SNAPSIFT - IMAGE SEGREGATION APP

A Project Report

Submitted by

Anuksha Koul	**112103074**
Payal Madavi	**112103080**
Arsh Maknojia	**112103084**

of

TY (Computer Engineering)

Under the guidance of

Dr. Tanuja R. Pattanshetti

COEP Technological University

DEPARTMENT OF COMPUTER ENGINEERING

COEP Technological University

April, 2024

DEPARTMENT OF COMPUTER ENGINEERING

COEP Technological University

CERTIFICATE

Certified that this project, titled “SNAPSIFT - IMAGE SEGREGATION APP” has been successfully completed by

Anuksha Koul	**112103074**
Payal Madavi	**112103080**
Arsh Maknojia	**112103084**

and is approved for the fulfilment of the requirements of “Software Engineering Mini Project- Stage II”.

SIGNATURE

Dr. Tanuja R. Pattanshetti
Project Guide
Department of Computer Engineering
COEP Technological University,
Shivajinagar, Pune - 5.

Abstract

In today's digital age, the exponential growth of digital photography has led to vast collections of personal and professional images stored across various devices and platforms. However, managing and organizing these images manually can be time-consuming and challenging. To address this problem, we propose the development of an Image Segregation Web App, an innovative solution designed to automate the organization of images based on individual recognition.

The primary objective of the Image Segregation Web App is to simplify the process of image management by leveraging cutting-edge techniques, specifically python libraries like cv2 and face recognition, to automatically categorize and segregate uploaded images into individual folders corresponding to recognized individuals. This automation not only reduces the burden on users to manually sort and label images but also enhances the accessibility and organization of personal image collections.

The project scope encompasses the design, development, and deployment of the Image Segregation Web App, focusing on the aforementioned components to deliver a functional and user-friendly application. The app will target a wide range of users, including individuals with large personal image collections, photography enthusiasts, and professionals seeking efficient image organization solutions

In summary, the Image Segregation Web App represents a promising innovation in image management, leveraging python libraries and web technologies to automate and enhance the organization of digital image collections. This abstract provides a comprehensive overview of the project's objectives, components, scope, and expected outcomes. Feel free to adjust or expand upon this abstract based on specific project requirements and goals.

Contents

Chapter 1

Synopsis

1.1 Project Title

Image Segregation Web App

1.2 Internal Guide

Dr. Tanuja R. Pattanshetti

1.3 Problem Statement

In the era of digital photography, individuals often face challenges in managing and organizing large collections of images stored across devices and platforms. Manual sorting and labeling of images can be time-consuming and inefficient. The need for automated image organization solutions is evident to enhance accessibility and usability of personal image libraries.

Chapter 2

Problem Definition and Scope

2.1 Problem Definition

The primary goal of the Image Segregation Web App project is to develop an automated solution for organizing and managing digital image collections. The specific objectives include:

- Implementing image analysis techniques to recognize individuals in uploaded images.
- Creating a user-friendly interface for uploading images, viewing segregated folders, and labeling images.
- Automating the process of organizing images into individual folders based on recognized individuals.
- Enhancing accessibility and usability of personal image libraries through automation.

2.2 Statement of Scope

The scope of the project includes:

- Developing a web-based application capable of analyzing uploaded images using python libraries cv2 and face-recognition.

- Automatically organizing images into individual folders based on recognized individuals.
- Providing user interactions through a responsive and intuitive user interface.
- Supporting basic image management functionalities such as uploading, viewing, labeling, and organizing images.

2.3 Software Context

The Image Segregation Web App will be developed as a web application using modern web technologies. It will leverage python libraries and frameworks for image analysis and recognition.

2.4 Major Constraints

Key constraints for the project include:

- Time constraints: The project timeline must accommodate design, development, testing, and deployment phases within a specified timeframe.
- Resource constraints: Availability of hardware, software, and expertise required for developing and deploying the application.
- Accuracy constraints: The image analysis module must achieve a satisfactory level of accuracy in recognizing individuals to ensure reliable folder segregation.

2.5 Outcome

The expected outcome of the project is a functional and user-friendly Image Segregation Web App that automates image organization based on individual recognition. The app will enhance user experience and efficiency in managing personal image collections.

