CSC 4330 Group F

Austin Kirk, Evan Kiser, Han Lee, Nicholas Levergne, and Andrew Little

Austin Kirk: MVC Diagram, Homepage Test Case, Homepage Code

Evan Kiser: MySQL Seed Script, Browse Test Case, Browse Code

Han Lee: Checkout Test Case, Checkout Code

Nicholas Levergne: Login, Account Creation, and Listing Test Cases and Code, Sprint 3 Document, Updated Project Backlog

Andrew Little: Profile Page Test Case, Profile Code

https://github.com/nickleverge/CSC4330ProjectGroupF

SPRINT 3 DOCUMENTS

Test Cases with Plans and Results

Account Creation

Test case description: This test case checks to see if the Account Creation functionality works as intended.

Input: Email, password, password typed again, first name, last name.

Condition or function under test: If users can create an account, and if the information they enter is posted to the MySQL database.

Expected Output: User redirected to login screen and a message tells them they are successful.

Output: Users are properly redirected and informed if their account was made or not. This can be verified by checking the accounts table in MySQL Workbench.

Login

Test case description: This test case checks to see if the Login functionality allows users to log in successfully.

Input: Email and password.

Condition or function under test: The user should be able to log in if they have the correct credentials or be informed if they do not. Users should be allowed to the Homepage if they pass the login.

Expected Output: Users should be brought to the Homepage on button press.

Output: Users are correctly brought to the Homepage or sent back to the Login page with a message telling them why.

Post Listing

Test case description: This test case checks to see if listings properly post to the server.

Input: Listing title, price, description, categories (check as many as apply), other (user custom category input), pictures (4).

Condition or function under test: The user should be able to post a listing and fill out as much information relating to it as they can. If the user enters bad or incomplete information, they should be notified of such, and the listing should not post.

Expected Output: The listing should be uploaded to the server if the information is filled out correctly, if not the user should be told why, and it should not post.

Output: Listings post appropriately with all information, except for a MySQL file size exception for pictures larger than 4 megabytes. Users are also properly redirected if the information is malformed or missing.

Home Page Display Test Case

Test case description: Home Page Listing Display

Input: Listing Title and Photos

Condition or function under test: To display the latest top 10 listings created on the website with their title, photo, and link to listing.

Expected Output: The 10 newest listings are displayed once the homepage has fully loaded in a grid and centered on the front of the homepage.

Output: Output unable to be tested currently as feature has not been fully completed for thorough testing purposes.

Browse Page

Test case description: Browse Page full listing

Input: Filter by price and tags

Condition or function under test: run a query search for listing that have the associated price and tags

Expected Output: a full list of all the listings

Output: Unable to be tested currently as feature has not been fully completed.

Checkout

Test case description: Checkout Page

Input: Payment information and items in cart

Condition or function under test: To view current items in cart, input payment info, and successfully purchase selected items.

Expected Output: A “success” message is displayed. Items from cart disappear and are added to orders.

Output: Not built yet, can’t properly test output

Profile Page

Input: Name

Condition or function under test: To view lister’s name

Expected Output: View the users’ name.

Output: Unable to be tested currently as feature has not been fully completed.

Profile Page: Listings

Test case description: Profile Page: View user’s listings

Input: User’s listings

Condition or function under test: To view the user’s listings

Expected Output: View all the user’s listings

Output: Unable to be tested currently as feature has not been fully completed

Profile Page: Edit Listing

Test case description: Profile Page: Edit user’s listing

Input: User’s listings

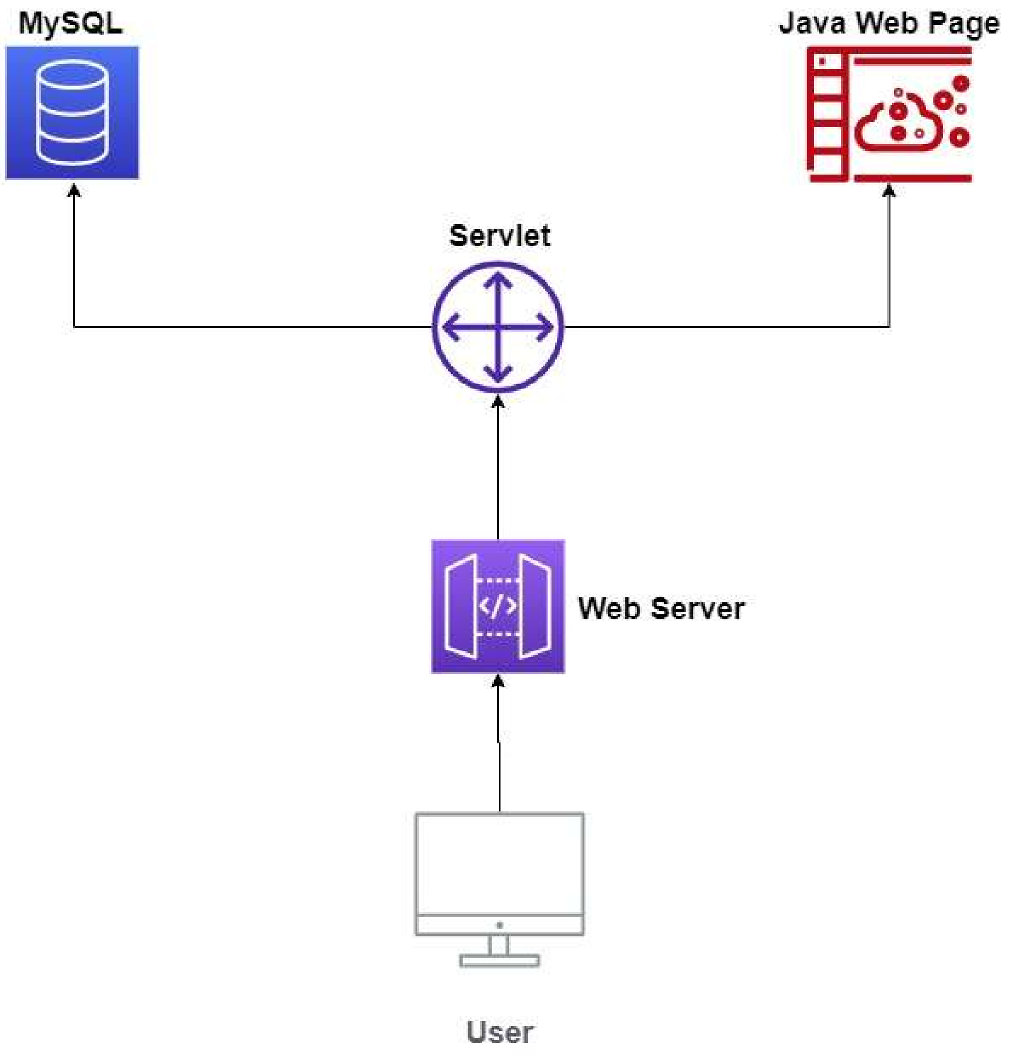
Condition or function under test: To edit the user’s listing

