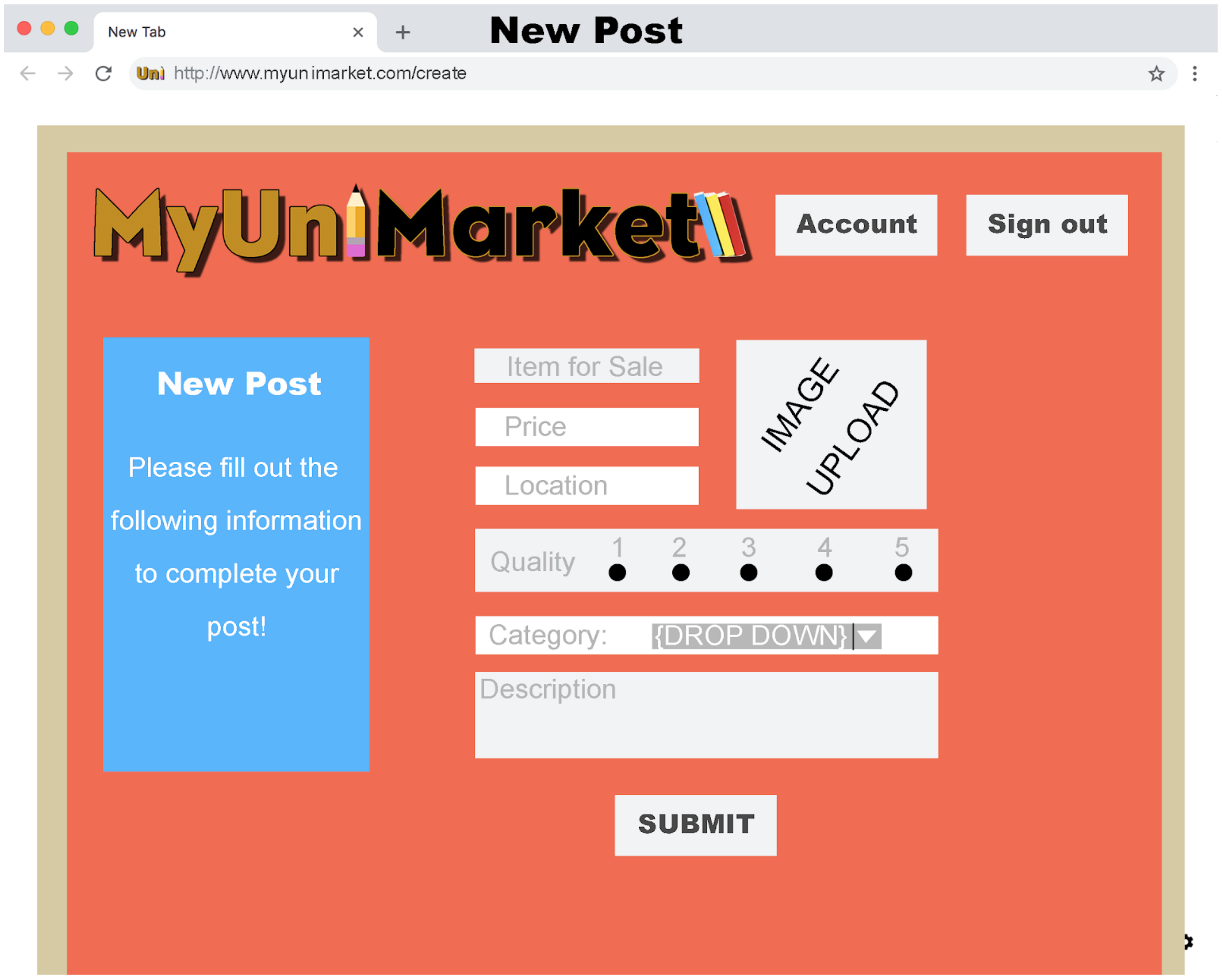


$80 - CS354 Textbook OS Design
This textbook is needed for CS354. It is lightly used with a few highlights.
Conley U. - Main Campus

