Phase II: Design Report

Team D

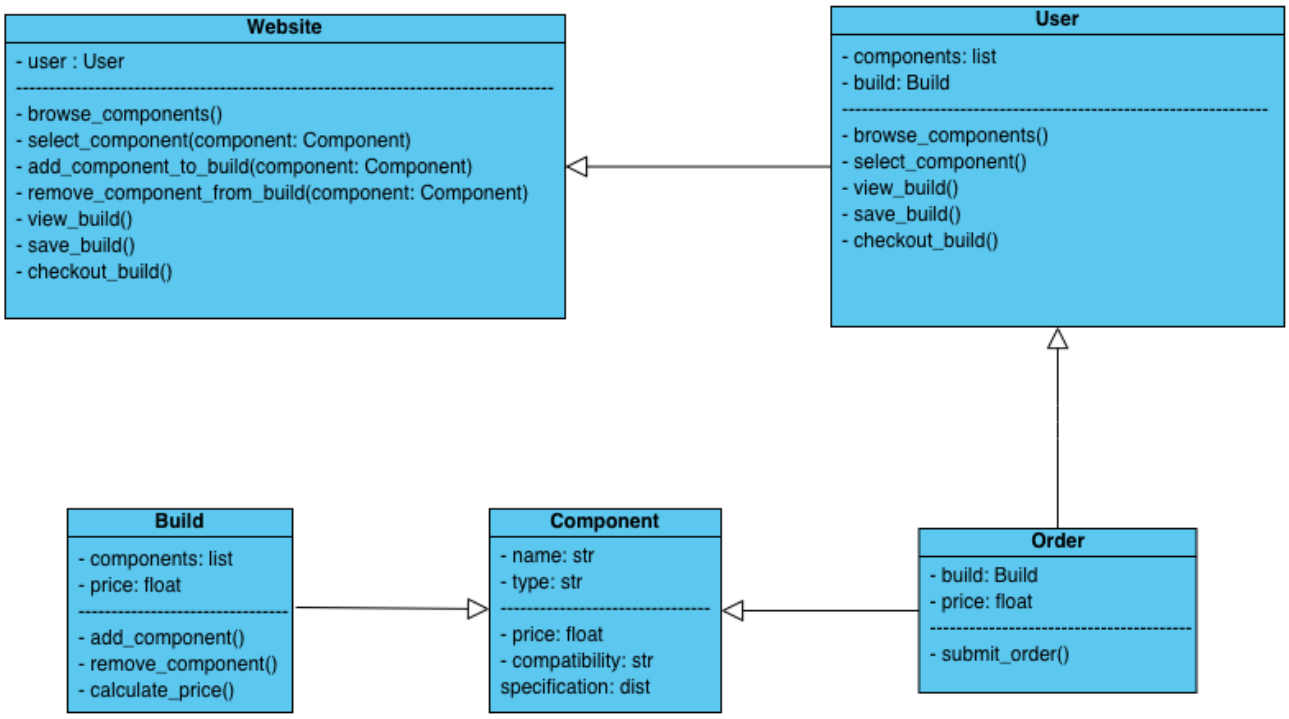
CSC 322 - Software Engineering

Written By:

Zuhayer Alvi, Shakir Hossain, Tahsin Parvez, Bari Al Shafian

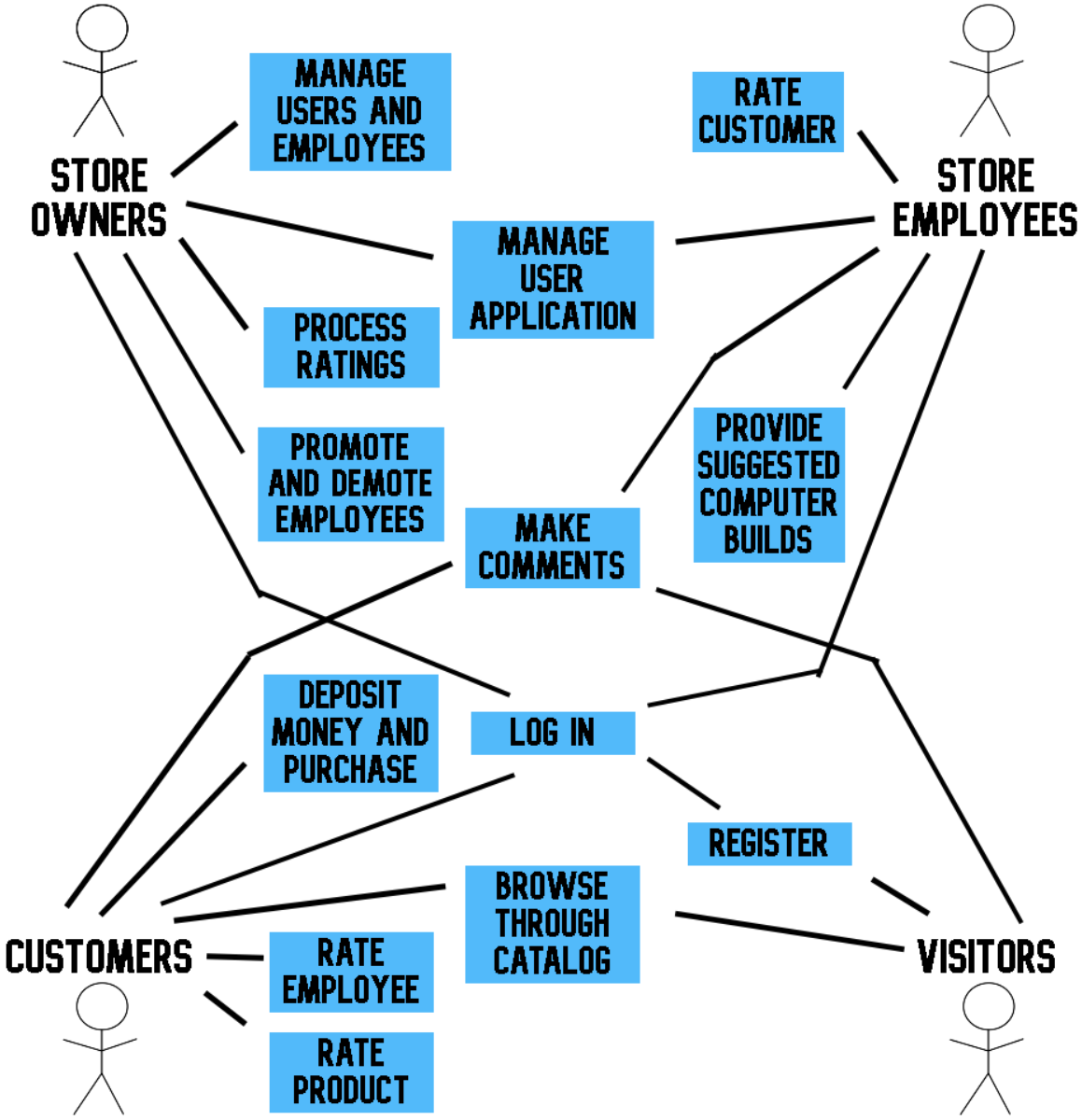
Table of Contents

Introduction



The Collaboration Diagram above shows us the relationship between certain objects and classes within our project, and the order in which they occur during a particular interaction or scenario. It illustrates how objects or classes collaborate with each other to accomplish a specific task or achieve a certain goal.

Use Case Diagrams



Manages Users and Employees

- Allows administrators to manage users and employee accounts. This will provide admins to securely and efficiently manage accounts while ensuring security and privacy of customer and employee information.

Manages User Application

- Allows administrators to manage user applications for account creation and modification(i.e. Account name changes, contact info, etc). Administrators will have the ability to review and approve user applications.

Processes Ratings

- Allows administrators to view, process, and analyze customer ratings and feedback on products, services and employees. Administrators will be able to review and respond to customer feedback while ensuring the relevance, necessity, and accuracy of the data provided in feedback.

Promote and Demote Employees

- Administrators will be able to access employee data, and can demote or promote employees based on their performance, skills, and job responsibilities.

Rates Customers

- This function will give employees to rate and provide feedback on their experiences with specific customers they interacted with. This function should be designed to encourage employees to provide honest feedback, while also allowing the company to monitor and improve the quality of customer service provided.

Makes Comments

- Allows customers to provide feedback and comments on their experiences with products and services. Companies can monitor these comments and the design will make it so the customers are encouraged to leave comments on their experiences.

Provide Suggested Computer Builds

- Clients will be able to view pre-built computer systems that are recommended for their specific use cases or needs. This function will be designed in a way for customers to find the right system for their budget and needs in a simple and efficient manner.

