

CS3704 PM3

Aneka Busam, Anurag Kulkarni, Arnav Tikhe, Bhavya Shanmugam, Deepika Reddy, Rushil Mittal, Vansh Parikh

High-Level Design:

To structure our system, we plan on using event-based architecture. Event-based architecture comes with many benefits such as flexibility, scalability, and asynchronous processing which is a huge benefit for our application. By using event-based architecture our application will become more adaptive and efficient in terms of recommending products to our users based on their filters and specifications. Additionally, every event can be processed and used for later which provides a deeper analysis of what the user is looking for, therefore making our application more efficient and easier to use. Our application focuses on customer experience and data processing, therefore other high-level design techniques could be used. However, event-based architecture does seem like the best tool because of its dynamic ability and efficient data processing along with the simplicity of its user interface.

Low-Level Design:

Our low-level design will contain user preferences and a rating system for simplicity:

Pseudocode for user input:

Function userPreference():

 Display “Enter preferences below”

 Preferences = {}

 Preferences[‘height’] = input(“enter height”);

 Preferences[‘weight’] = input(“enter weight”);

 Preferences[‘top size’] = input(“enter your size for tops”);

 Preferences[‘bottoms size’] = input(“enter your size for bottoms”);

 Preferences[‘price’] = input(“enter price preference”);

```
save(preferences);
if(save)
    Display "Preferences saved";
Else
    Display "Preferences not saved";
```

Pseudocode to submit a review:

```
Function submitReview(item, rating):
Review['item', 'rating', 'comment'] = item, rating, input('Comment below your
thoughts about the item);
```

```
If review != null:
    display("Thank you! Submitted review);
Else:
    display("Problems submitting review. Please try again");
```

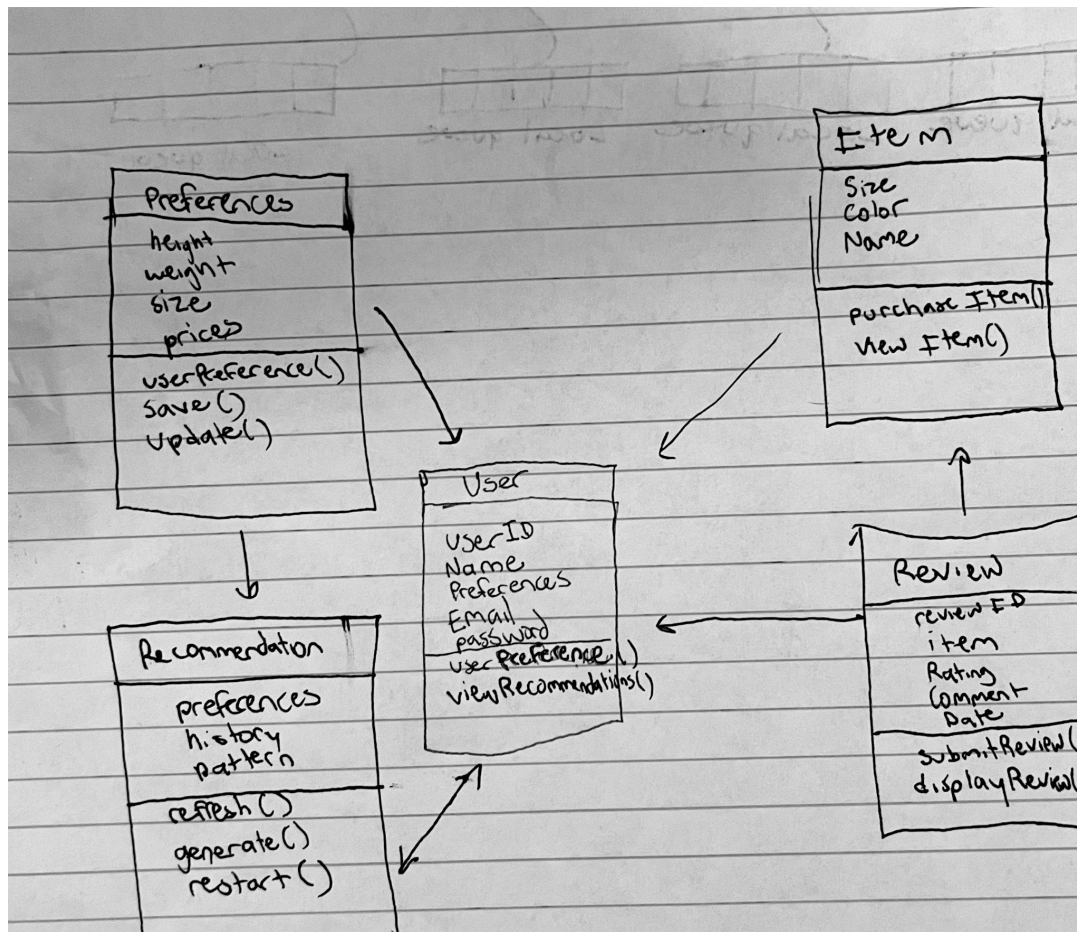
Pseudocode to display a review:

```
Function displayReview(reviewlist):
For review in review[]
    Display review;
```