Expected Output: Successfully edited listing

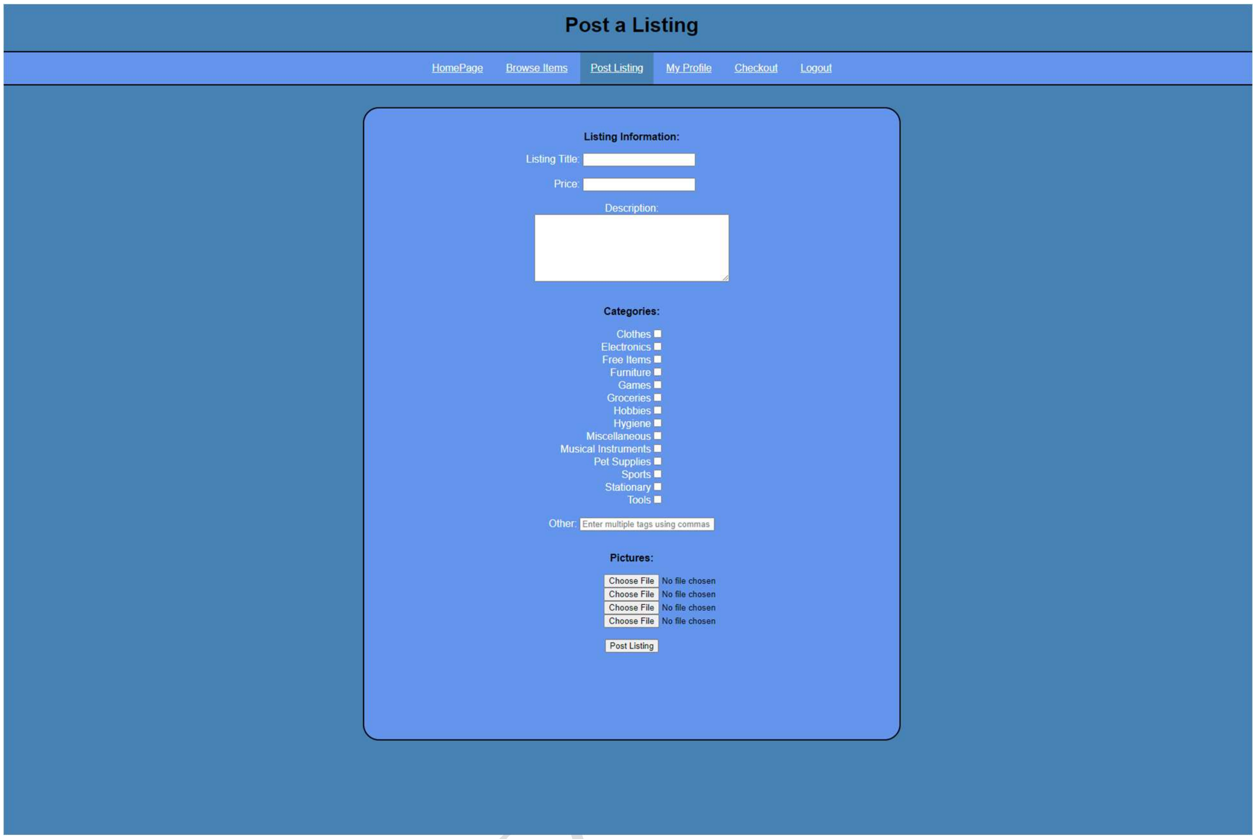
Output: Unable to be tested currently as feature has not been fully completed

New Design Document

Model View Controller Design Document



New Web Pages



Listing Information:

Listing Title:

Price:

Description:

Categories:

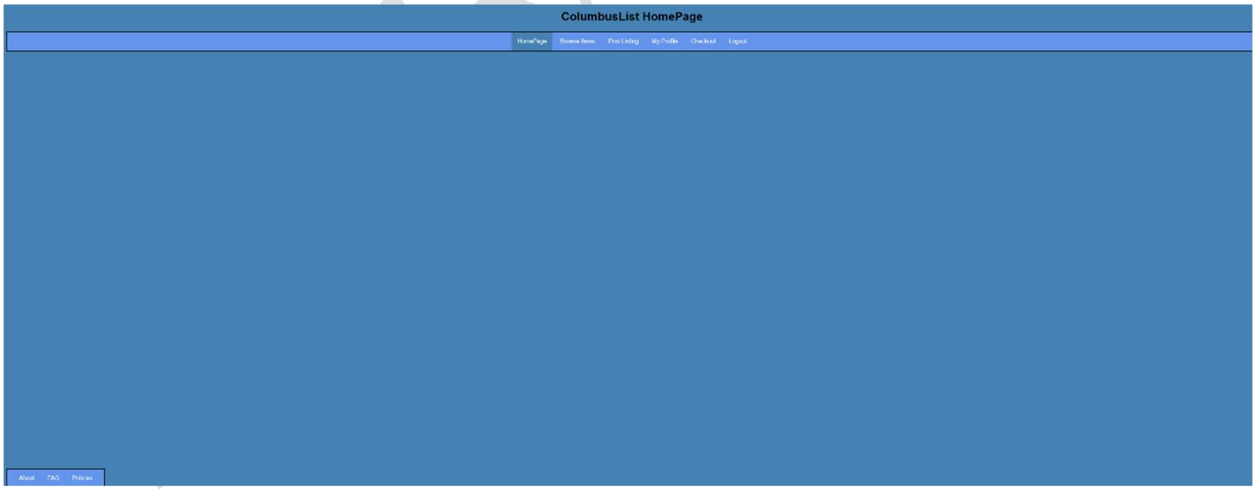
Clothes □
Electronics □
Free Items □
Furniture □
Games □
Groceries □
Hobbies □
Hygiene □
Miscellaneous □
Musical Instruments □
Pet Supplies □
Sports □
Stationary □
Tools □

Other: Enter multiple tags using commas

Pictures:

Choose File No file chosen
Choose File No file chosen
Choose File No file chosen
Choose File No file chosen

Post Listing



SPRINT 2 DOCUMENTS

Feature Descriptions Using User Stories

Description

1. Login
    a. As a user, after I login I want to be redirected to the homepage
2. View Home Page
    a. As a user, I want to be able open the homepage so that I can view, search, and filter through listings.
        i. View
            1. I want to be able to view listing so that I can see the title, price and tags before clicking to get the rest of the details.
        ii. Search
            1. I want to be able to search what is for sale so I can find exactly what I need.
        iii. Filter
            1. I want to be able to filter by price so that I can find listings in my price range.
3. View Wishlist
    a. As a user, I want to be able view my Wishlist so I can add listings.
        i. Add
            1. I want to add listings so I can watch for price changes
4. View Profile
    a. As a user, I want to be able to access my profile page so that I can edit or remove my listings
        i. Edit Listings

1. I want to be able to edit all the details in my listings I created.

ii. Remove Listing

1. I want to be able to remove listings that I created that have been purchased.

5. Create New Listing

a. As a user, I want to create a new listing that contains a title, description, images, price, contact information, and tags.

i. Title

1. I want to be able to add a title to concisely give the buyer an idea of what the item is.

ii. Description

1. I want to be able to add a description to the listing so I can answer common questions potential buyers may have about the product.

iii. Images

1. I want to be able to upload images of the product/service I want to sell to give the buyer a better idea of what they're getting.

iv. Price

1. I want to be able to add a price so that I can only attract buyers that can afford my listing.

v. Contact information

1. I want to be able to add contact information to my listings so that potential buyers can contact me to buy the product.

vi. Tags

1. I want to be able to select a pre-defined categories for easier search abilities.

6. Notifications

    a. As a user, I want to get notifications for transaction requests and Wishlist updates

        i. Transaction Request

            1. I want to get notifications for when someone wants to arrange a transaction.

        ii. Wishlist

            1. I want to get a notification if the price of an item on my Wishlist is changed so that I can decide if I want the product/service.

7. Payment Action

    a. As a user, I want to be able to securely pay a seller for a product/service so that I can safely pay online.

8. Report Action

    a. As a user, I want to have access to a report button to help prevent suspicious account activity.

9. Logout Action

    a. As a user, I want to able to logout of my profile so that no one can access my account when I am not at my computer.