Deposits Money and Makes Purchases

- Circuit City will require a secure checkout process for customers to make purchases. In addition, this function will allow users to add funds into their accounts for the purchase of products. Users will be able to manage their account balances and purchase products through the checkout process.

Browses Through Catalog

- These functions will aid in providing a clear and concise overview of available products, by allowing customers to search, sort, and filter products based on preferences.

Rates Employees

- Customers will be able to rate and provide feedback on their experiences with customer service representatives. This function will be designed to encourage customers to provide honest and constructive feedback, while allowing Circuit City to monitor and improve the performance of its employees.

Rates Products

- Customers will also be able to rate and provide feedback on their experiences with products they purchased. This function will be designed to encourage customers to provide honest and constructive feedback, while allowing Circuit City to monitor and improve the quality of products based on feedback of its clients.

Registers for an Account

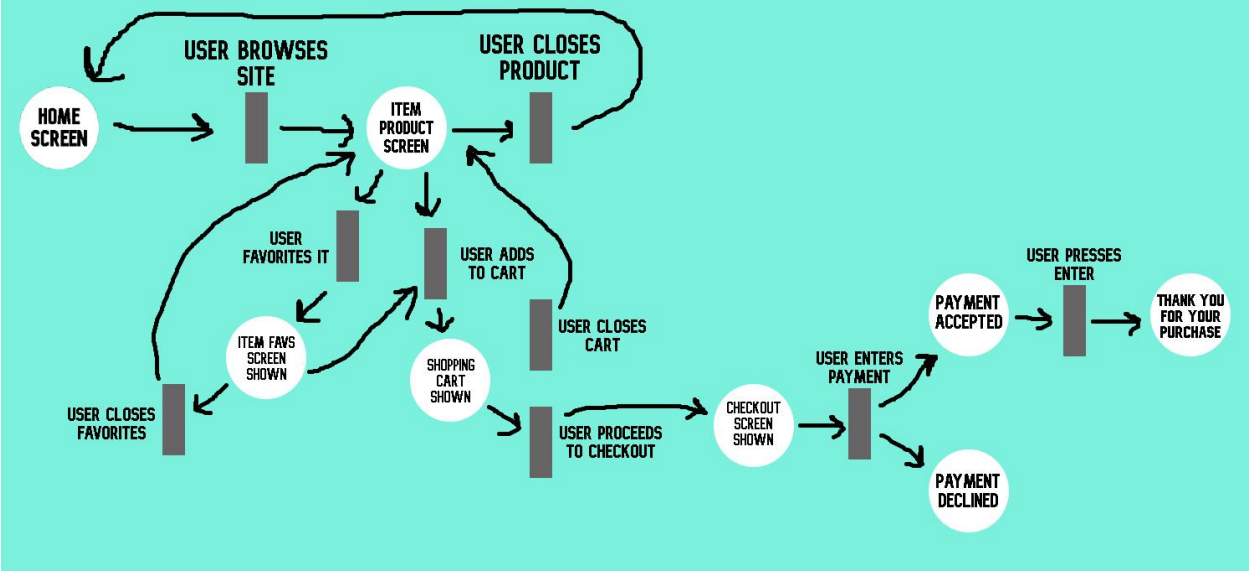
- This GUI function will allow customers to create accounts with Circuit City. This will ensure customers can save information such as products, as well as keeping the privacy and security of their personal information(i.e. Addresses, billing info).