Explanation: Each of these functions are examples of pseudo-code for this project. They all show a very low-level idea of how our code flow will look.

For our user preferences, this saves basic necessities for the algorithm to find and optimize selections that will better cater to the client than random articles of clothing. To submit a review, it takes parameters of the item and rating, allowing for user input for the comment. It then checks if the review was submitted or not using an if-else. For displaying a review, it iterates through the array of reviews to display it on the interface. A creational pattern being used could be beneficial due to the need for independence when creating users, labeling preferences, and utilizing reviews. A strategy pattern can also be used to dynamically change code or algorithms when necessary.

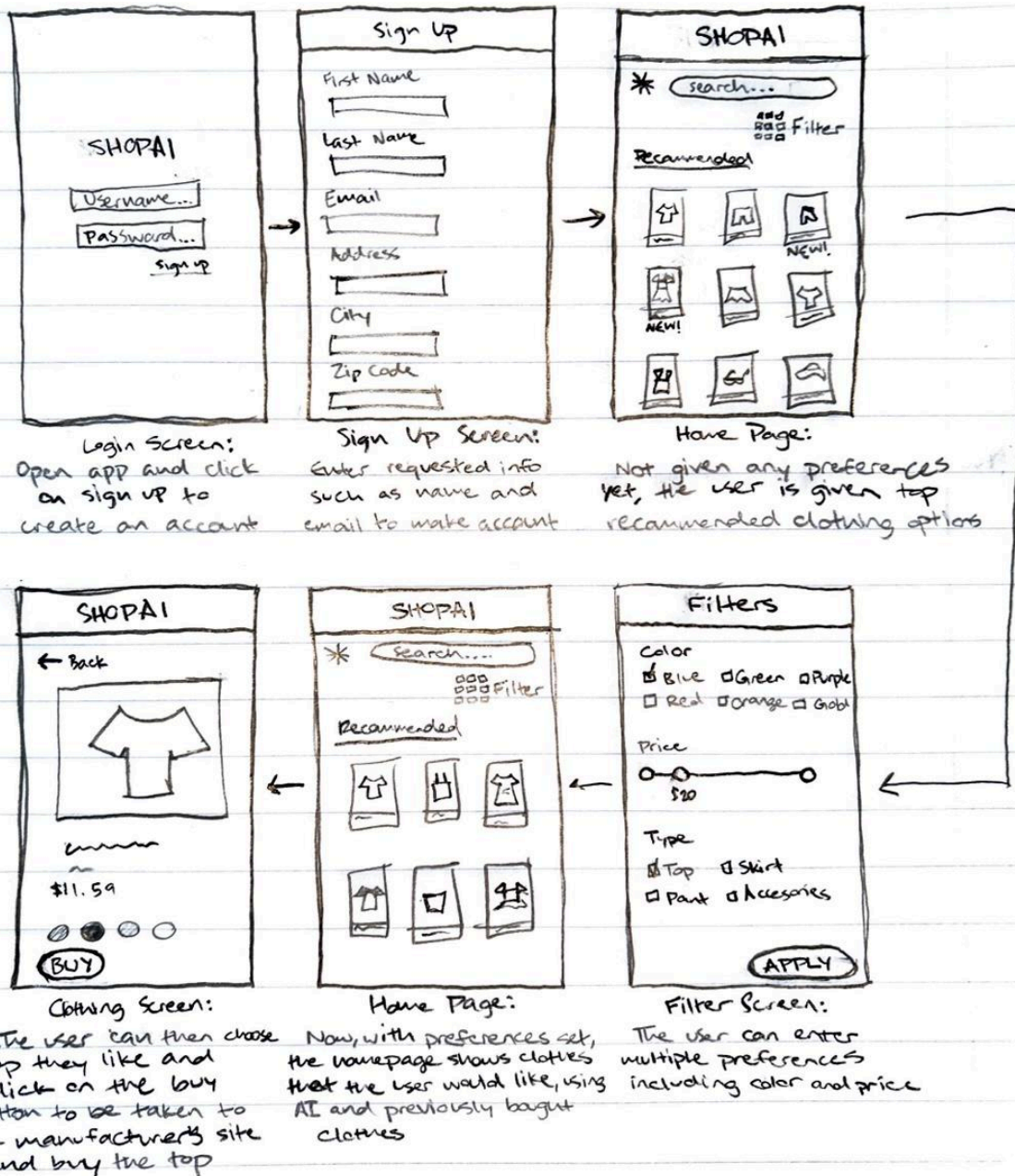
Class Diagram:



This class diagram shows the connections between a few classes: user, item, review, recommendation, and preferences. These all are interconnected within each other, have their own attributes and methods, and play a role amongst each other. The user is put at the center as all of these functions are necessary for the user. The user class includes attributes with methods that show how this class handles user profile information and conducts interactions. The preferences class holds personal information as attributes and manages such information. The item class holds information about the actual article of clothing, enabling the user to view and purchase such items. The recommendation class holds preferences and history to generate recommendations based on the user's preferences and past behavior. The review class holds attributes of the review to create and display user reviews.

Design Sketch (Mock User Interface):

Storyboard: Buying a cheap blue top



We have chosen to design our application as we did above for various reasons. For one the login page allows users to log into their application ensuring security

between accounts and a more personalized experience for each user's preferences. Our home page allows users to set their preferences and then get recommendations for certain clothes and products. That page then takes the user to a product page where they can buy the product they are interested in straight from the application itself. We have chosen to design our application like this because it provides a sequential order of how the process works as well as providing an easy-to-follow interface for our users. Our application is designed as such because we have prioritized easy navigation and an easy user interface.

Project Check-in:

The project check-in survey was submitted along with the attached repository link.

Process Deliverable:

Scrum Meeting Notes - Team Shop Project

Date: 04 / 05 / 2024

Attendees: Aneka Busam, Anurag Kulkarni, Arnav Tikhe, Bhavya Shanmugam, Deepika Reddy, Rushil Mittal, Vansh Parikh

Duration: 1 Hour

Agenda:

1. Review of the last meeting's action items
2. Progress updates from each team member
3. Roadblocks or challenges faced
4. Planning next steps

1. Review of Last Meeting's Action Items

- Finalize functional and non-functional requirements documents.
- Initial planning for integration with payment and online catalog APIs.

Meeting Notes:

Rushil Mittal

~ Discussed how to incorporate pipe and filter design methods to build our application.
~ Finalized with the group that pipe and filter are the best tools in terms of providing recommendations and filtering data.

Aneka Busam:

~ Summarized what was discussed in the last meeting and checked up on how far along the team is with the current tasks.

~Planned a meeting with Anurag to help him design the user interface

Anurag Kulkarni:

~ Discussed progress on designing the user interface and what features to add/remove from the last scrum meeting

~ Needed help with suggestions for the user interface design - planned a meeting with Aneka

Arnav Tikhe

~ Discussed progress on an algorithm for a recommendation system in terms of clothes and products

~ Discussed roadblocks and challenges faced while talking about progress on algorithms such as personalization, and ethical/privacy controls

Bhavya Shanmugam

~ Provided pseudocode for our application - more specifically for user input, submitting a review, and, displaying a review

~ Initiated conversation on the next steps and what can be done to improve our application

Deepika Reddy

~ Discussed the next steps in our application development process, including help with the AI algorithm for the recommendation system

~ Discussed how she helped Arnav with the recommendation system for clothes and the filtering system

Vansh Parikh

~ Discussed how the user login screen should be designed as well as the user sign-up page

~ Provided the low-level design and pseudo-code for our application