Constraints

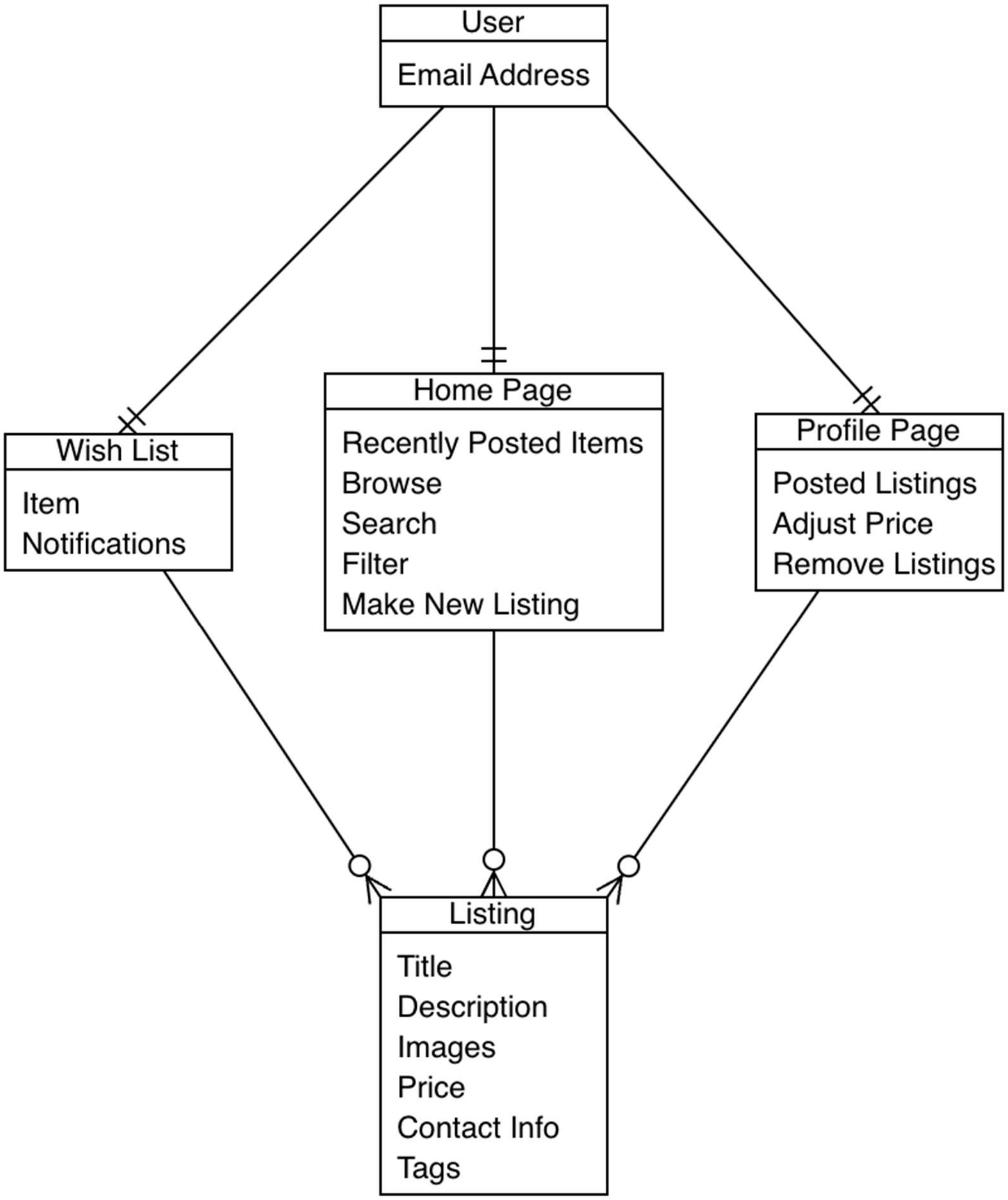
1. Sign up

    a. Users can only gain access to the website after they sign up and that their email is verified as a school email.