Log in to an Account

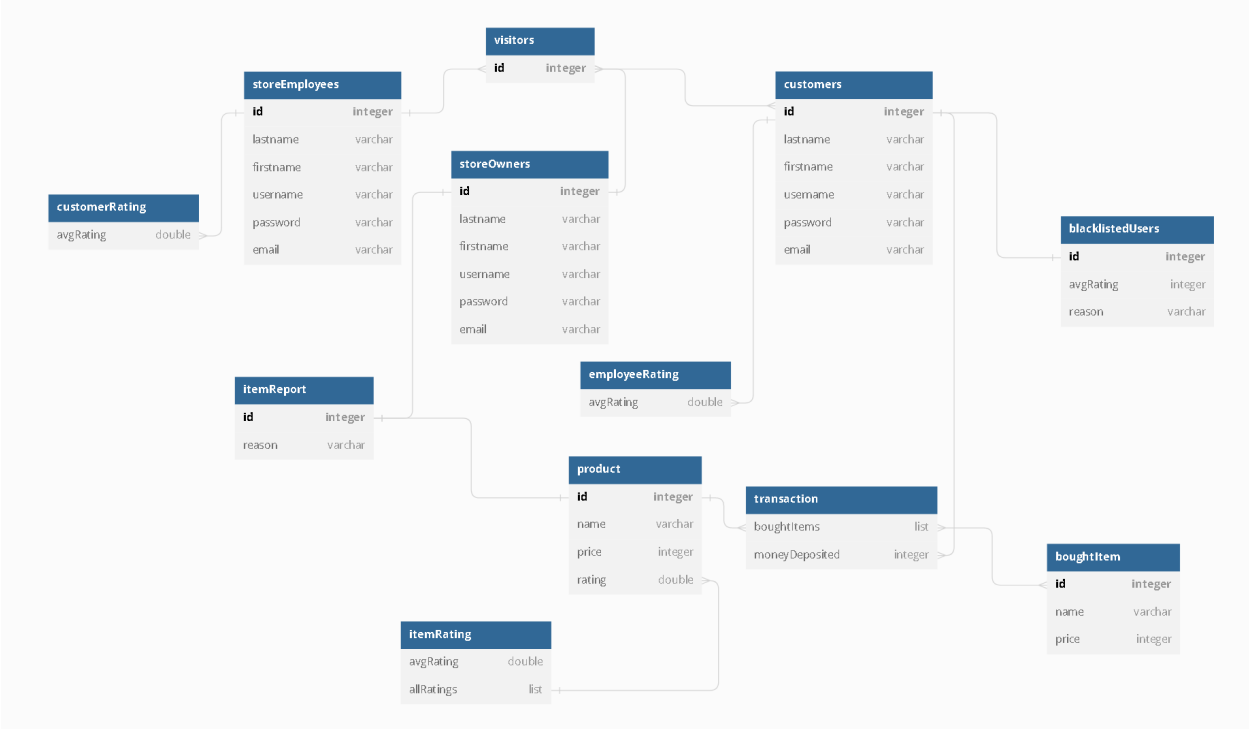
- Will allow customers to access their accounts and information associated with their accounts. In addition, there will need to be a ‘password reset’ and security features such as CAPTCHA or two-factor authentication to ensure a safe and secure login to each account.

Petri Net

Ordinary Use Case Petri Net



ER Diagram



Detailed Design

User Login System:

```
// Check if user is already logged in
if (userLoggedIn()) {
    redirect("homepage.php"); // redirect to homepage if user is
already logged in
}

// Check if the user has submitted the login form
if (formSubmitted()) {
    // Get the username and password submitted by the user
    $username = sanitize($_POST["username"]);
    $password = sanitize($_POST["password"]);

    // Validate the username and password
    if (isValidUsernameAndPassword($username, $password)) {
        // Login successful
        $_SESSION["username"] = $username; // Store the username
in a session variable
        redirect("homepage.php"); // redirect to homepage
    } else {
        // Login failed
        $errorMessage = "Invalid username or password";
    }
}

// Display the login form
showLoginForm($errorMessage);
```

Component Selection for Building PC:

```
// Initialize component variables
$selectedCPU = null;
```

```
$selectedMotherboard = null;
$selectedMemory = null;
$selectedStorage = null;
$selectedGPU = null;
$selectedPSU = null;
$selectedCase = null;

// Display component selection pages
displayCPUPage();
displayMotherboardPage();
displayMemoryPage();
displayStoragePage();
displayGPUPage();
displayPSUPage();
displayCasePage();

// Validate user selections and calculate total cost
if (userSelectionsValid()) {
    $totalCost = calculateTotalCost($selectedCPU,
$selectedMotherboard, $selectedMemory, $selectedStorage,
$selectedGPU, $selectedPSU, $selectedCase);
    displaySummaryPage($selectedCPU, $selectedMotherboard,
$selectedMemory, $selectedStorage, $selectedGPU, $selectedPSU,
$selectedCase, $totalCost);
} else {
    displayErrorPage();
}
```

Customer Depositing Money Into Account:

```
// Check if user is logged in
if (!userLoggedIn()) {
    redirect("login.php"); // redirect to login page if user is
not logged in
}
```