2.6 Applications

The Image Segregation Web App will benefit individuals with large image collections, photography enthusiasts, and professionals needing efficient image management solutions. It can be used for personal image organization, event management, and digital asset management.

2.7 Software Resources Required

The development of the Image Segregation Web App will require the following software resources:

- Programming languages: JavaScript, Python
- Web frameworks: html,css,js (frontend), Flask (backend)
- Machine learning libraries: face-recognition, OpenCV
- Integrated development environment (IDE): Visual Studio Code
- Version control system: Git

Chapter 3

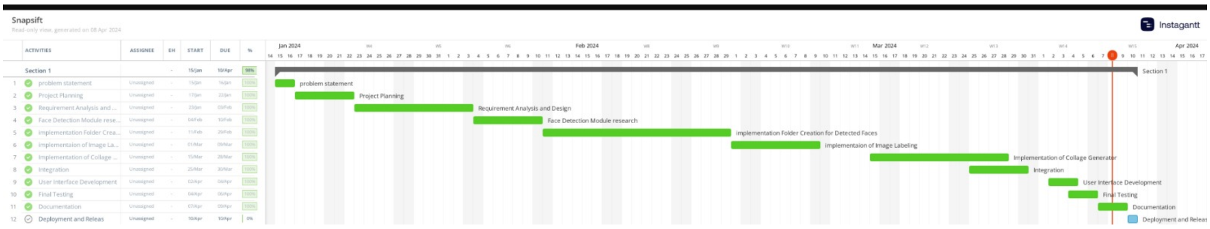
Project Plan

3.1 Project Schedule

The project schedule outlines the timeline and milestones for the development of the Image Segregation Web App.

3.1.1 Gantt Chart

Below is the Gantt chart illustrating the project timeline:



Snapsift

Read-only view, generated on 08 Apr 2024

	ACTIVITIES	ASSIGNEE	EH	START	DUE	%
	Section 1		-	15/Jan	10/Apr	98%
1	problem statement	Unassigned	-	15/Jan	16/Jan	100%
2	Project Planning	Unassigned	-	17/Jan	22/Jan	100%
3	Requirement Analysis and ...	Unassigned	-	23/Jan	03/Feb	100%
4	Face Detection Module rese...	Unassigned	-	04/Feb	10/Feb	100%
5	implementation Folder Crea...	Unassigned	-	11/Feb	29/Feb	100%
6	implementaion of Image La...	Unassigned	-	01/Mar	09/Mar	100%
7	Implementation of Collage ...	Unassigned	-	15/Mar	28/Mar	100%
8	Integration	Unassigned	-	25/Mar	30/Mar	100%
9	User Interface Development	Unassigned	-	02/Apr	04/Apr	100%
10	Final Testing	Unassigned	-	04/Apr	06/Apr	100%
11	Documentation	Unassigned	-	07/Apr	09/Apr	100%
12	Deployment and Releas	Unassigned	-	10/Apr	10/Apr	0%

Instagantt
Jan 2024
Feb 2024
Mar 2024
Apr 2024
Section 1
problem statement
Project Planning
Requirement Analysis and Design
Face Detection Module research
implementation Folder Creation for Detected Faces
implementaion of Image Labeling
Implementation of Collage Generator
Integration
User Interface Development
Final Testing
Documentation
Deployment and Releas

Figure 3.1: Project Gantt Chart

Chapter 4

Software Requirements Specification

4.1 Introduction

The Software Requirements Specification (SRS) document outlines the detailed requirements and specifications for the development of the Image Segregation Web App. This section provides an overview of use cases and functional models.

4.1.1 Use-Cases

The use cases describe the interactions between users and the system. Key use cases for the Image Segregation Web App include:

- **Upload Image:** Users can upload images to the web app for analysis and organization.
- **View Segregated Folders:** Users can view automatically generated folders containing segregated images based on recognized individuals.
- **Label Images:** Users have the option to manually label images for better organization and categorization.

