MAAKDE

Fashion Design Web Application

Software Design Document

Name(s):

Kennedy Johnson
Erik Vardumyan
Aaron Gian Madrid
Andres Macias
Matthew Kim
Deni Cabaravdic

Date:

11 November 2022

Revisions

Version	Primary Author	Descriptions of Version	Date Completed
1.0	MAAKDE	Initial Release	11-20-2022

Table of Contents

1. Introduction

1.1. Purpose

The purpose of this documentation is to outline the project's specifications and expectations. This SDD document will break down each aspect of the project's progression from its overall overview to the minor components and go over its overall architecture. This document is intended for Project Managers, Software Engineers, and any other positions involved in the implementation of the web application.

1.2. Scope

This document describes the implementation details of the EyeCandy (EC) Web Application. EC will consist of several major components: Recommending Outfits, the mySQL Database, User Preferences and potentially more.Each of the components shall be explained in the details of this document.

1.3. Overview

This document will go over the overall architecture of the application, including the database and its interactions with the front end of the application. The document will also include UML diagrams for important elements of the applications.

1.4. Reference Material (Optional)

- MySQL: mysql.com, 'About MySQL'. [Online]. Available: https://www.mysql.com/about [Accessed: 10- Nov- 2022]
- Tensorflow: tensorflow.org, 'Why TensorFlow'. [Online]. Available: https://www.tensorflow.org/about [Accessed: 10- Nov- 2022]

1.5. Definitions and Acronyms (Optional)

- EC: EyeCandy
- SDD: Software Design Document
- API: Application Programming Interface
- JS - Java Script
- UI - User Interface
- User - The user is the client accesses the application
- AI - Artificial Intelligence
- ML - Machine Learning
- MySQL - Database Management System
- TensorFlow - library for machine learning and artificial intelligence

2. System Overview

2.1. System Overview Diagram

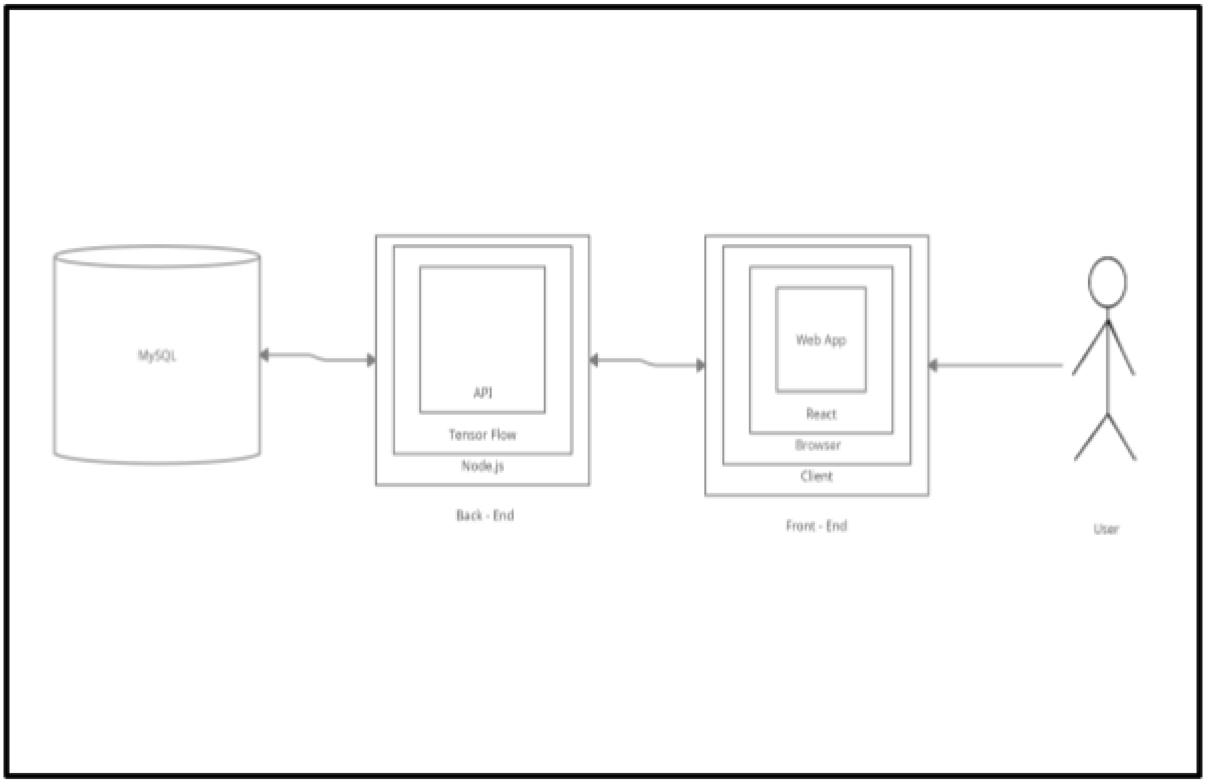


Figure 1

In Figure 1, it shows the architectural structure chosen for the development of the EC. The back-end will be in charge of communicating and retrieving information from the APIs the EC system will be dependent on. NodeJS was the chosen framework due to its quickness and efficiency, and is the framework that the team has the most familiarity with. Furthermore, NodeJS is also highly scalable which will allow the system to grow further and accommodate a wider range of customers.

The EC system will also be storing personal information about its users as well as details about their outfit selections along with other users. Storing these details will enable the system to be more efficient in the long run. Having the ability to visit the database for the same articles of clothing will help with improving efficiency. It will also have the ability to store user information and will enable the system to maintain user profiles and keep records of their outfit builds. Additionally, by following this architecture and structure, it can potentially be extended further into a web application.