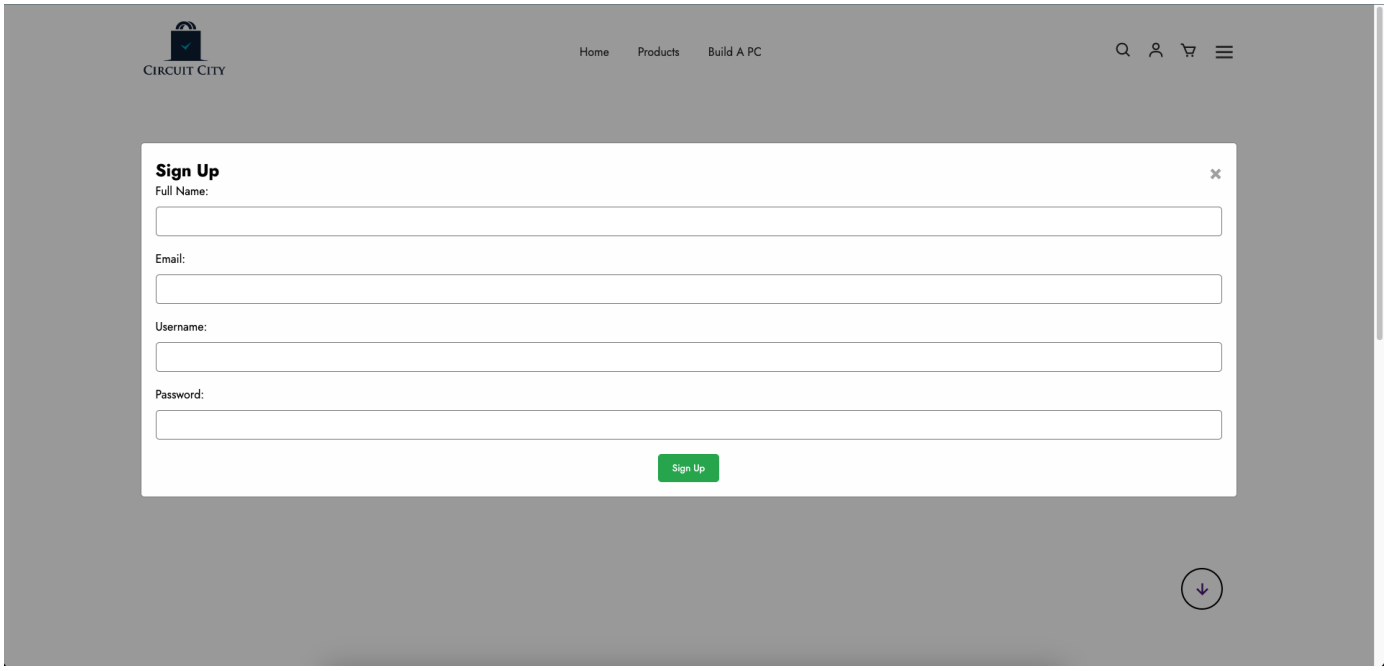
```
// Display the deposit form
displayDepositForm();

// Check if the user has submitted the deposit form
if (formSubmitted()) {
    // Get the amount of money deposited by the user
    $amount = sanitize($_POST["amount"]);

    // Validate the amount of money
    if (isValidDepositAmount($amount)) {
        // Deposit successful
        $transactionID = generateTransactionID();
        $date = getCurrentDate();
        recordDeposit($transactionID, $date, $amount);
        updateAccountBalance($amount);
        displayDepositConfirmationPage($transactionID, $amount);
    } else {
        // Deposit failed
        $errorMessage = "Invalid deposit amount";
        displayDepositForm($errorMessage);
    }
}
```