4.1.2 Use Case View

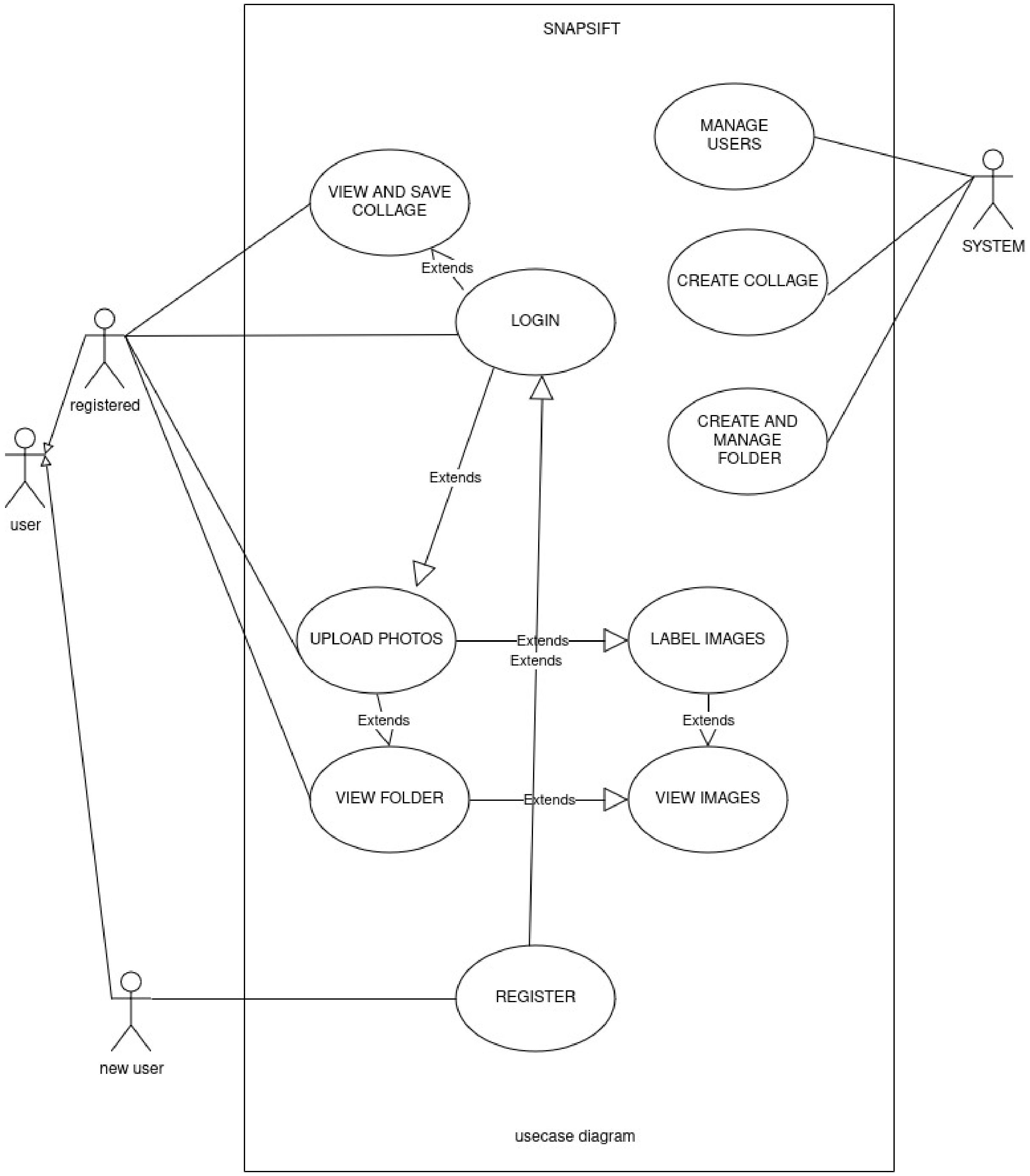


Figure 4.1: Use Case

4.2 Data Model and Description

4.2.1 Data Objects and Relationships

The data model defines the structure of data objects and their relationships within the Image Segregation Web App. Key data objects include:

- **Image:** Represents uploaded images for analysis and organization.
- **User:** Represents users interacting with the app, including preferences and metadata.
- **Folder:** Represents folders containing segregated images based on recognized individuals.