3. System Architecture

3.1. Architectural Design

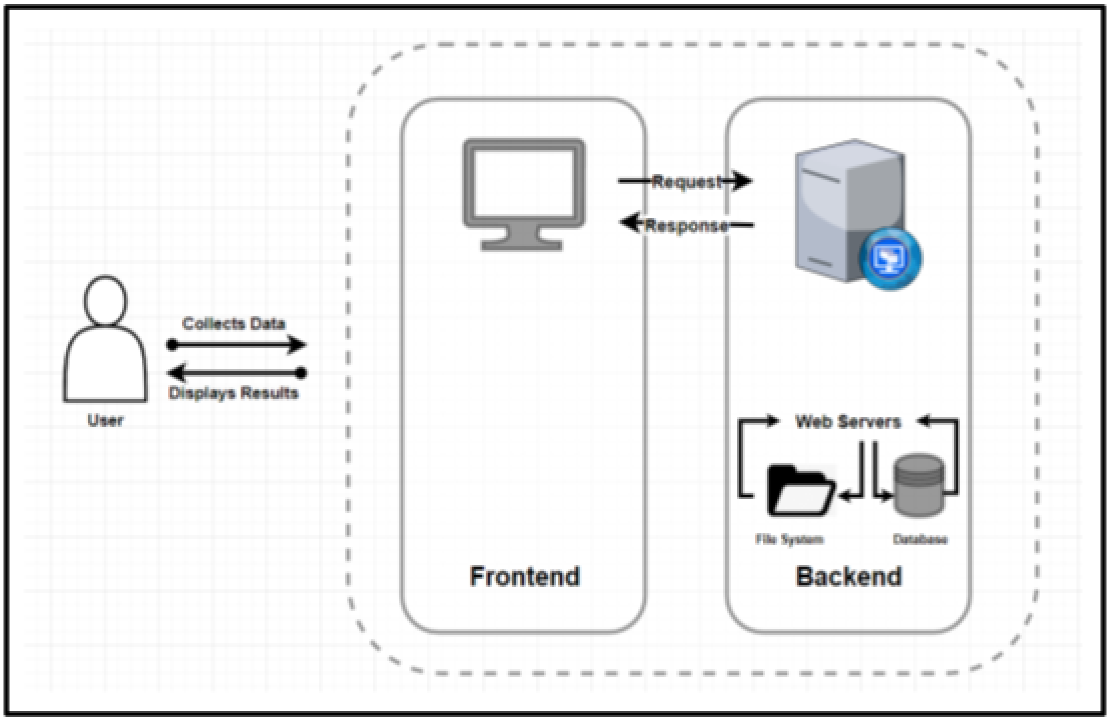


Figure 2

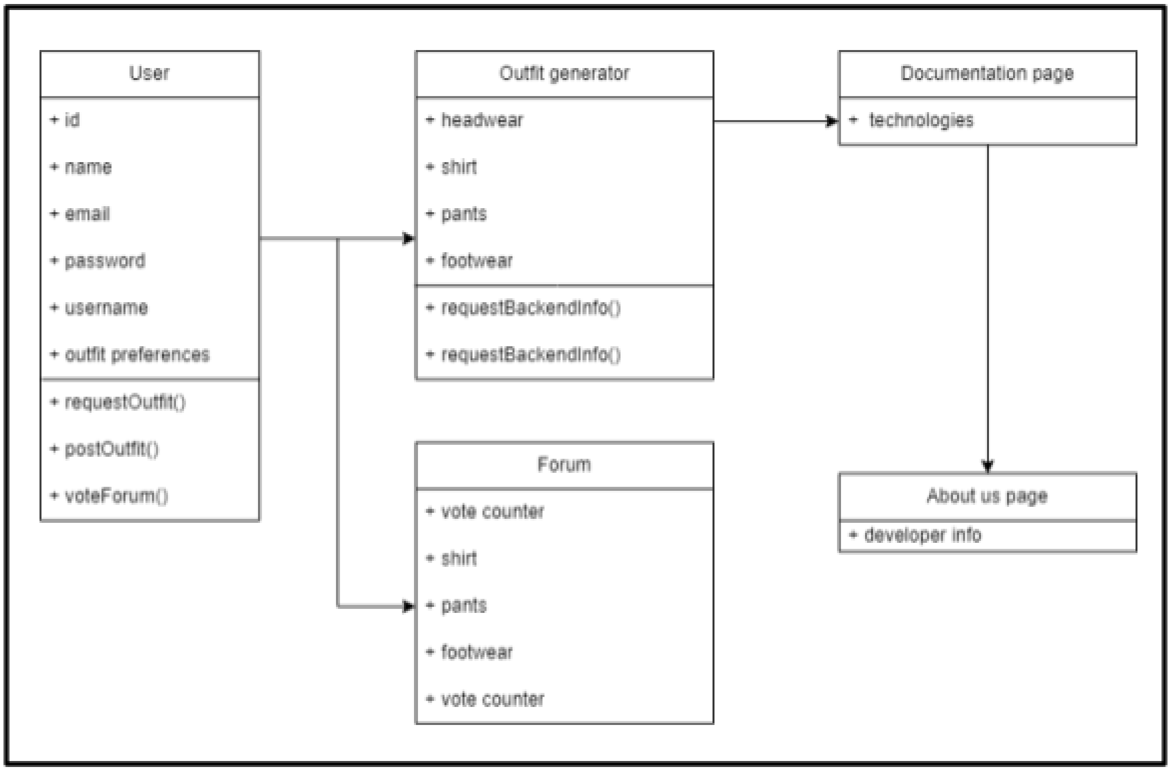
The structure will follow the basic architecture of a simple web application. This architecture seems to be the best in terms of security, since the user will not be in direct access with the data. The user will give data to the front end which will process the information and request access from the backend to store the data. Using this method is beneficial due to the high scalability it provides since each of the servers are on multiple machines. The backend will consist of the mySQL database and the Tensorflow models that were created, and this will be where the data will be saved and stored. Once it is stored, the backend will give a response back to the front end server and display the results of the action to the user.