Customer Rating PC Build

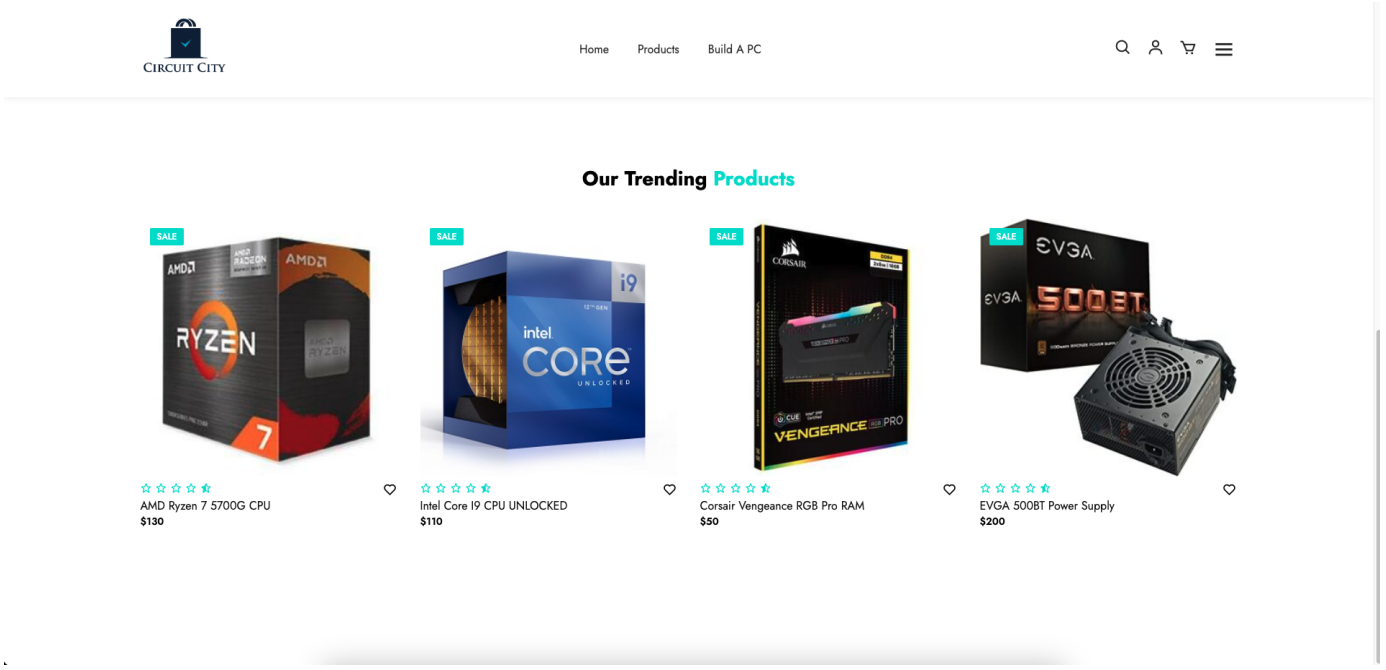
```
// Step 1: Customer selects their PC build and clicks on "Rate
Your Build" button
onClick(RateYourBuildButton) {
   // Step 2: Show rating form modal to customer
   displayModal(RatingForm);

   // Step 3: Customer rates their PC build on a scale of 1 to 5
   let rating = selectRating(1, 5);

   // Step 4: Customer leaves a comment about their PC build
```

```
    let comment = leaveComment();

    // Step 5: Submit the rating and comment to the server
    submitRatingAndComment(rating, comment);

    // Step 6: Show success message to customer
    displayMessage("Thank you for rating your PC build!");
}
```