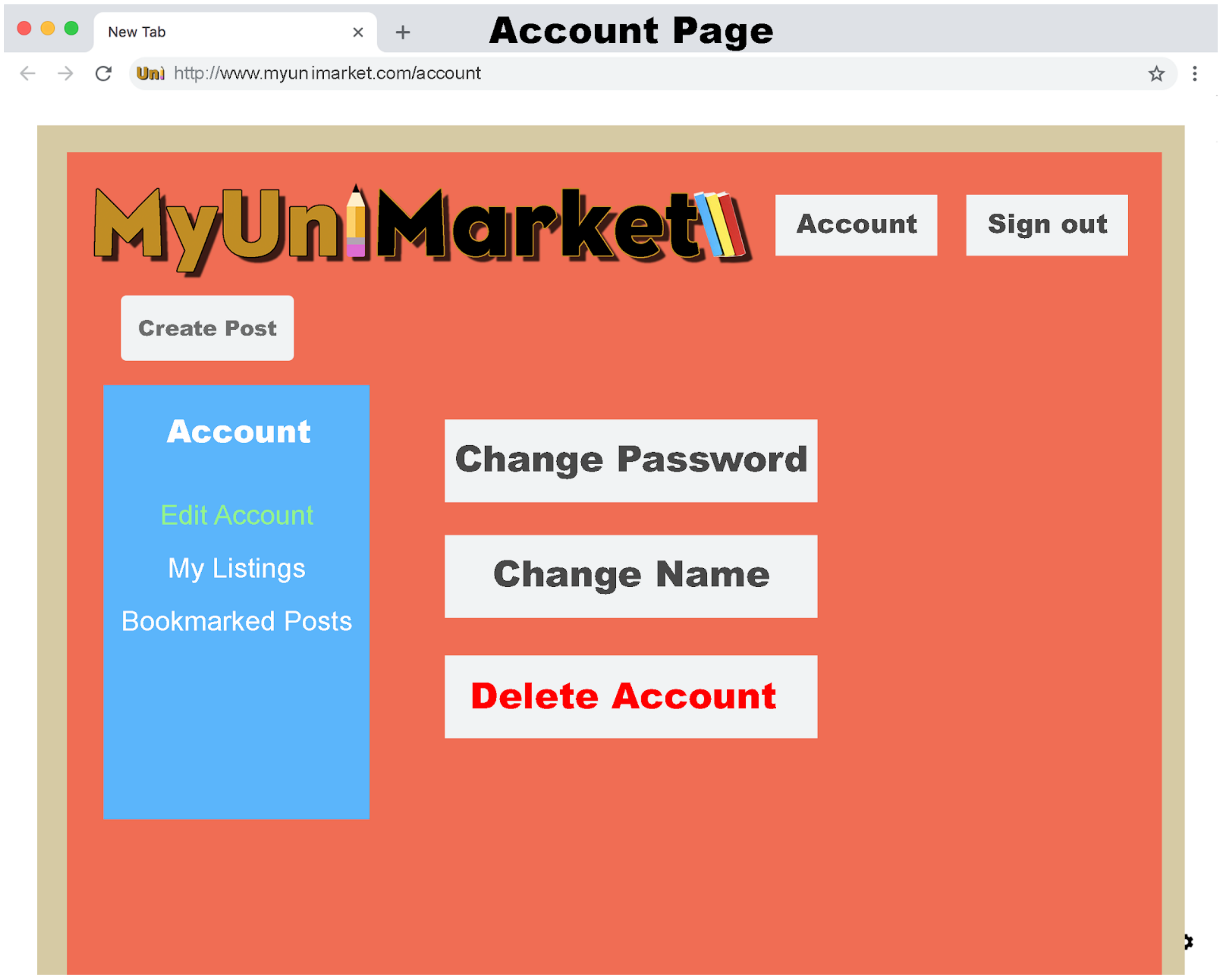
$400 - SAMSUNG Galaxy S9
This textbook is needed for CS354. It is lightly used with a few highlights.
John S. - Chauncey

$20 - Men's Rollerblades Size 12
This textbook is needed for CS354. It is lightly used with a few highlights.
George F. - North West Lafayette