3.2. Interface Design

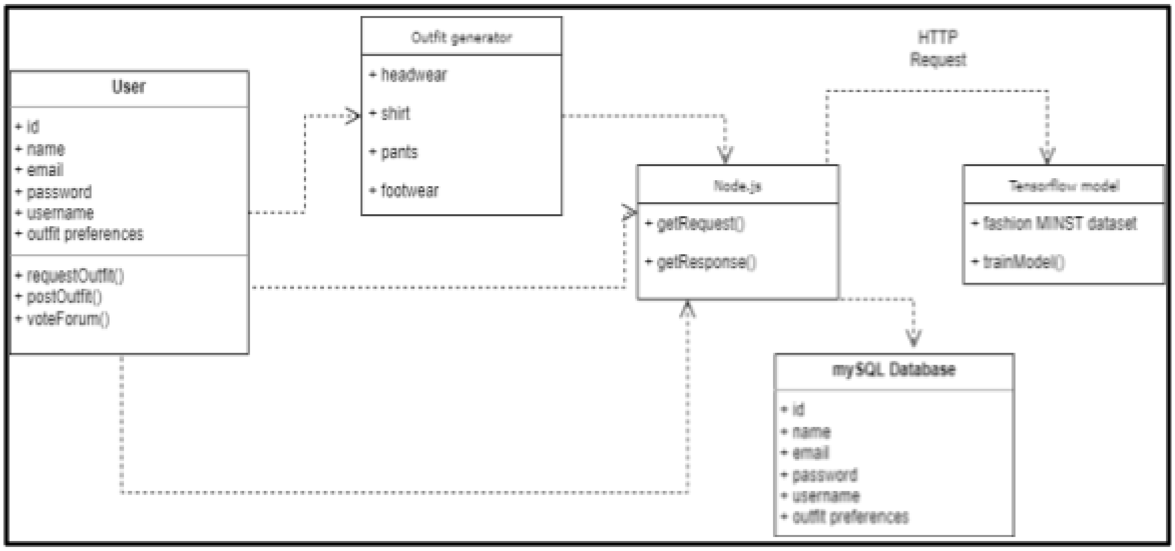
The way the interface has been designed is to allow an ease of access for the user. When trying to access their previous items or information, the user shall simply be able to login to their account and have immediate access to their previous outfits that will be saved to the user's account through the database.