4.3 Functional Model and Description

4.3.1 Description of Functions

The functional model describes the specific functions and capabilities of the Image Segregation Web App, including:

- Image analysis and recognition using python libraries and algorithms.
- Automatic organization of images into individual folders based on recognized individuals.
- User interface functionalities for image upload, folder display, and manual labeling.

4.3.2 Data Flow Diagram

The data flow diagram illustrates the flow of data and information within the app, depicting the interactions between users, data objects, and system components.

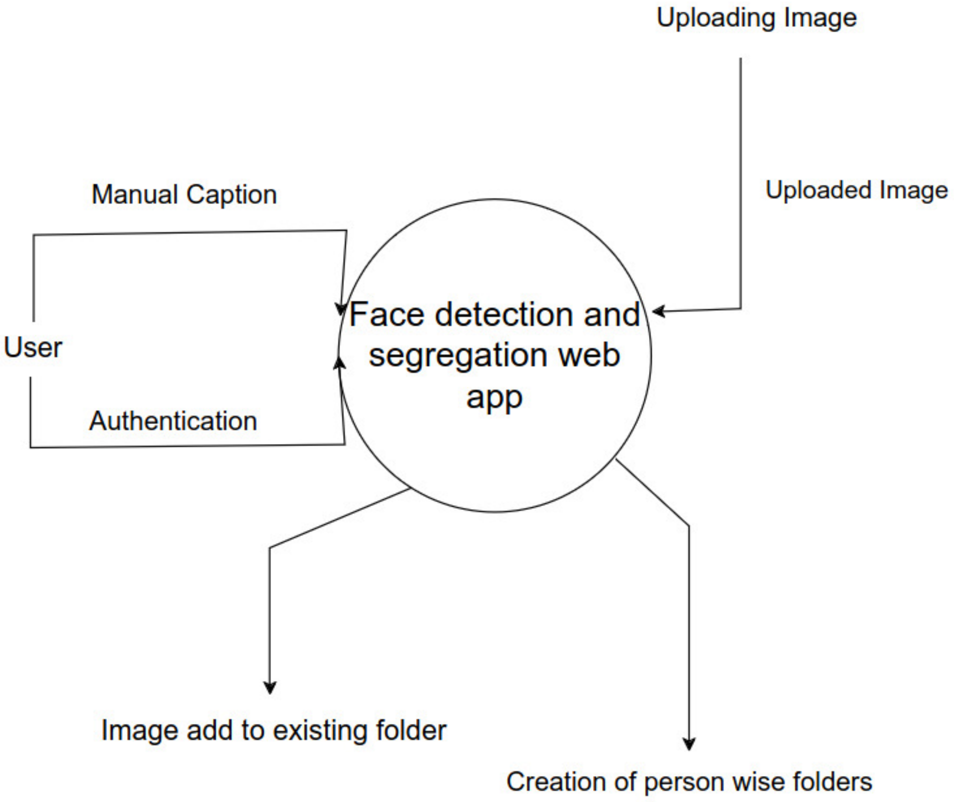

Uploading Image
Uploaded Image
Manual Caption
User
Authentication
Face detection and segregation web app
Image add to existing folder
Creation of person wise folders