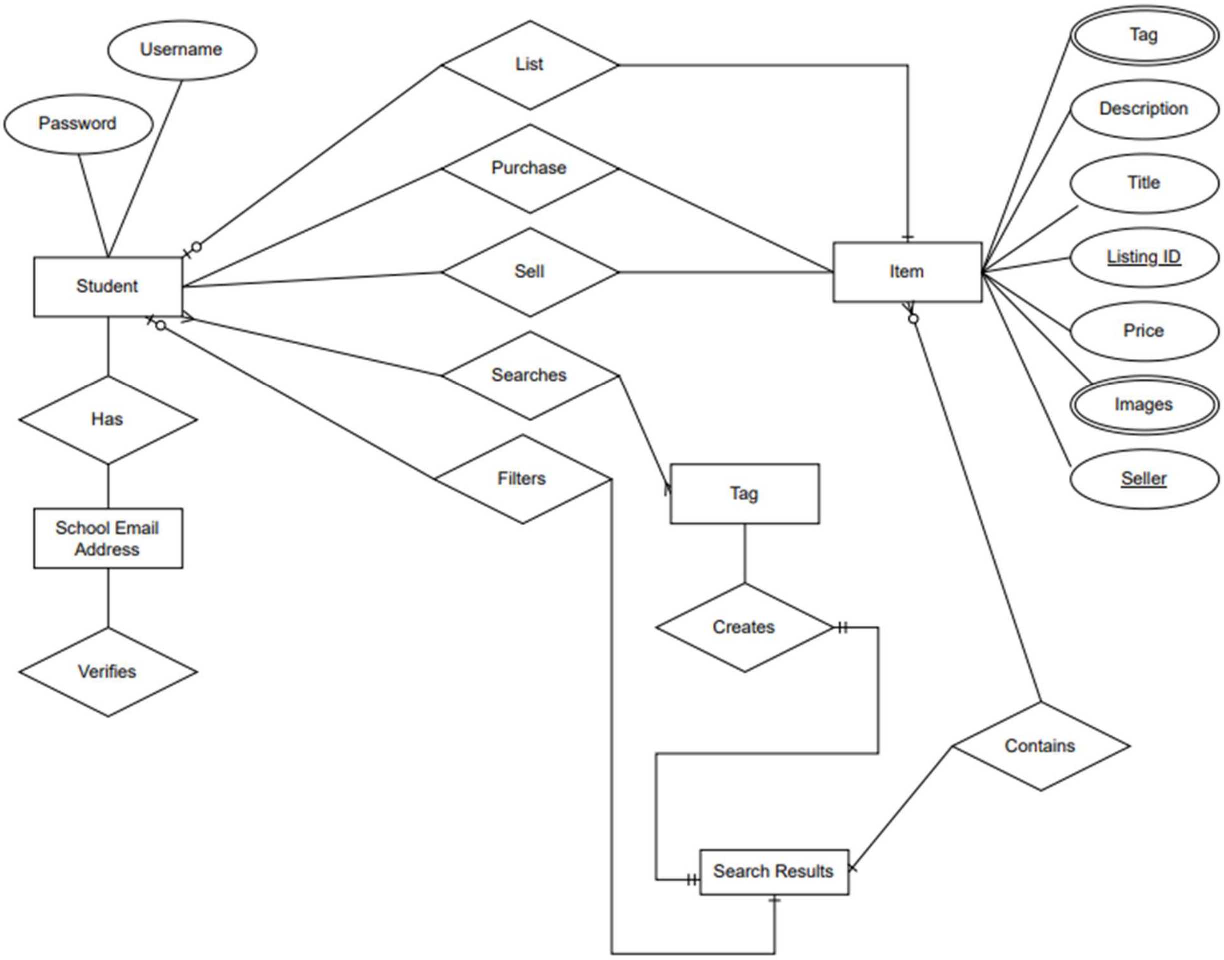
Comments

1. Based on Brock's, Adam's, Emily's, and Brandon's scenarios

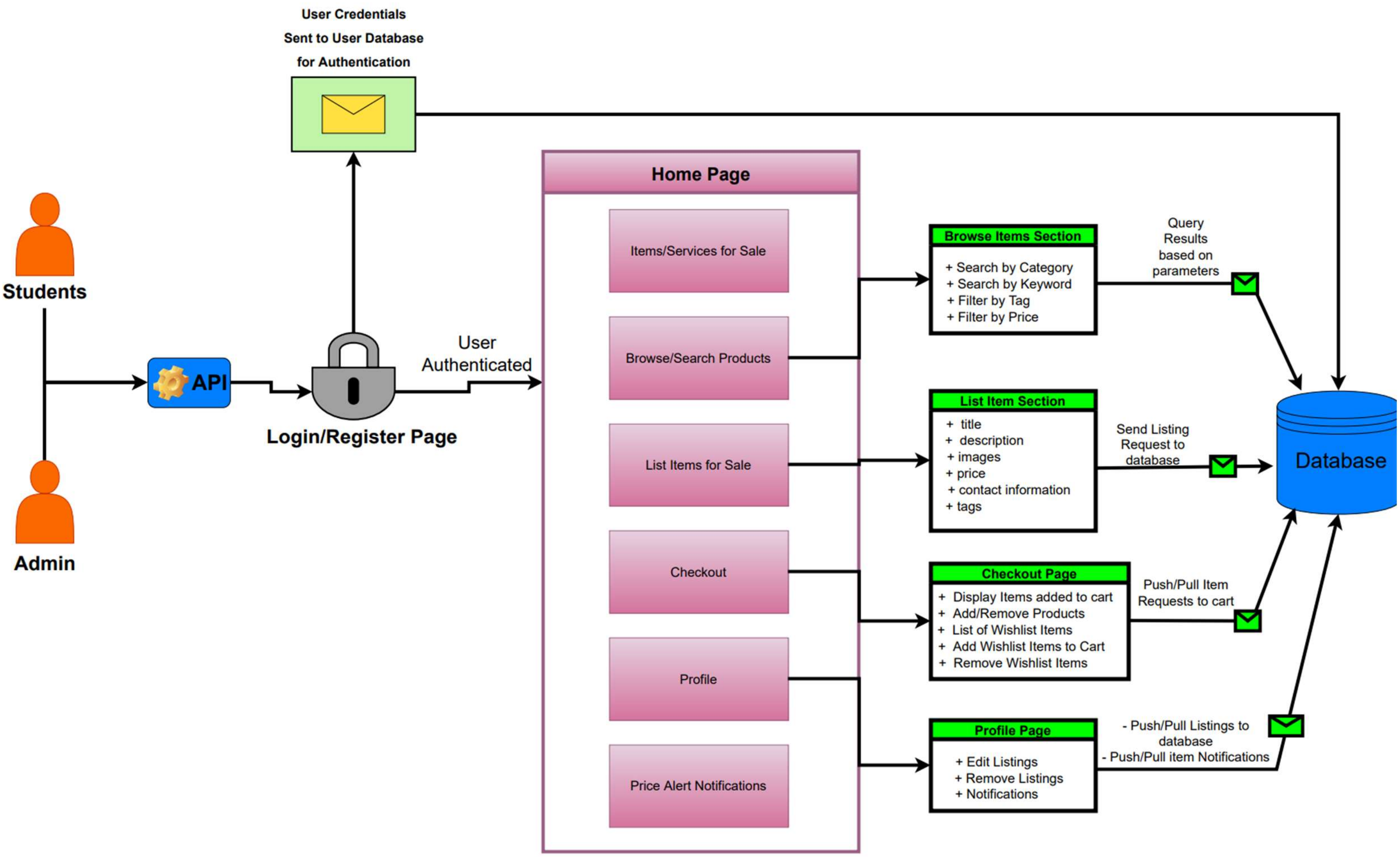
Design Documents



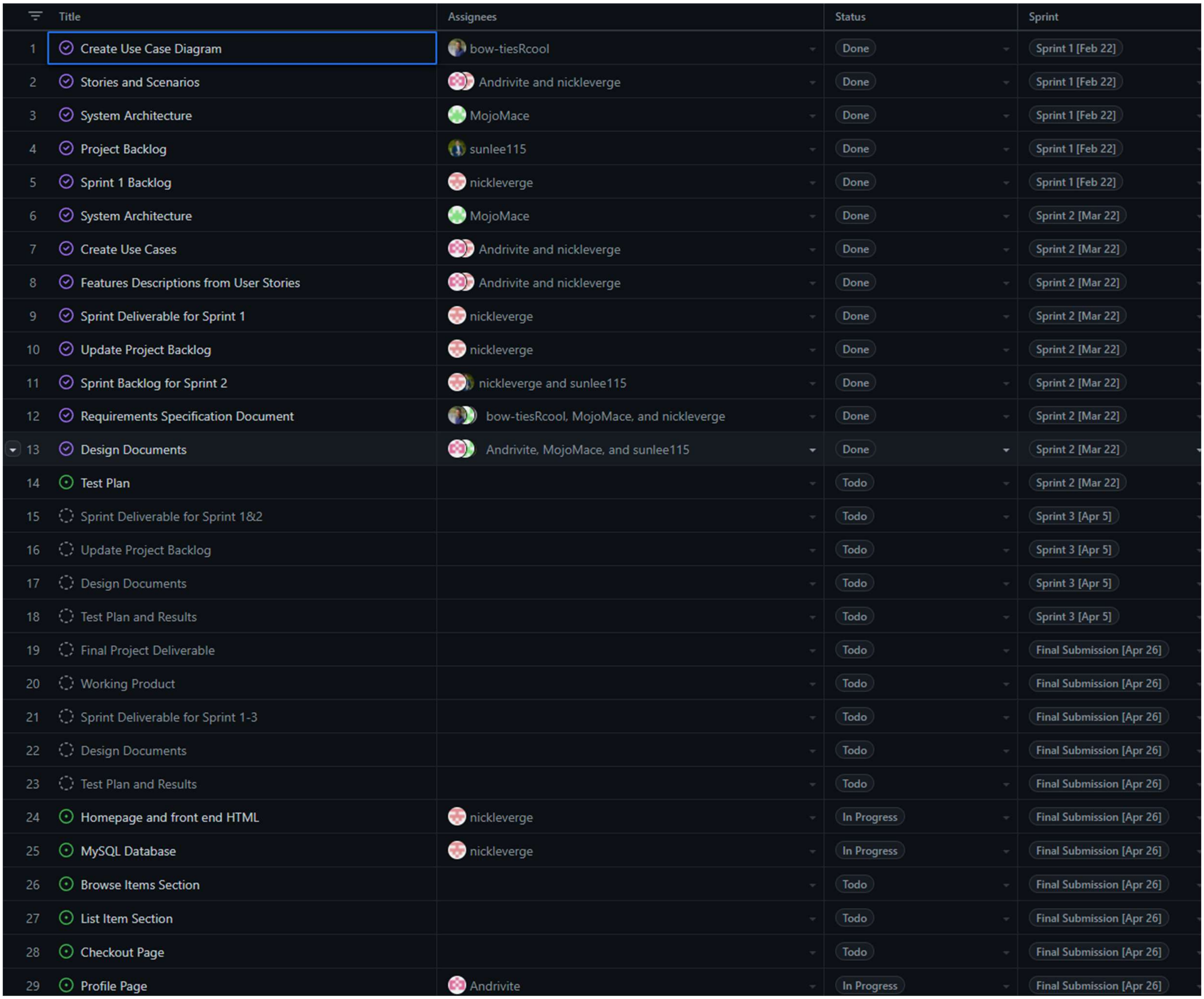
ER Diagram


Username
Password
Student
Has
School Email Address
Verifies
List
Purchase
Sell
Searches
Filters
Tag
Creates
Search Results
Contains
Item
Tag
Description
Title
Listing ID
Price
Images
Seller