3.3. Decomposition Description

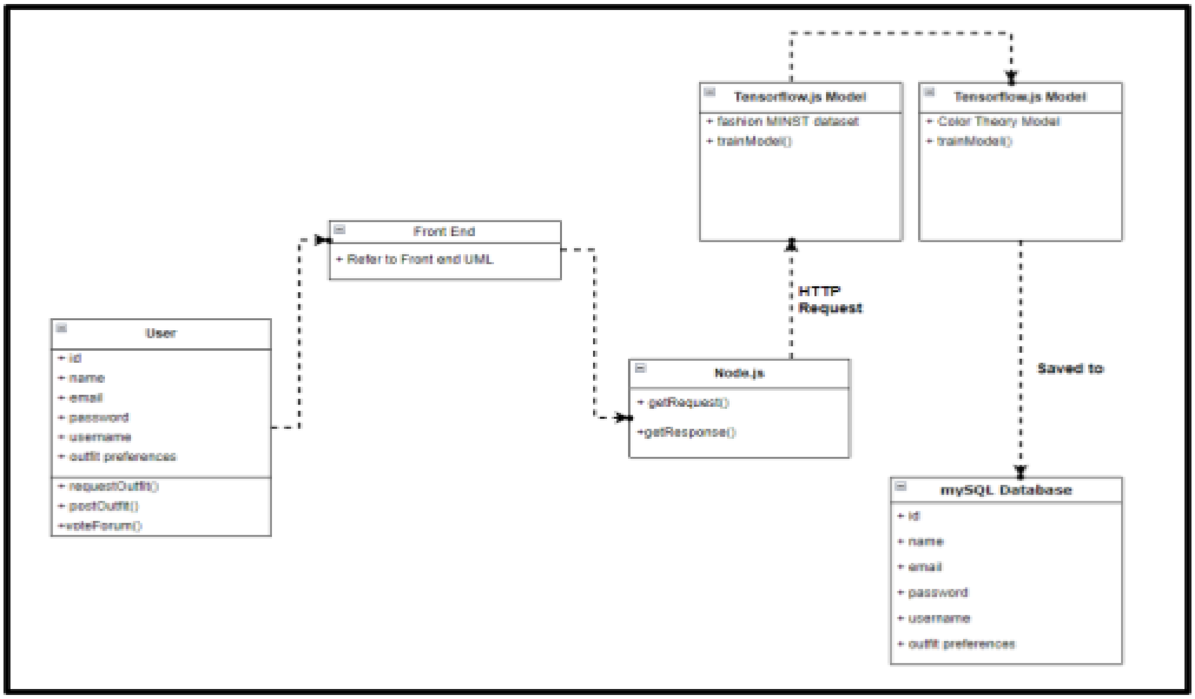
3.3.1. Front-end UML Diagram



3.3.2. Front-end to Back-end UML Diagram



3.3.3. Back-end UML Diagram

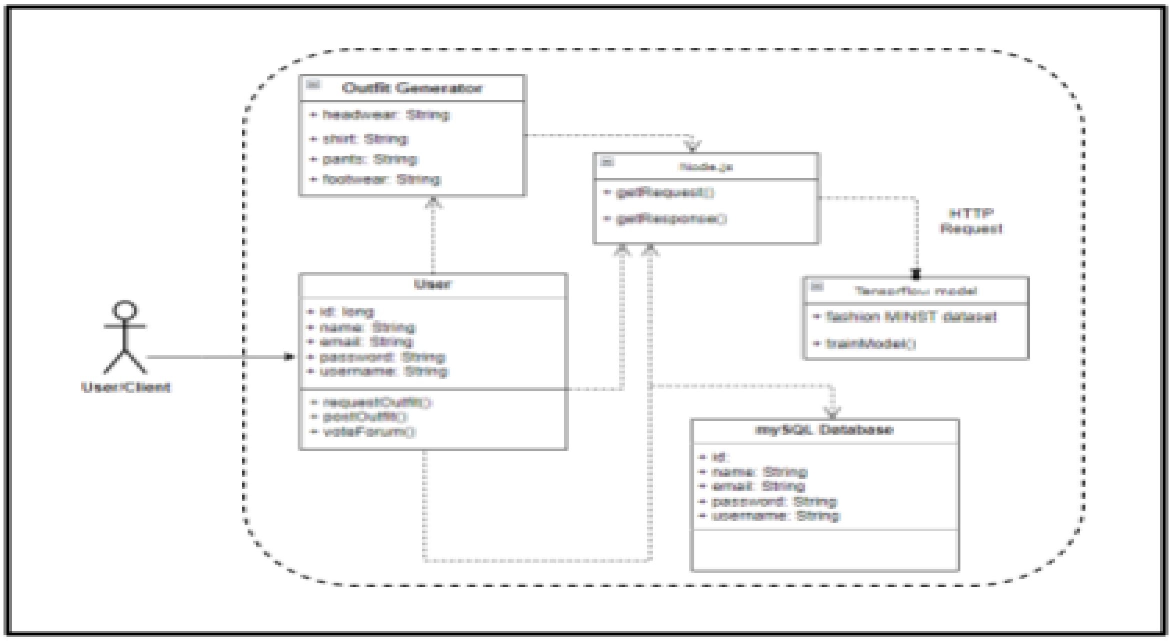


3.4. Design Rationale

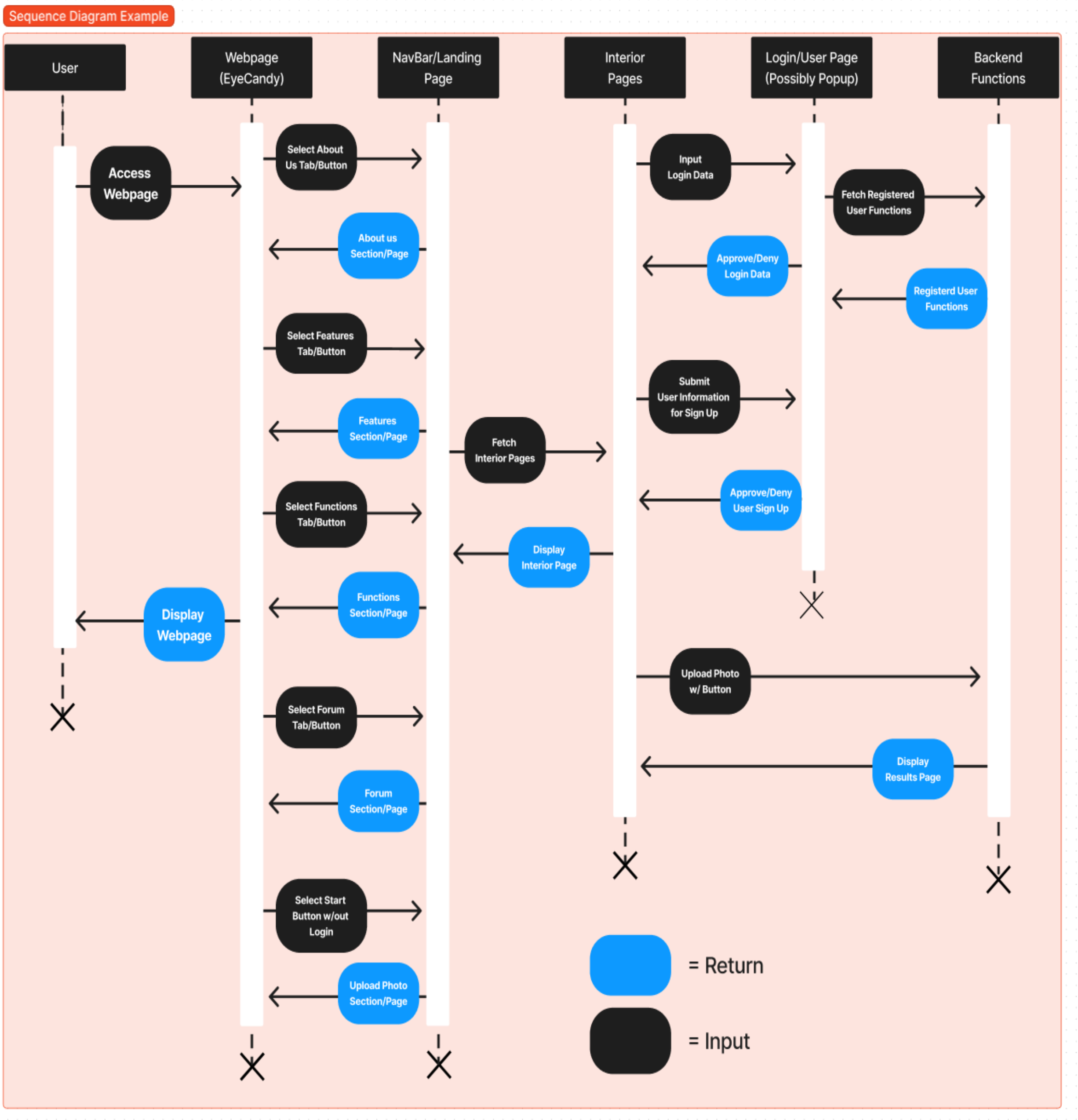
Regarding the design, it was decided that a web application would be developed as it would be more familiar to the engineers and something they could work with more easily. A client server architecture is being utilized because there are no other alternatives for a web application. A database is being used to store user information, outfits, and more. The server is meant to provide endpoint services that's accessible to the user upon requests. While the client is what the user is interacting with with our front-end for services which are processed with a REST API. A REST API was chosen over alternatives such as GraphQL because REST APIs are widely used and have been around for so long that they've developed a level of maturity with stability and convenience unlike others such as GraphQL. Images are a crucial part of the application and the idea of a mobile application will have the ability to take images on the go. Through some research, it was found that the timeline would be unrealistic down that path because most of the engineers are unfamiliar with the tech stack required for mobile development and are more comfortable with web development.

