Software Development Final Project - Outfit Tinder

Team Number: 5
Team Name:
Team Members: Mary Kodenkandath, Soumya Devulapali, Nandini Nema, Matis Ul de Morais
Application Name: Outfit Tinder
Application Description:

Clothing tinder is an application that aims to help young adults pick out their clothing for the day. Everyday most people struggle to figure out what they are going to wear and usually waste time in their day doing it. Especially for busy students who have so much going on, having to think about choosing an outfit can be very difficult. This app can save young adults so much time in the morning, and choose an outfit instantly with just a couple clicks on this app. Users will also be able to get outfit inspiration by swiping through their for you page and be able to like clothing they may want to buy in the future. Not only will users be able to choose their outfit, but choosing your outfit will just become playing a game every morning!

Clothing Tinder works by presenting an example outfit choices to the user when the user logs in. The user can choose to “swipe right” or click a green check mark on the outfit if they like it and or they can “swipe left” if they don’t like the outfit. The outfits that they like, will be saved into their database and they can see the clothes that they have liked for their own inspiration. The user also has an option of uploading pictures of their own clothes, first tops and then bottoms, and can do a tinder swipe through their own clothes to put together an outfit for themselves through the morning.

Vision Statement: For young adults who struggle with fashion inspiration. Outfit Tinder is an online application that not only will give you outfit inspiration, but also pull pieces from your own closet to cultivate unique outfits. Unlike Pinterest, our product allows users to upload pictures of their own clothes, and turns the dreaded activity of picking out your outfit in the morning into a fun game.

Version Control: On repository

Developmental Methodology:
We will be using trello to track our progress and update each other who is doing what. Each week during our meeting we will talk about what we have done and what we will do for the upcoming week. We will start by using the agile methodology. We will start by talking about our requirements for the week, and then have sprints and each team member will have to get an assignment done for the week. Then at the end of the week and the next meeting we will collaborate together and develop as a team. Then we will create plans for the next week and continue the cycle while reviewing and monitoring our progress.

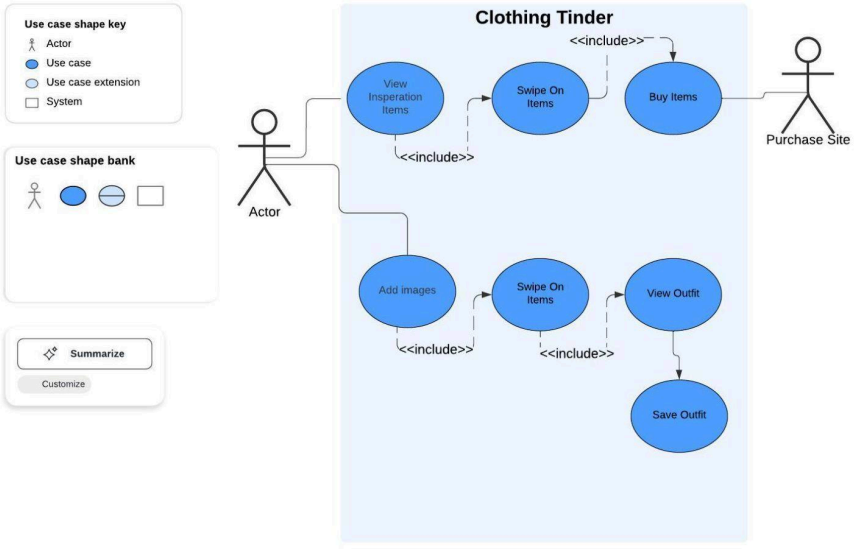
Communication Plan: Our team plans to communicate through i messages. We plan to communicate regularly and have weekly updates and checkpoints for ourselves.

Meeting Plan:

Team Meeting: Tuesday/Thursday at 4:00 p.m. (in-person)

Weekly Meeting: 3:15-3:30 p.m. on Mondays (Zoom)

Case Diagram:



Wireframe:

MVP:

-Updates on Github Project Board and Trello project

-Login Page

-A registration page

-A home page

-Inspiration page where users can swipe right or left and those pictures get stored somewhere

-page of stored websites

-A UI that allows the user to interact with the application - this can comprise of multiple pages

-A server that allows the UI to communicate with the database

-A database that stores user information

-Passwords must be hashed and stored in the database

-Session Management - The user must be able to log in and out of the application and the session must be maintained

-Application is built within Docker containers - you can find some updates to the docker-compose.yaml in the write-up below.

AT THE END:
-Project Report
-Project Presentation

EXTRA:
-A profile page (where the user can see their information - optional)
-A database that stores data from an API (optional) - if you are requesting repeat data from the API with every call, you could cache some data in your database.

-Expected Tools:

- Project Tracker (Trello)
- VCS repository (GitHub)
- Database (MySQL, PostgreSQL or similar)
- IDE (VSCode)
- UI Tools (e.g. Wireframing tools, HTML, HandleBars etc)
- Application Server (NodeJS)
- Deployment environment (LocalHost, Heroku, Google, or similar)
- External APIs (e.g. Google Maps, Twitter etc.) *if applicable*
- Testing tool (PyUnit or similar) - *if applicable*
- Auto-documenter (optional. Doxygen or similar) - *if applicable*
- Additional tools you may have used - *if applicable*
- Framework (e.g. Laravel, Ruby on Rails, Node.js, Android Studio) *if applicable*
- Hardware (e.g. RaspberryPi, Arduino) *if applicable*

-