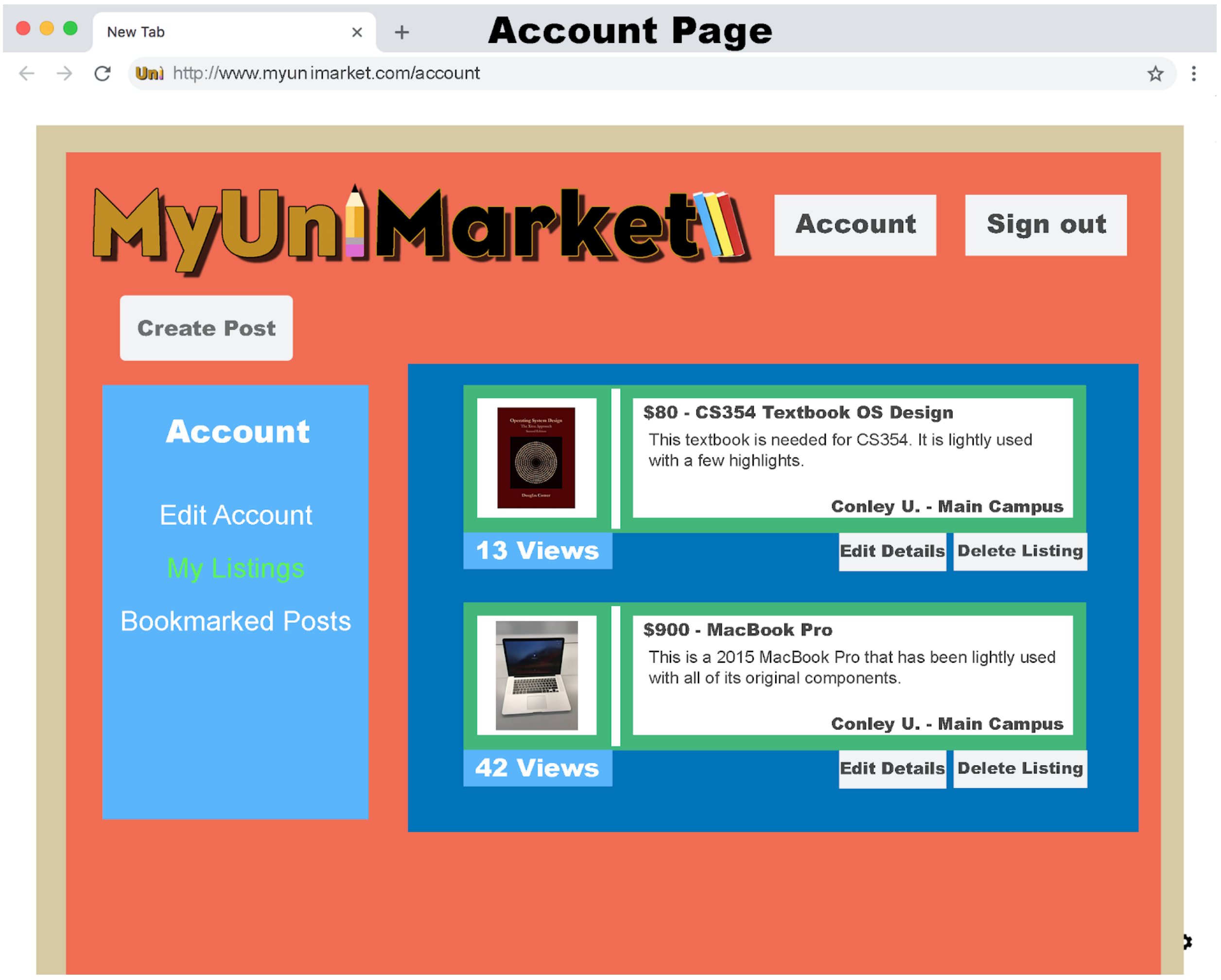
This is the main page a user is presented with after logging in. On the left, there is a panel that contains different categories. Users can quickly filter the main page by selecting a category. Under the category section, there is another panel which includes the "Filter by Price" feature in which a user can set a max price. The central part of this mockup includes example postings, all of which are the most current listings. These listings include basic information, and for more details, the user can click on a post. On the top of the listings, there is a search bar that will filter the list by keywords that a user can choose. The Sign out button signs the user out and returns them to the welcome page. The Account button leads users to their personal account page where they can edit information and review their posts. The Create Post button brings users to the "New Post" page and allows them to add a new listing.



This is the page that appears after the user clicks the Create Post button on any of the other pages. We have the same Account/Sign out buttons with the same functionality. In the middle of the page, there are required fields in order to create a new post. These fields include the name of the product, price, quality, category, description, image, and location. Finally, there is a submit button at the bottom that creates a new post if all the required fields are filled with appropriate information.