4. Component Design/Detailed Design

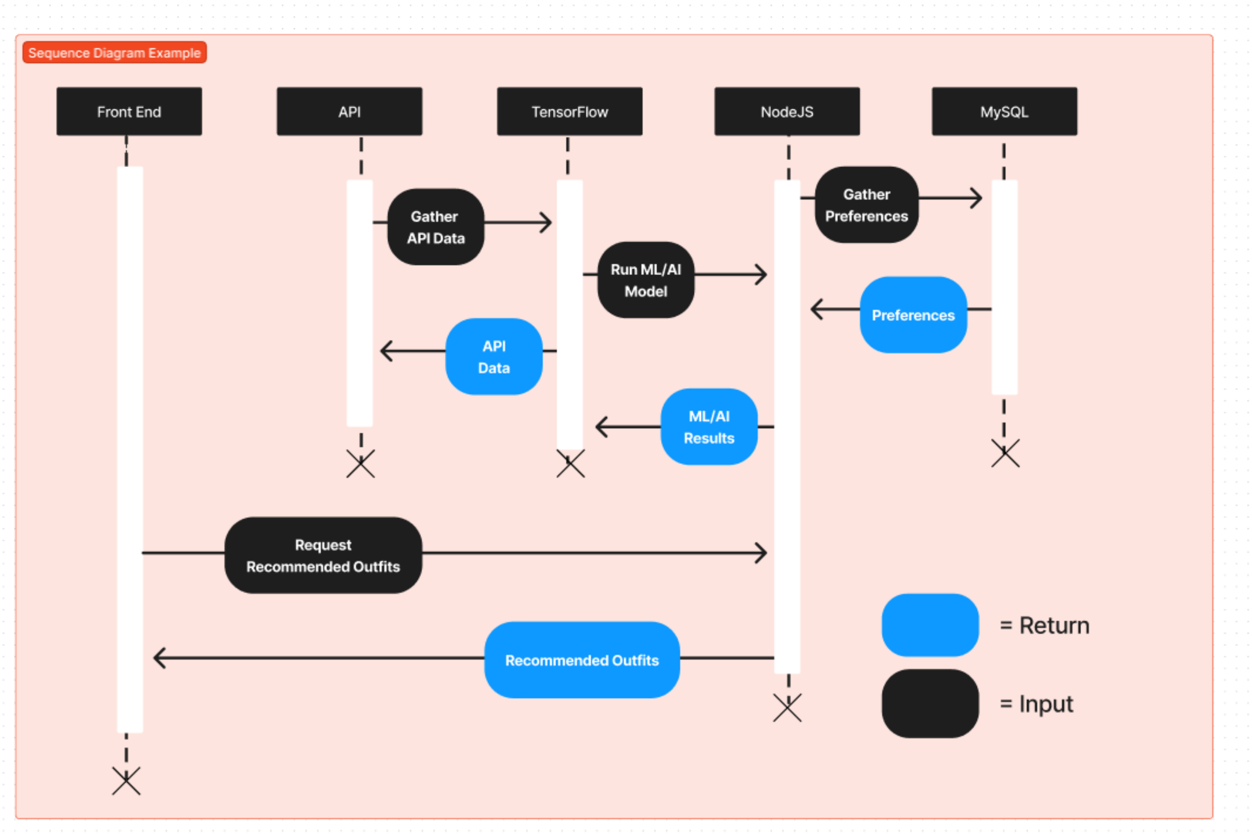
4.1. Complete Class Diagram



4.2. Front-end Sequence Diagram



4.3. Back-end Sequence Diagram



5. Human Interface Design

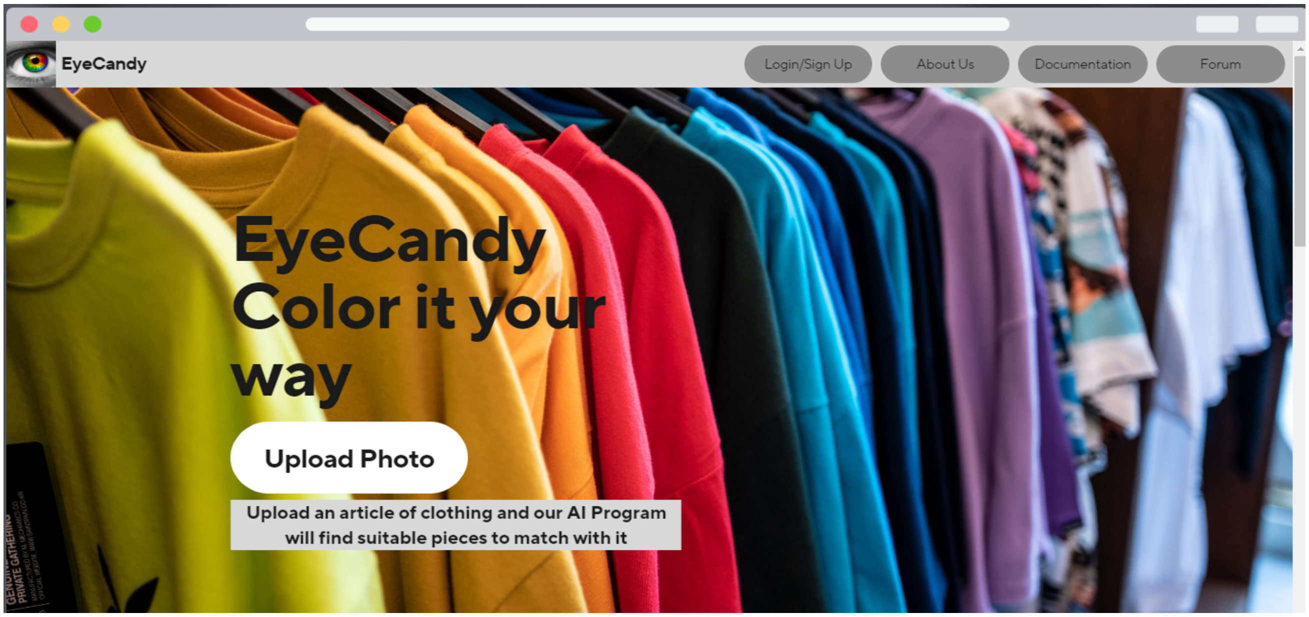
5.1. Overview of User Interface

A user will be able to login to the page through a specific password and username. Once logged in, the user will be allowed to skim through previous outfits that have been stored in their account through the database. If they want to upload a picture of the item they need recommendations for, the main page will allow the user to do that. Once uploaded, the model will look at the picture, identify the article of clothing and its color, and then look to recommend a certain number of outfits based on the user's desire. When the user sees a recommendation they like, they will be able to like it and that will save to their database (based on their password and username) where they look back and see that recommendation. Through a later update, the user will also be able to search other users' favorites to see if they would like those outfit choices. Once checking their final recommendation, the user will be able to log out and dress with their newfound fashion choice.