Updated System Architecture

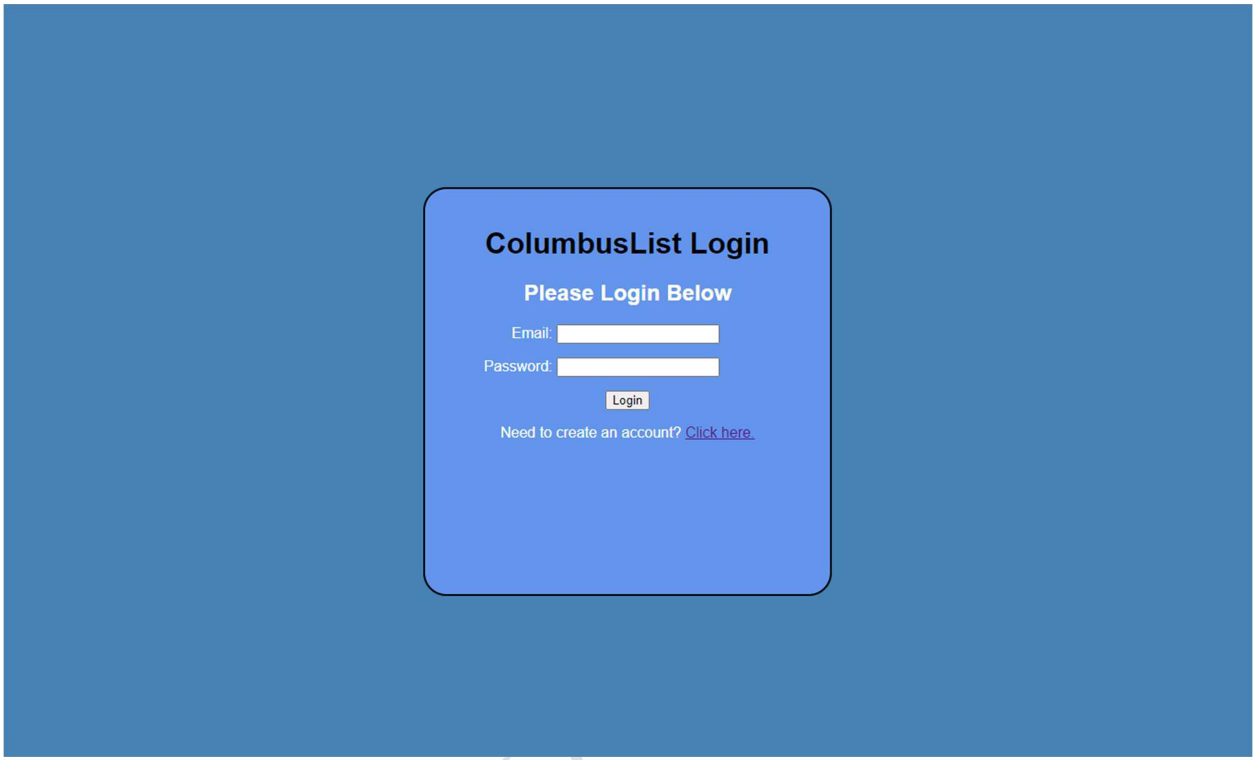


Updated Project and Sprint Backlog



	Title	Assignees	Status	Sprint
1	Create Use Case Diagram	bow-tiesRcool	Done	Sprint 1 [Feb 22]
2	Stories and Scenarios	Andrivite and nickleverge	Done	Sprint 1 [Feb 22]
3	System Architecture	MojoMace	Done	Sprint 1 [Feb 22]
4	Project Backlog	sunlee115	Done	Sprint 1 [Feb 22]
5	Sprint 1 Backlog	nickleverge	Done	Sprint 1 [Feb 22]
6	System Architecture	MojoMace	Done	Sprint 2 [Mar 22]
7	Create Use Cases	Andrivite and nickleverge	Done	Sprint 2 [Mar 22]
8	Features Descriptions from User Stories	Andrivite and nickleverge	Done	Sprint 2 [Mar 22]
9	Sprint Deliverable for Sprint 1	nickleverge	Done	Sprint 2 [Mar 22]
10	Update Project Backlog	nickleverge	Done	Sprint 2 [Mar 22]
11	Sprint Backlog for Sprint 2	nickleverge and sunlee115	Done	Sprint 2 [Mar 22]
12	Requirements Specification Document	bow-tiesRcool, MojoMace, and nickleverge	Done	Sprint 2 [Mar 22]
13	Design Documents	Andrivite, MojoMace, and sunlee115	Done	Sprint 2 [Mar 22]
14	Test Plan		Todo	Sprint 2 [Mar 22]
15	Sprint Deliverable for Sprint 1&2		Todo	Sprint 3 [Apr 5]
16	Update Project Backlog		Todo	Sprint 3 [Apr 5]
17	Design Documents		Todo	Sprint 3 [Apr 5]
18	Test Plan and Results		Todo	Sprint 3 [Apr 5]
19	Final Project Deliverable		Todo	Final Submission [Apr 26]
20	Working Product		Todo	Final Submission [Apr 26]
21	Sprint Deliverable for Sprint 1-3		Todo	Final Submission [Apr 26]
22	Design Documents		Todo	Final Submission [Apr 26]
23	Test Plan and Results		Todo	Final Submission [Apr 26]
24	Homepage and front end HTML	nickleverge	In Progress	Final Submission [Apr 26]
25	MySQL Database	nickleverge	In Progress	Final Submission [Apr 26]
26	Browse Items Section		Todo	Final Submission [Apr 26]
27	List Item Section		Todo	Final Submission [Apr 26]
28	Checkout Page		Todo	Final Submission [Apr 26]
29	Profile Page	Andrivite	In Progress	Final Submission [Apr 26]

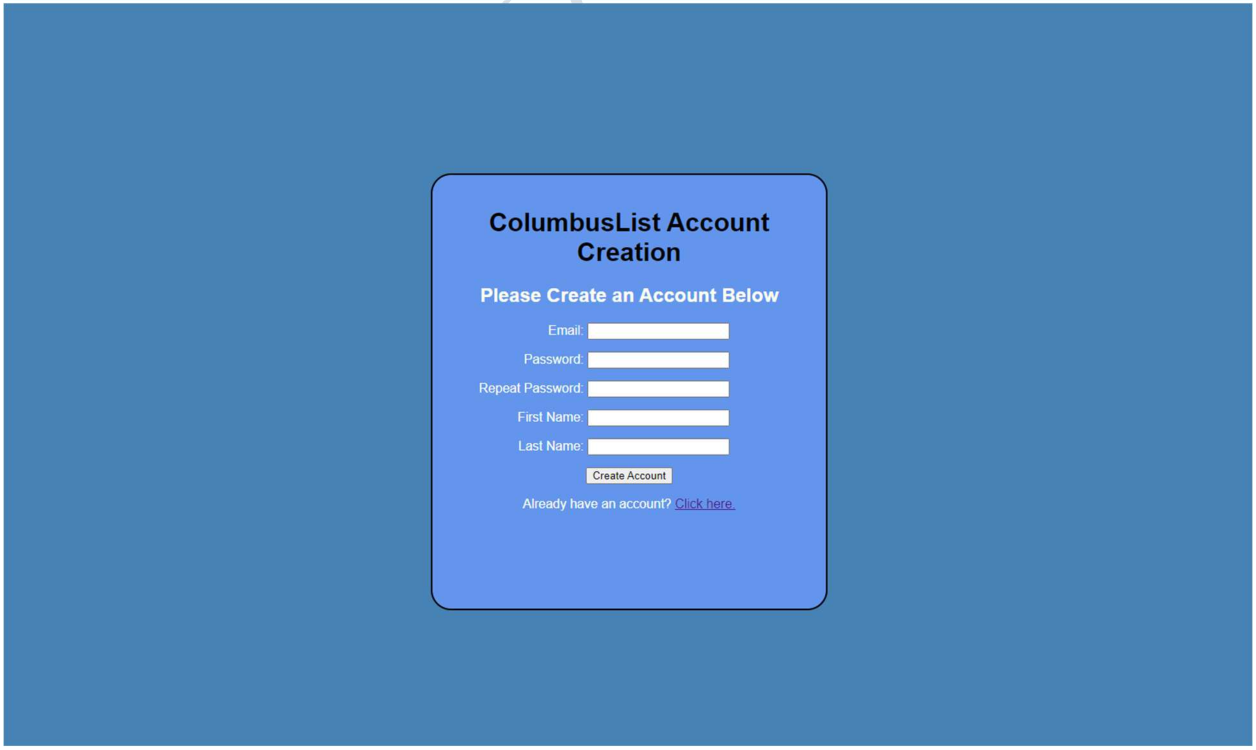
Login, Account Creation, and Homepage



ColumbusList Login

Please Login Below

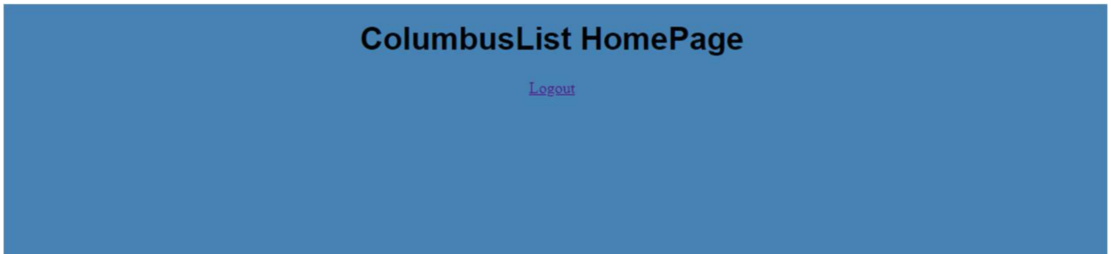
Need to create an account? Click here.