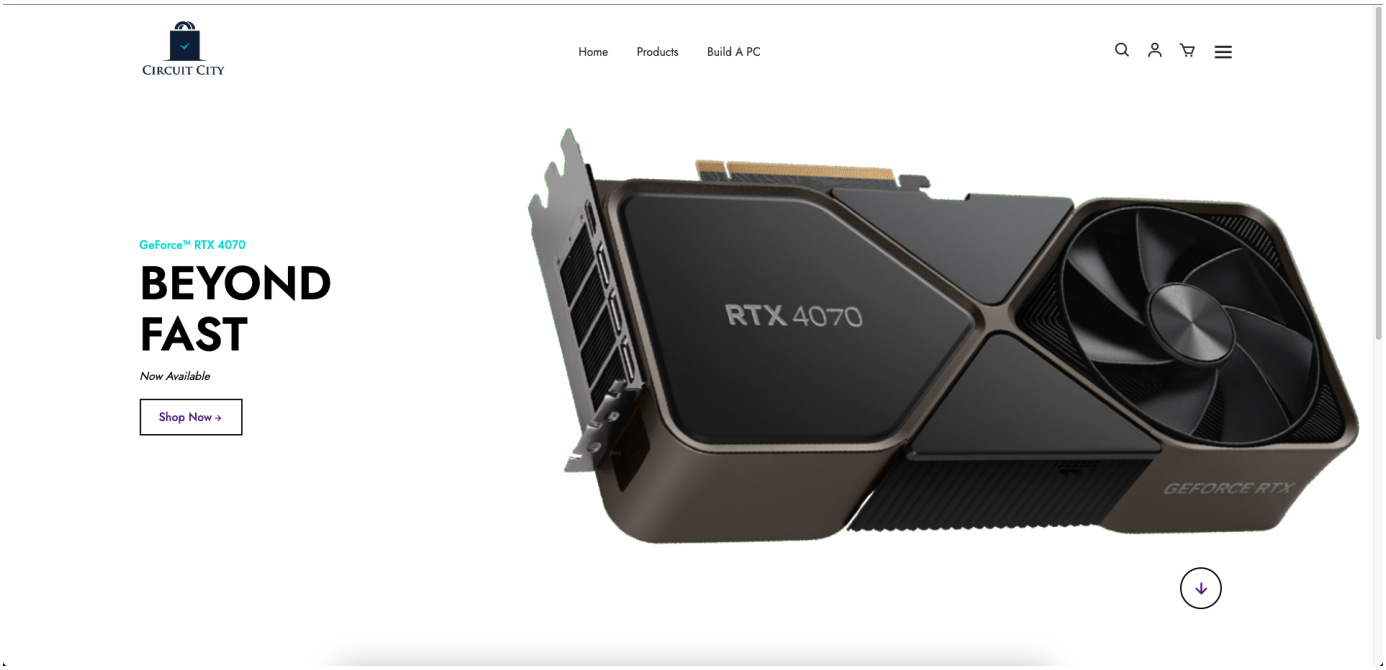
System Screens



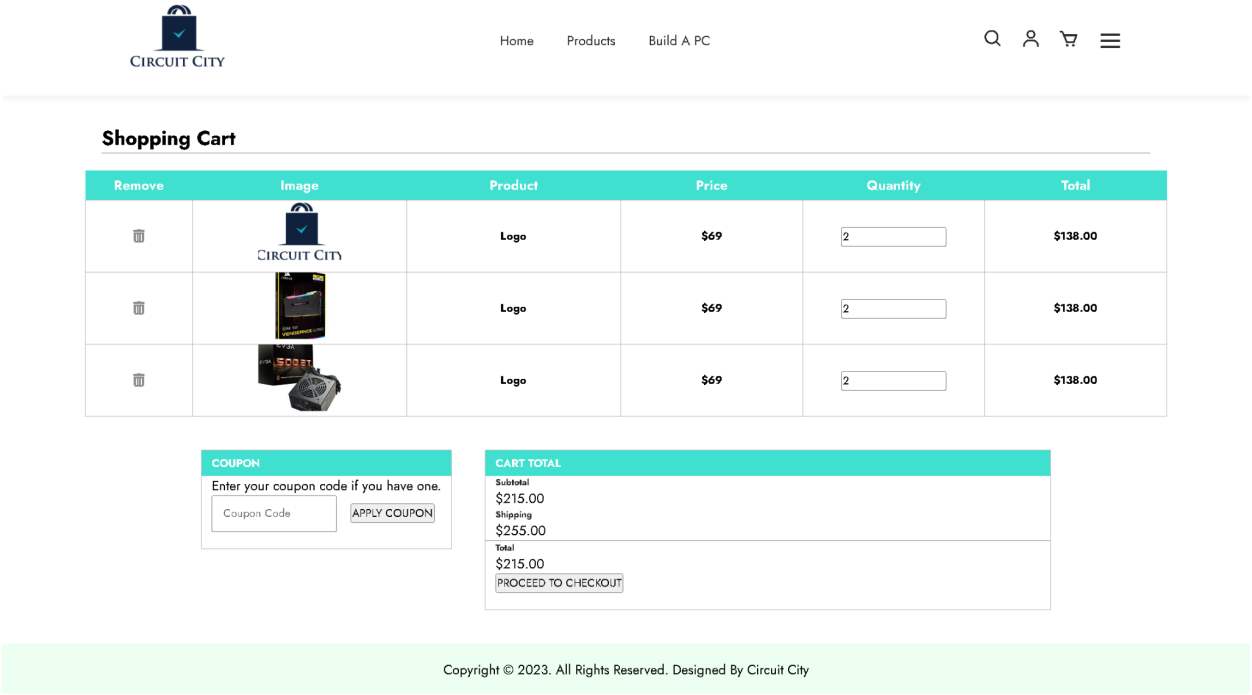
- New user signup modal: Prompts users to signup when clicking on the user icon on the navigation bar.