5.2. Screen Images

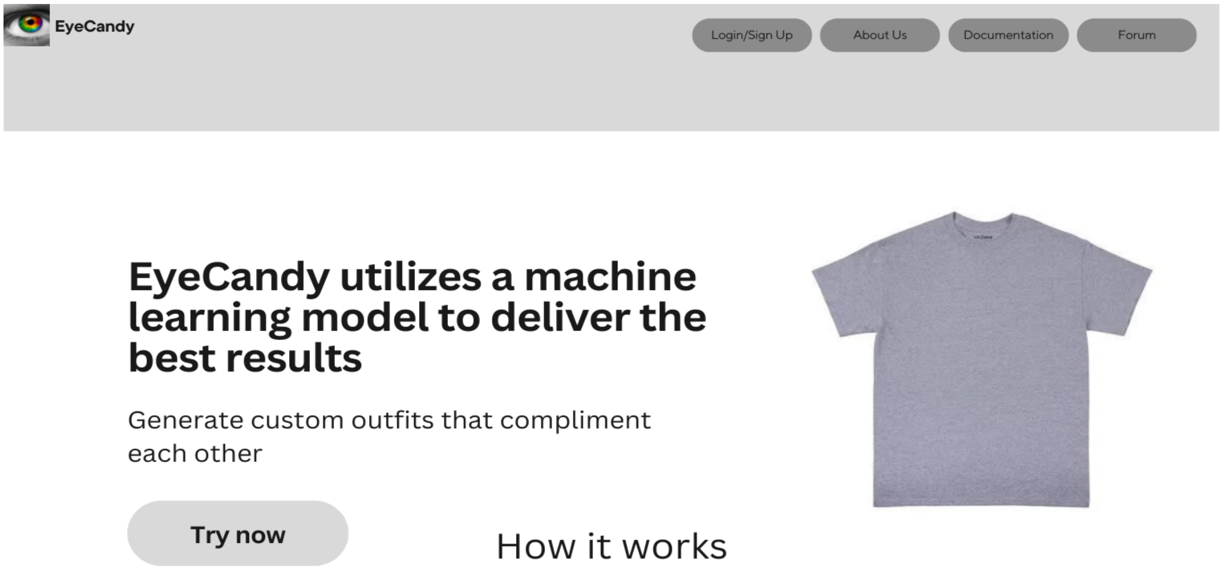
These are the temporary mock-up designs of the websites. **Note:** These designs are not the official design.

5.2.1. Main Homepage: when user first enters our website



Upload an article of clothing and our AI Program will find suitable pieces to match with it

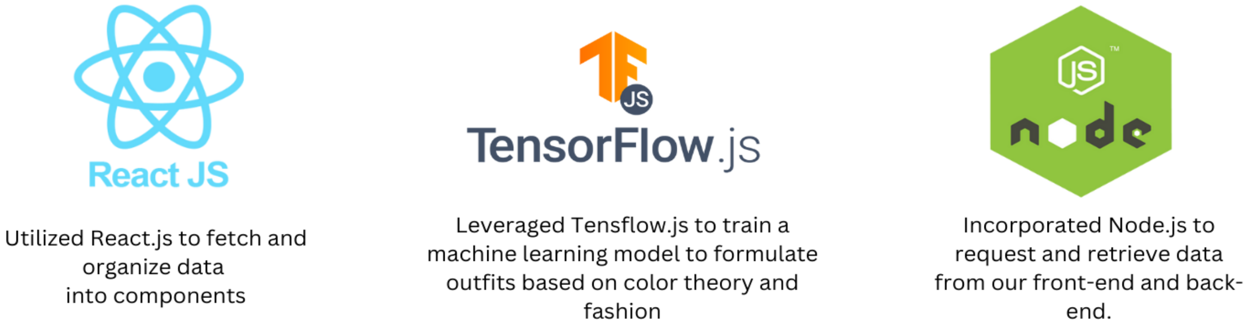
5.2.2. About Us Tab: brief description of what we do and how we do it



EyeCandy utilizes a machine learning model to deliver the best results

Generate custom outfits that compliment each other

5.2.3. About Us Tab (continued): as user scrolls down the page

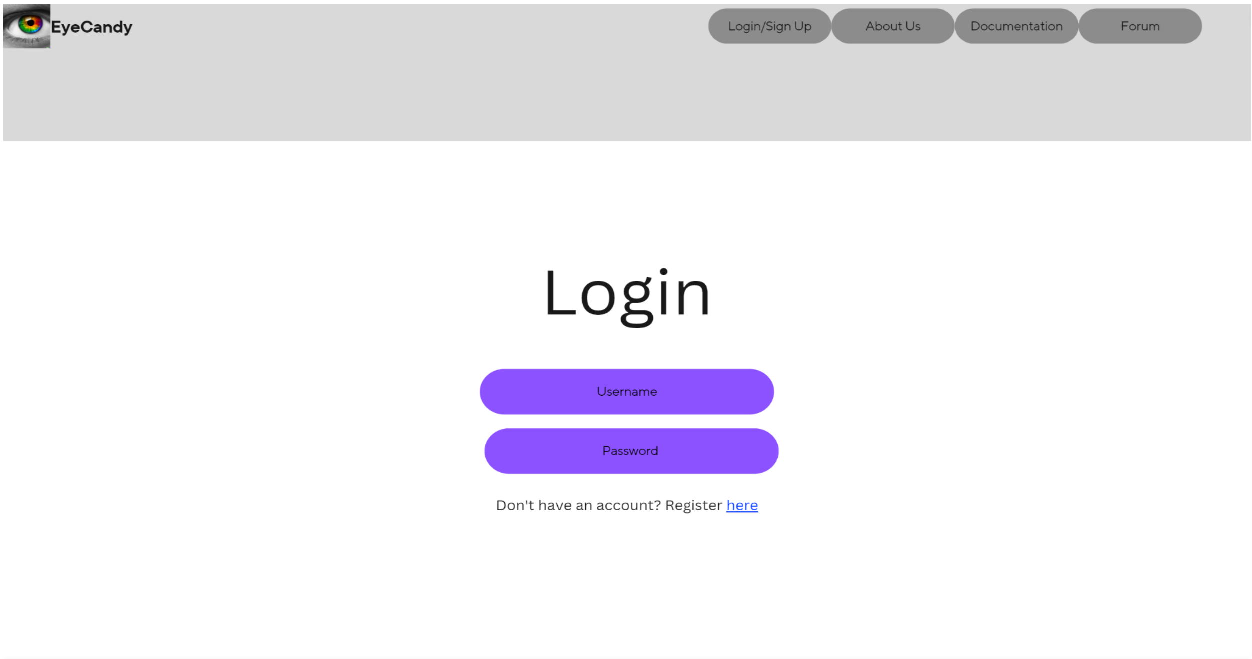


Utilized React.js to fetch and organize data into components

Leveraged Tensflow.js to train a machine learning model to formulate outfits based on color theory and fashion