ColumbusList Account Creation

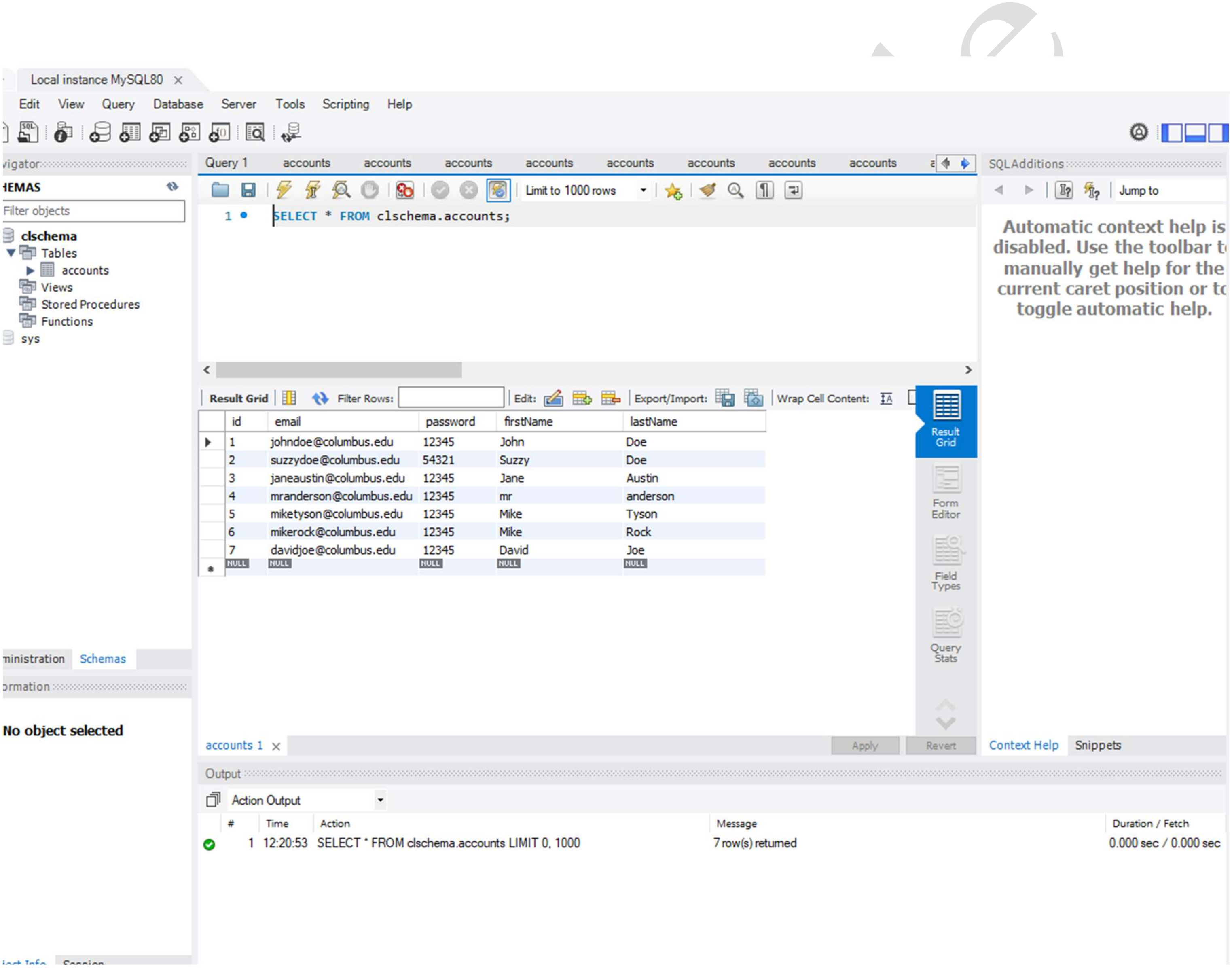
Please Create an Account Below

Already have an account? Click here.



ColumbusList HomePage

MySQL Database


Local instance MySQL80 ×
Edit View Query Database Server Tools Scripting Help
vigator
HEMAS
Filter objects
clschema
Tables
accounts
Views
Stored Procedures
Functions
sys
ministration Schemas
ormation
No object selected
Query 1 accounts accounts accounts accounts accounts accounts accounts accounts
Limit to 1000 rows
Result Grid Filter Rows: Edit: Export/Import: Wrap Cell Content:
Result Grid
Form Editor
Field Types
Query Stats
accounts 1 ×
Apply Revert
SQLAdditions
Jump to
Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.
Context Help Snippets
Output
Action Output

```
SELECT * FROM clschema.accounts;
```

id	email	password	firstName	lastName
1	johndoe@columbus.edu	12345	John	Doe
2	suzzydoe@columbus.edu	54321	Suzzy	Doe
3	janeaustin@columbus.edu	12345	Jane	Austin
4	mranderson@columbus.edu	12345	mr	anderson
5	miketyson@columbus.edu	12345	Mike	Tyson
6	mikerock@columbus.edu	12345	Mike	Rock
7	davidjoe@columbus.edu	12345	David	Joe
NULL	NULL	NULL	NULL	NULL

#	Time	Action	Message	Duration / Fetch
1	12:20:53	SELECT * FROM clschema.accounts LIMIT 0, 1000	7 row(s) returned	0.000 sec / 0.000 sec

SPRINT 1 DOCUMENTS

Personas and Scenarios

Brock, a graduating student at Columbus University

Brock, age 23, has just finished his degree in Chemical Engineering at Columbus University. He has a girlfriend and no children. He is a local from the area around Columbus University, but he is not opposed to moving out of the state to further his career. He graduated over the summer with honors and is looking forward to entering the industry professionally. He has even received a great offer, but it requires him to move to a neighboring state. Fortunately, his girlfriend has also found career opportunities in the same area.

Brock already has a new apartment already picked out for himself and his significant other, and he feels that it is time he parts ways with his old college dorm furniture to make the move easier. While he has sold some items over Facebook marketplace before, he wants his old furniture to support the next class of Columbus University students. His girlfriend suggests a new online marketplace website, ColumbusList. Users of the site must verify that they are students with their school email address. Brock makes a mental note of the site for later.

Adam, a freshman at Columbus University

Adam, age 18, is new student at Columbus University. He is from out of state and is acclimating to his new environment. He graduated high school cum laude and is now pursuing a degree in business administration. He is single, but two of his high school friends are also attending Columbus University with him. One of his friends is also his roommate.

Adam and his friends all join the ultimate frisbee club. Adam is also looking at joining the Club of Business Associates. He wants to quickly create a network of friends at his new college to enrich his college experience with. He also needs his and his roommate's dorm to be well furnished so they can invite others to hang out, but they are tight on money and as they are both new to the area, they aren't sure where they can find affordable furniture at. Adam is Googling online marketplaces when he comes across a website called ColumbusList.

Emily, a Sophomore at Columbus University

Emily, age 20, is a sophomore at Columbus University pursuing a degree in Physics. Emily is always on the move since she works many hours to pay for her school. Emily's old laptop from high school is no longer working, so she needs a new one. Emily looks at many surrounding stores but sees that the laptops with the specifications that she needs for classes are out of her budget of $500. She decides to use ColumbusList, a website she had sold her freshman year textbook's on, to search for a used laptop.

Brandon, a Senior at Columbus University

Brandon is in his last Semester at Columbus University. He has some extra time on the weekends since he isn’t taking as many classes. Brandon and his fraternity brothers get an idea to start a fundraiser for a food drive in his city. Brandon decides to advertise his service on ColumbusList.

Brock's scenario

Brock decides it's past time he sells his old dorm furniture. He searches and finds the website his girlfriend recommended, ColumbusList.com. He is greeted by a homepage that prompts him to either log in or signup. He clicks on "signup", which brings him to a sign-up form. He reads the page instructions and enters in his school email with a new password. He is then prompted to log in with his newly created credentials and does so.

Now at the home page, he finds and clicks the "List Items for Sale" button. He is brought to a form that requires him to enter a title, description, image, price, contact information, and tags. He enters in *"Used Bedframe"*, *"This is a twin sized bedframe, brought new and used for three years.",* he takes a picture of the bedframe and uploads it, he then puts $100 dollars for the price, adds his email, and adds the tag *"Dorm Furniture"*.

Brock then similarly enters in three more items of furniture. After half an hour, he receives an email from a buyer named Adam about to of the pieces of furniture. Brock really wants to sell the other two pieces of furniture as well, so he decides to lower the price from his personal profile page. The same buyer reaches out to Adam for those two pieces shortly after. He and Adam arrange the pickup of the furniture, and Adam pays Brock securely over ColumbusList. Brock then removes the four listings he posted and logs out.

Adam's scenario

Adam decides to begin his search for dorm furniture on ColumbusList.com. He searches and finds the website and is prompted to either log in or sign up. He clicks sign up, skims the sign-up form, and quickly enters his personal email and a password. He is then told that his sign up failed because his provided email is not a @columbus.edu email. Perturbed at his failing to read instructions, Adam uses his school email instead, and successfully creates an account.

Adam then logs in and is brought to the homepage. He finds a search bar and enters in "Dorm *Furniture"*. Several listings for furniture are brought up along with images of them. He clicks on two that are in his price range, finds that they are listed by a seller named Brock, and emails Brock about his interest in the furniture. There are also two other pieces of furniture that Adam wants but are out of his price range, so he puts them on his wish list. He is shortly after notified that the prices of the items on his wish list have changed. He notes that they are also sold by Brock, and so he emails Brock again. They arrange a pickup date, and Adam pays Brock using his credit card over ColumbusList.

Emily's scenario

Emily begins looking for a budget laptop on ColumbusList.com. She types the URL into her browser and is greeted by the homepage, where Emily clicks on the "login" button. Emily enters her school email and her password. She clicks on "login" and a new page begins loading.

Sarah is now greeted with the homepage. She clicks on the search bar and enters "Laptop". Sarah is greeted with many laptops for sale. Many of the laptops are out of Sarah's $500 budget, so she clicks on the filter by price button. Sarah then enters a minimum price of 300 and a maximum price of 500 and clicks on the filter button. Now all the laptops on Sarah's page are within the price range of $300 to $500. Sarah browses through the laptop until she finds a portable laptop with sufficient specifications that a Columbus University senior, Jack, is selling. She arranges a pickup and pays Jack using cash.