- Trending products section



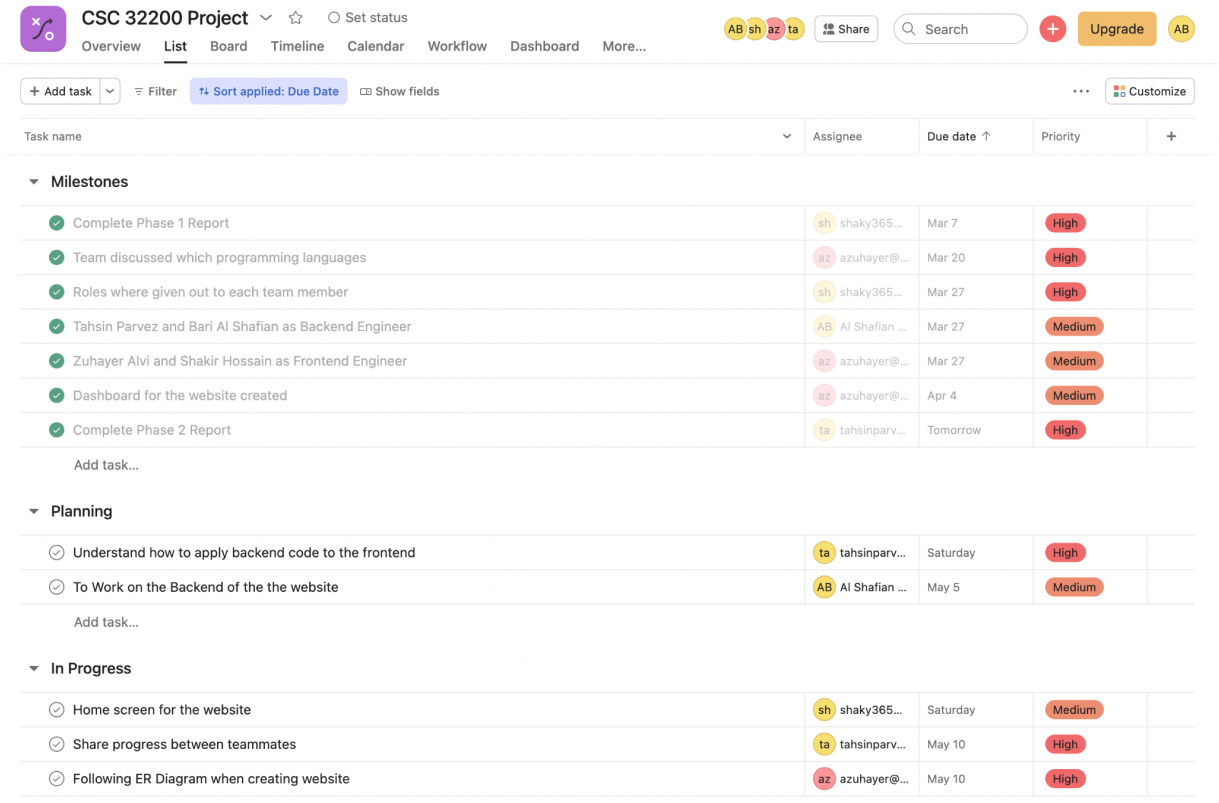
- Home page screen



- Shopping cart screen

Group Meetings/Memos

Date	Agenda
<20/03/2023>	<Discussed which programming languages to use, shared ideas of what we want the website to look like and gave insight to possible UI elements to implement>
<27/03/2023>	<Possible roles were discussed, front-end was given to Zuhayer Alvi and Shakir Hossain while back-end was given to Tahsin Parvez and Bari Al Shafian>
<15/04/2023>	<Outlined the first version of the website, setup main files and Phase 1 report has been completed already>
<20/04/2023>	<Share progress between teammates, give advice on certain areas of the website, and schedule the next meeting as desired>



GitHub Repository Address

- https://github.com/azuhayer/Circuit-City