Incorporated Node.js to request and retrieve data from our front-end and back-end.

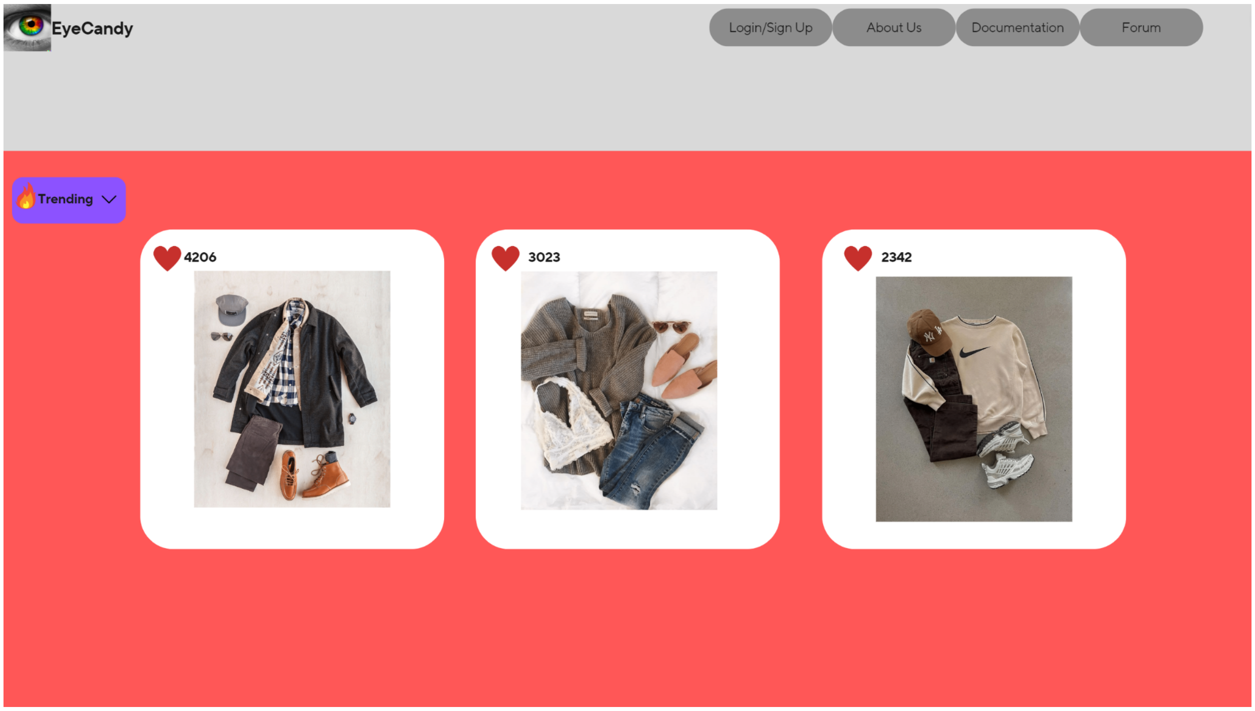
5.2.4. Login Page: To assess user profile or preferences user needs to sign in or sign up



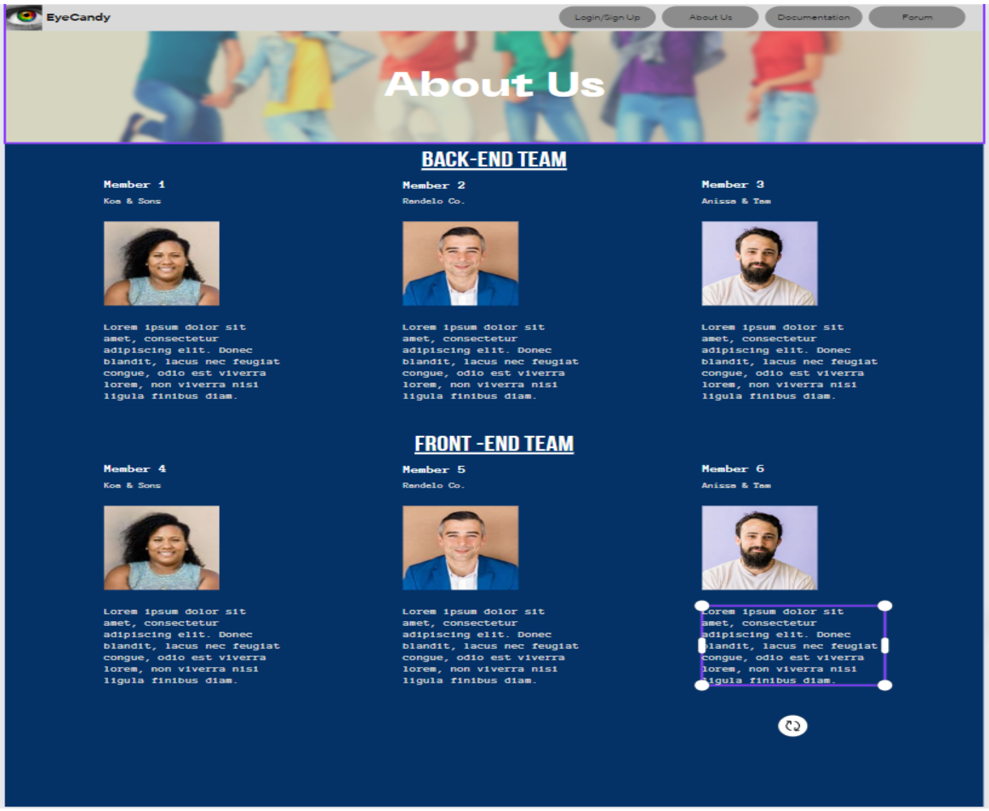
Login

Don't have an account? Register here

5.2.5. Forum Tab: users can see other recommendations from other members



5.2.6. About Us Tab (continued): as user scrolls down



About Us

BACK-END TEAM

Member 1
Koa & Sons

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec blandit, lacus nec feugiat congue, odio est viverra lorem, non viverra nisi ligula finibus diam.