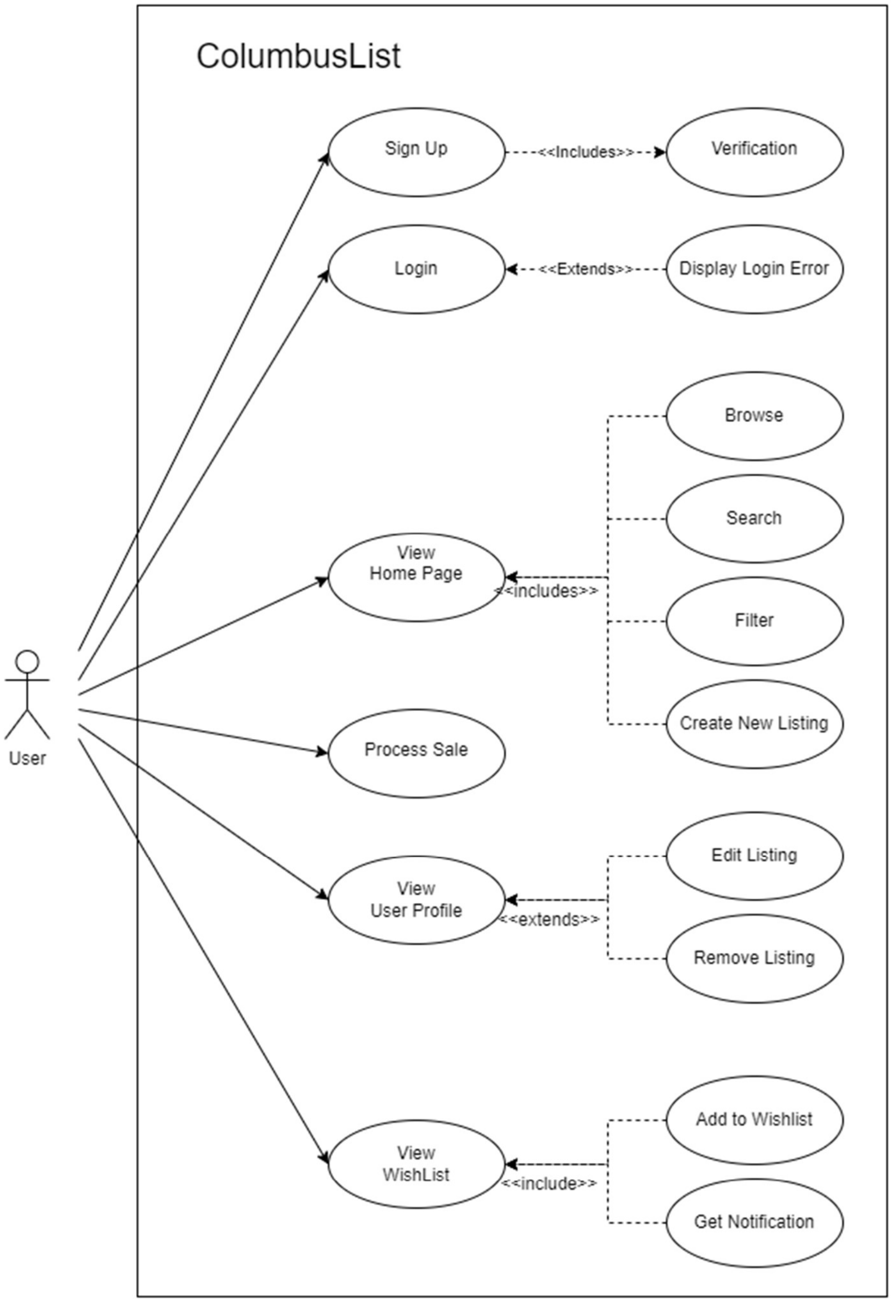
Brandon's scenario

Brandon decides it's time to start advertising his car washing services. He searches and finds the website ColumbusList.com. He is greeted by a homepage that prompts him to either log in or signup. He clicks on "signup", which brings him to a sign-up form. He reads the page instructions and enters in his school email with a new password. He is then prompted to log in with his newly created credentials and does so.

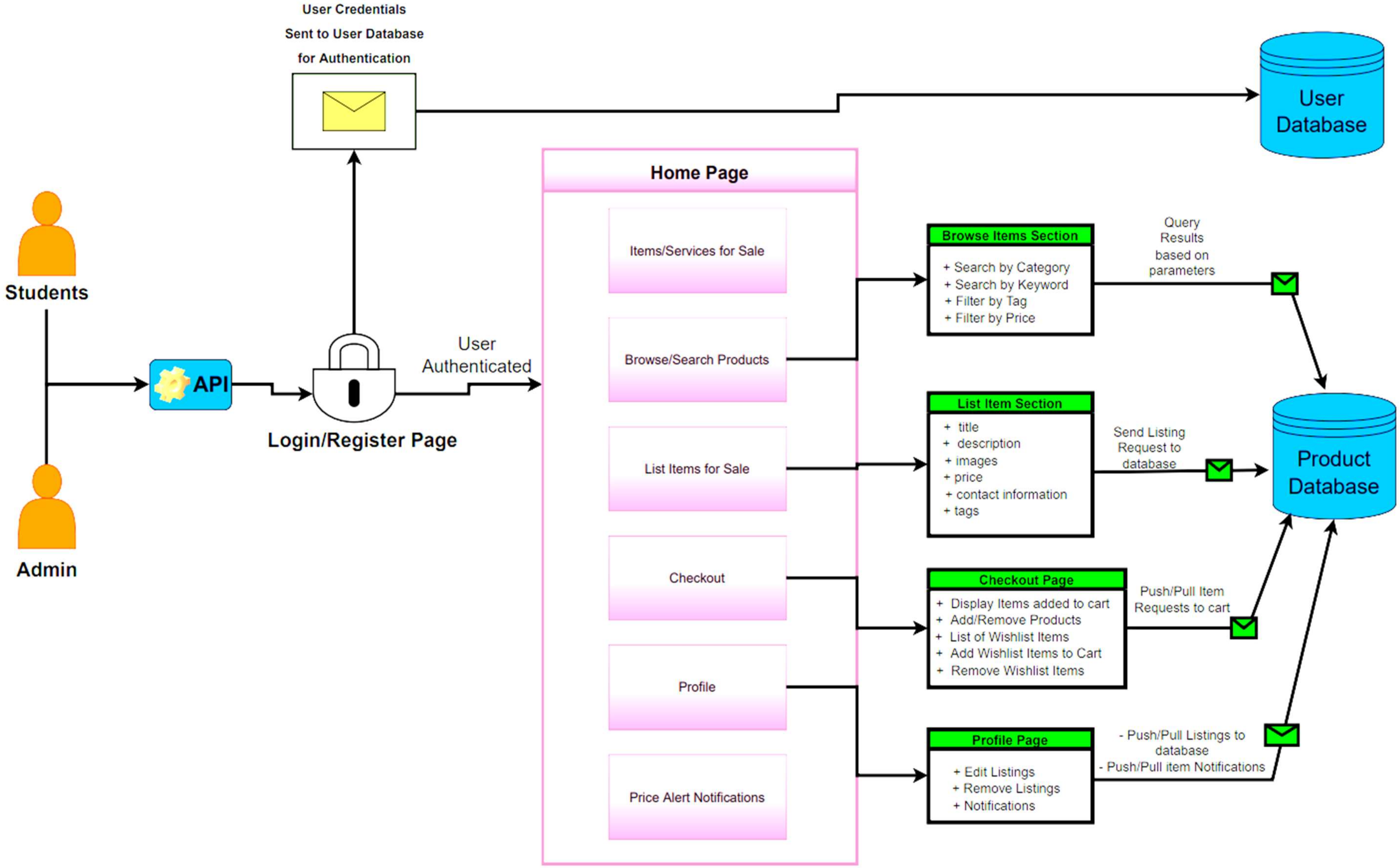
Now at the home page, he finds and clicks the "List Service for Sale" button. He is brought to a form that requires him to enter a title, description, image, price, contact information, and tags. He enters in *"Car Wash for The Weekend – Food Drive"*, *"Car wash from 10am-3pm every weekend at Greek Row, all proceeds go to the local Food Bank",* he adds a picture of his friends washing a car, he then puts $10 dollars for the price, adds his email, and adds the tag *"Car Wash"*.

Brandon gets questions about the food drive from multiple people, and a few others volunteer to help.