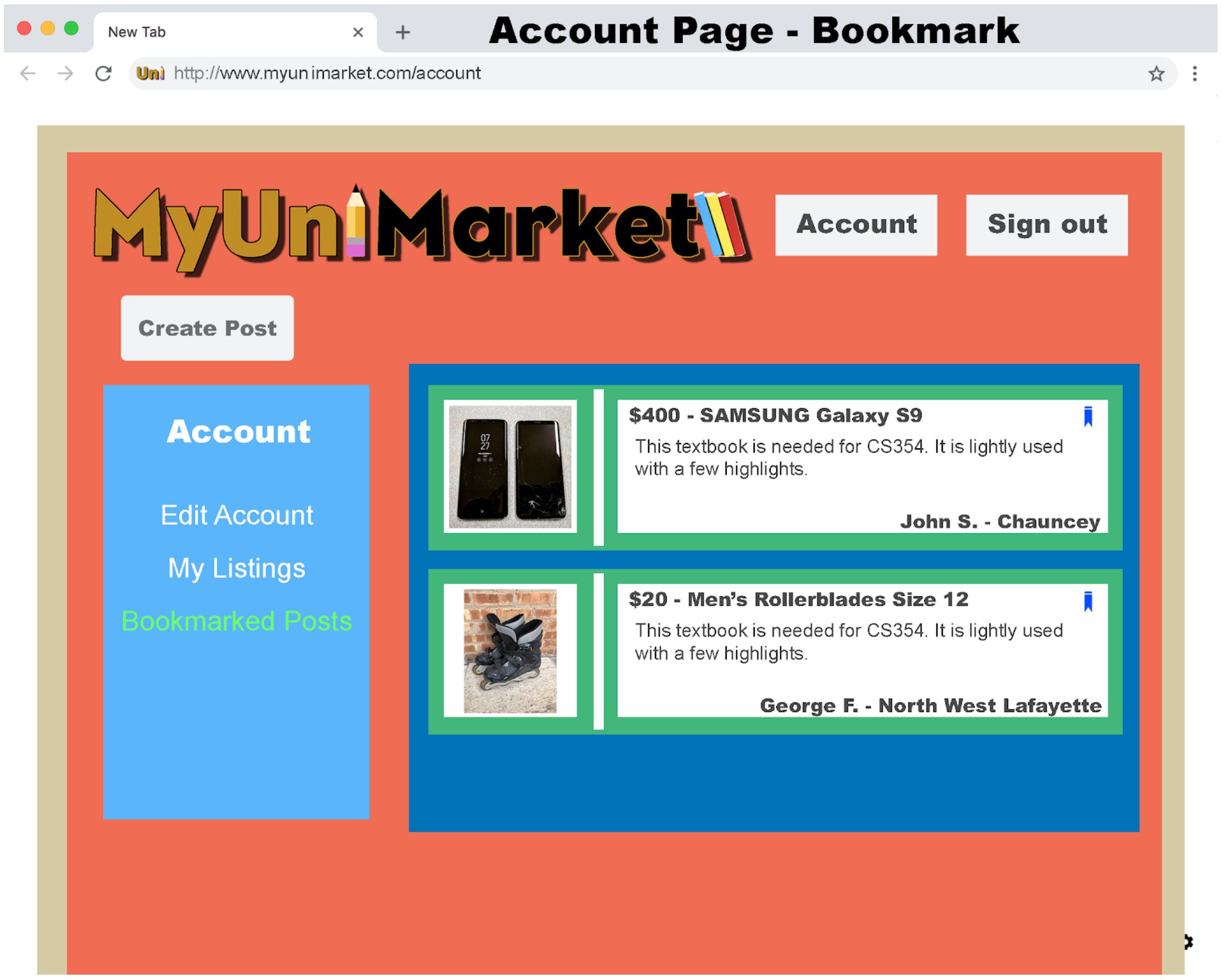
This page is requested if the user clicks on the Account button. On the left, there is a quick navigation menu where the user can select from editing their account, viewing their listings, or viewing their saved posts. Once again, the Account, Sign out, and Create Post buttons have the same functionality as mentioned previously. On this specific part of the Account page, the user can edit their password, change their username, or in the event they no longer need MyUniMarket, delete their account. Deleting an account removes it permanently from the database.



$80 - CS354 Textbook OS Design
This textbook is needed for CS354. It is lightly used with a few highlights.
Conley U. - Main Campus

$900 - MacBook Pro
This is a 2015 MacBook Pro that has been lightly used with all of its original components.
Conley U. - Main Campus

This is the second part to the account page as selected from the navigation menu: My Listings. This page displays the listings the user has created and allows the user to edit the details or delete them. On the top, again, we have the same three buttons with the same functionality. Towards the center, the user can see all the live listings they have created along with the number of views each has received. The Edit Details button allows the user to edit the fields in which they wrote during the creation phase. The Delete Listing lets the user delete their post as a whole whether their transaction is complete or they no longer want to sell their item.



$400 - SAMSUNG Galaxy S9

This textbook is needed for CS354. It is lightly used with a few highlights.

John S. - Chauncey

$20 - Men's Rollerblades Size 12

This textbook is needed for CS354. It is lightly used with a few highlights.

George F. - North West Lafayette

This page is requested when the user selects "Bookmarked Posts" from the account navigation menu. It allows users to view the posts they have saved earlier for convenience. Once again, there are the same three buttons with the same functionality as before. The blue navigation menu has the same functionality mentioned previously as well. In the middle, there are the bookmarked listings that the user has saved by clicking on the bookmark icon on the top right of the posts they desire. The listings have all their original assets including the price, name, brief description, seller's username, and location.