Member 2
Randelo Co.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec blandit, lacus nec feugiat congue, odio est viverra lorem, non viverra nisi ligula finibus diam.

Member 3
Anissa & Tam

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec blandit, lacus nec feugiat congue, odio est viverra lorem, non viverra nisi ligula finibus diam.

FRONT -END TEAM

Member 4
Koa & Sons

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec blandit, lacus nec feugiat congue, odio est viverra lorem, non viverra nisi ligula finibus diam.

Member 5
Randelo Co.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec blandit, lacus nec feugiat congue, odio est viverra lorem, non viverra nisi ligula finibus diam.

Member 6
Anissa & Tam

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec blandit, lacus nec feugiat congue, odio est viverra lorem, non viverra nisi ligula finibus diam.

5.3. Screen Objects and Actions

Developer descriptions and their readability will be straight to the point and concise, giving a brief overview of each developer and their key focus.

The login page and its associated inputs (username and password) will allow the user to enter the website as themselves. It will be a simple and intuitive approach to logging in with only one place to do so. The upload of images will be encased in a center location of the page and it will allow the user to upload an image of the article of clothing they want to lead their recommendation. The nav bar illustrates the users ability to select through the application into other topics. This navbar object will allow the user to see things like an about page, recommendations page, feature page, and more. As this is a user application based on recommendations and even social media aspects, the user will be able to see other user features on a feature page to see if their style would match with others or if they'd like to try on different recommended outfits.

6. Requirements Matrix

Requirement ID	Requirement Description	Design Component	Test Case Number
GR_SRS_001	This web application shall include a homepage for basic user navigation	Human User Interface	5.2.1
GR_SRS_002	This web application shall provide the user with multiple options to click through	Human User Interface	5.2.1
GR_SRS_003	This web application shall grant the user the ability to select through different pages needed	Human User Interface	5.2.1
GR_SRS_004	This web application shall give users the freedom to look at multiple outfits to choose their look with our recommendation	Human User Interface	5.2.5
GR_SRS_005	This web application shall give the users the ability to search up other users on the website	Human User Interface	5.2.1
GR_SRS_006	This web application shall allow users to access the contact page to directly talk to the team	Human User Interface	5.2.6
GR_SRS_007	This web application shall help expand a users understanding of self fashion through UI aids such as hints	Human User Interface	5.2.2
UR_SRS_001	This program shall have a navigation bar for users to navigate through the website.	Human User Interface	5.2.1
UR_SRS_002	This program shall provide users the ability to upvote other user generated outfits.	Human User Interface	5.2.5

UR_SRS_003	This program shall provide a forum for users to input their article of clothing they'd like to base their outfit on	Human User Interface	5.2.5
UR_SRS_004	This program shall be able to generate a fashionable outfit based on input and preferences.	Human User Interface	5.2.5
UR_SRS_005	This program shall give users the ability to select preferences for their generated outfits.	Human User Interface	5.2.2
UR_SRS_006	This program shall have a main page where the fashion outfits are generated	Human User Interface	5.2.1
UR_SRS_007	This program shall have a contact page on the web application	Human User Interface	5.2.6
UR_SRS_008	This program shall have an about us section on the webpage	Human User Interface	5.2.6
UR_SRS_009	This program shall allow the user to create a personalized profile to save their creations	Human User Interface	5.2.5
FR_SRS_001	The web application shall be able to reroute users to different webpages	Human User Interface	5.2
FR_SRS_002	A website landing page should be included in this web application formatted with various remote buttons and other interactive buttons.	Human User Interface	5.2.1
FR_SRS_003	Website should be able to be logged into to keep user data safe	Human User Interface	5.2.4
FR_SRS_004	The web application shall have a home page where	Human User Interface	5.2.1

	a user interface is displayed to call the backend to retrieve the generated fashion fit via react components.		
FR_SRS_005	The web application shall have an about us page displaying our Linkedin and Github portfolios	Human User Interface	5.2.6
FR_SRS_006	The web application shall have a resources page showing the documentation and the supplementary API's we have integrated.	N/A	N/A
FR_SRS_007	The web application shall have a forum page where the generated outfits can be displayed in a forum style format	Human User Interface	5.2.5
FR_SRS_008	The web application shall have the ability to display the most upvoted outfits	Human User Interface	5.2.5
FR_SRS_009	The web application shall have a login and registration page	Human User Interface	5.2.4
FR_SRS_010	The web application shall include include various CTA's (primary and secondary) throughout the page, especially within landing page	N/A	N/A
FR_SRS_011	The website shall be responsive to multiple screen types and platforms	N/A	N/A
BR_SRS_001	This program shall have a database in order to have user profiles	System Architecture	3.3.3
BR_SRS_002	This program shall have an API to call upon when running the clothing	System Architecture	3.3.3

	recognition.		
BR_SRS_003	This program shall save new user profiles to the database to recall profile	System Architecture	3.3.3
BR_SRS_004	This program shall save user preferences into user profile in the database to recall information within profile	Component Design/Detailed Design	4.3
BR_SRS_005	This program shall authenticate login with the database to allow user to access profile	Component Design/Detailed Design	4.3
BR_SRS_006	This program shall be able to supported by the latest browsers	N/A	N/A
BR_SRS_007	This program shall be able to request access to photo library of user	Component Design/Detailed Design	4.3
BR_SRS_008	This program shall allow the user to choose a photo from their library	System Architecture	3.3.3