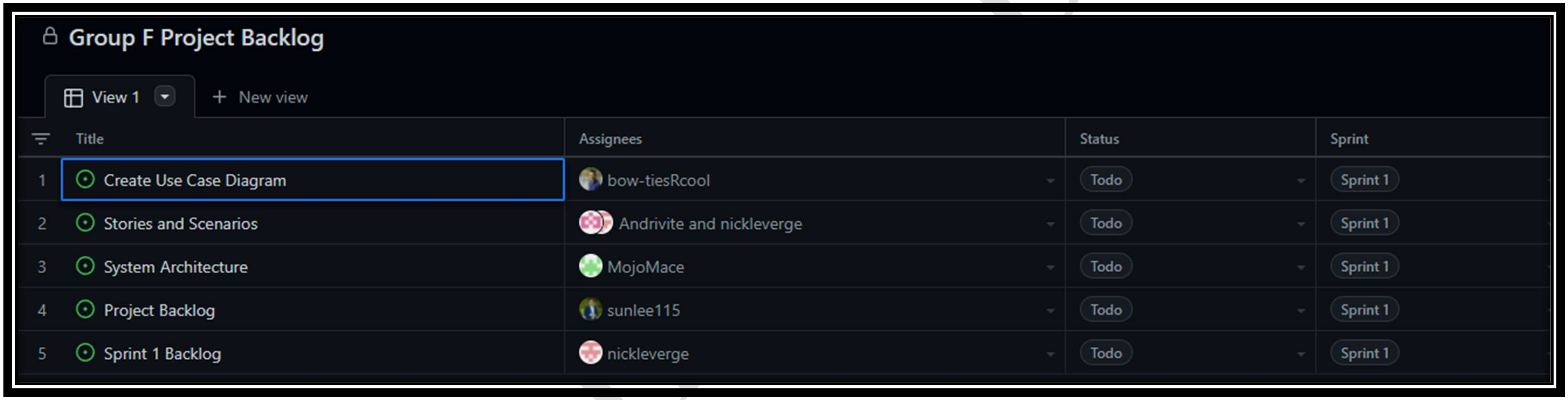
Use Case Diagram



System Architecture



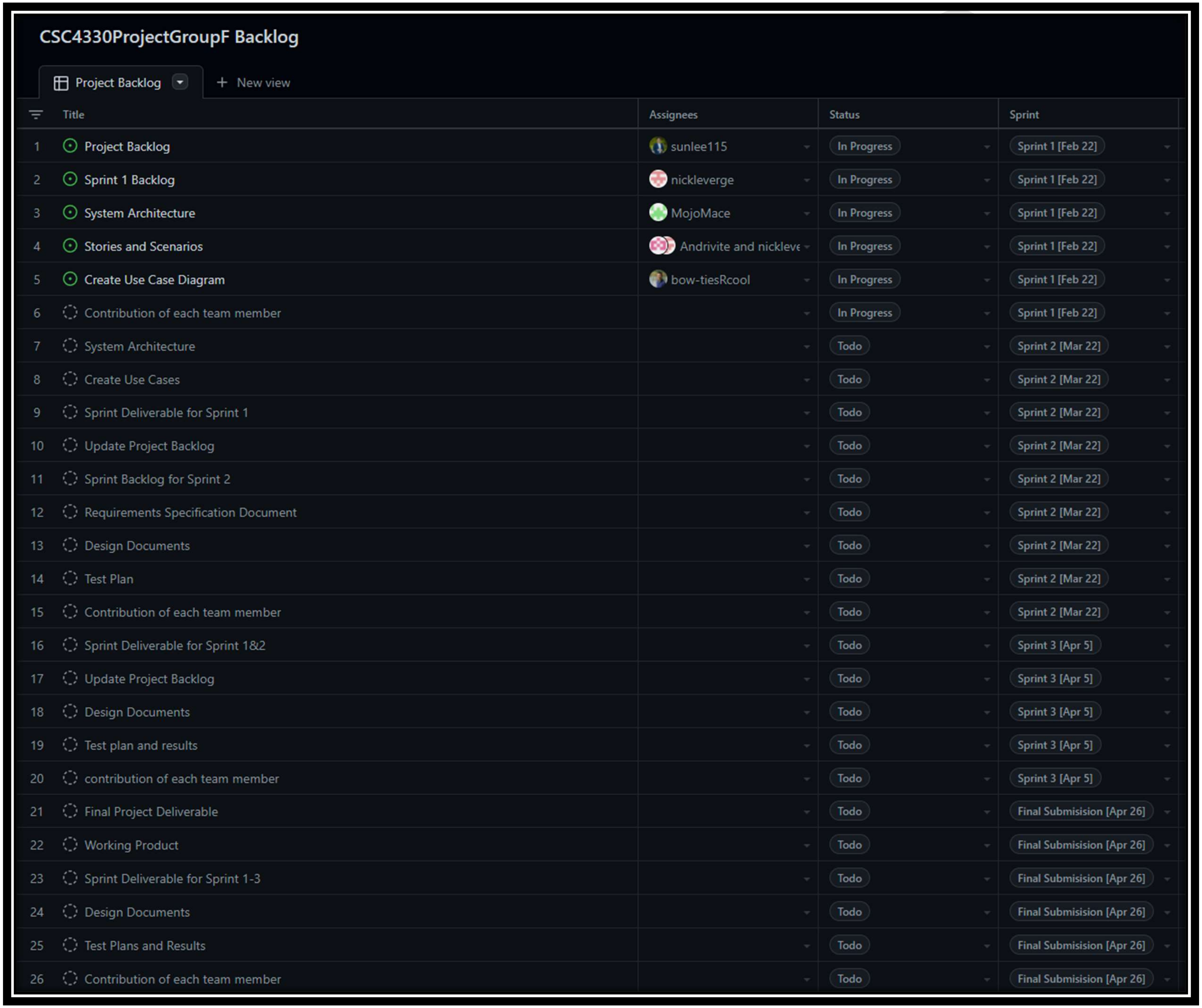
Sprint 1 Backlog and Team Member Responsibilities



Group F Project Backlog

	Title	Assignees	Status	Sprint
1	Create Use Case Diagram	bow-tiesRcool	Todo	Sprint 1
2	Stories and Scenarios	Andrivite and nickleverge	Todo	Sprint 1
3	System Architecture	MojoMace	Todo	Sprint 1
4	Project Backlog	sunlee115	Todo	Sprint 1
5	Sprint 1 Backlog	nickleverge	Todo	Sprint 1

Full Project Backlog



CSC4330ProjectGroupF Backlog

	Title	Assignees	Status	Sprint
1	⊙ Project Backlog	sunlee115	In Progress	Sprint 1 [Feb 22]
2	⊙ Sprint 1 Backlog	nickleverge	In Progress	Sprint 1 [Feb 22]
3	⊙ System Architecture	MojoMace	In Progress	Sprint 1 [Feb 22]
4	⊙ Stories and Scenarios	Andrivite and nickleve	In Progress	Sprint 1 [Feb 22]
5	⊙ Create Use Case Diagram	bow-tiesRcool	In Progress	Sprint 1 [Feb 22]
6	◌ Contribution of each team member		In Progress	Sprint 1 [Feb 22]
7	◌ System Architecture		Todo	Sprint 2 [Mar 22]
8	◌ Create Use Cases		Todo	Sprint 2 [Mar 22]
9	◌ Sprint Deliverable for Sprint 1		Todo	Sprint 2 [Mar 22]
10	◌ Update Project Backlog		Todo	Sprint 2 [Mar 22]
11	◌ Sprint Backlog for Sprint 2		Todo	Sprint 2 [Mar 22]
12	◌ Requirements Specification Document		Todo	Sprint 2 [Mar 22]
13	◌ Design Documents		Todo	Sprint 2 [Mar 22]
14	◌ Test Plan		Todo	Sprint 2 [Mar 22]
15	◌ Contribution of each team member		Todo	Sprint 2 [Mar 22]
16	◌ Sprint Deliverable for Sprint 1&2		Todo	Sprint 3 [Apr 5]
17	◌ Update Project Backlog		Todo	Sprint 3 [Apr 5]
18	◌ Design Documents		Todo	Sprint 3 [Apr 5]
19	◌ Test plan and results		Todo	Sprint 3 [Apr 5]
20	◌ contribution of each team member		Todo	Sprint 3 [Apr 5]
21	◌ Final Project Deliverable		Todo	Final Submisision [Apr 26]
22	◌ Working Product		Todo	Final Submisision [Apr 26]
23	◌ Sprint Deliverable for Sprint 1-3		Todo	Final Submisision [Apr 26]
24	◌ Design Documents		Todo	Final Submisision [Apr 26]
25	◌ Test Plans and Results		Todo	Final Submisision [Apr 26]
26	◌ Contribution of each team member		Todo	Final Submisision [Apr 26]