Figure 4.2: DFD Level 0

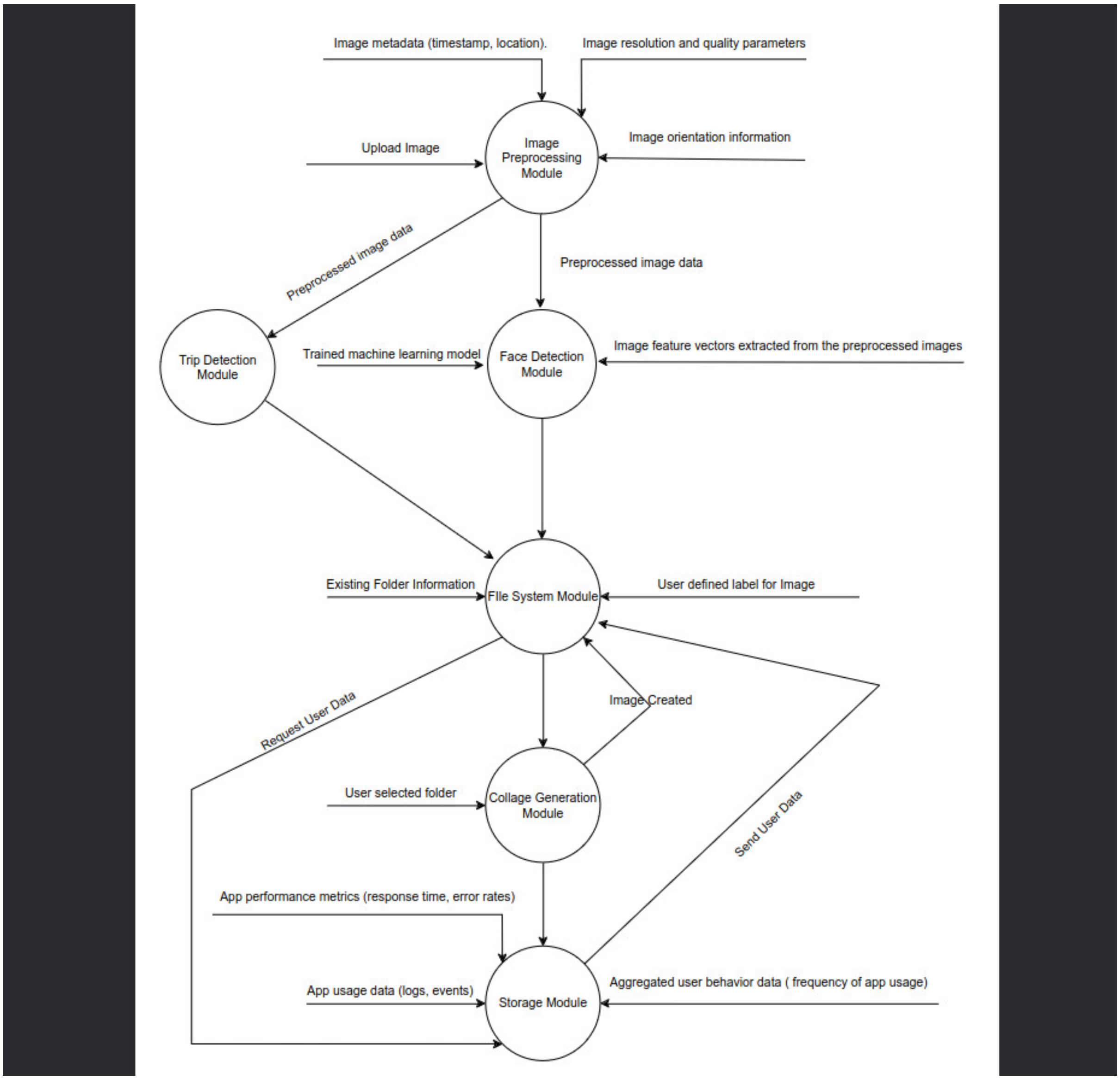


Image metadata (timestamp, location).
Image resolution and quality parameters
Upload Image
Image Preprocessing Module
Image orientation information
Preprocessed image data
Preprocessed image data
Trip Detection Module
Trained machine learning model
Face Detection Module
Image feature vectors extracted from the preprocessed images
Existing Folder Information
FIle System Module
User defined label for Image
Request User Data
Image Created
User selected folder
Collage Generation Module
Send User Data
App performance metrics (response time, error rates)
App usage data (logs, events)
Storage Module
Aggregated user behavior data (frequency of app usage)

Figure 4.3: DFD Level 1

4.3.3 Activity Diagram

The activity diagram visually represents the workflow and sequence of activities involved in key app functionalities, such as image processing, folder creation, and user interactions.

4.3.4 Non-Functional Requirements

Non-functional requirements specify constraints and quality attributes of the app, including performance, reliability, usability, and scalability.

4.3.5 Design Constraints

Design constraints outline limitations and considerations that may impact the app's design and implementation, such as technology choices, resource constraints, and compatibility requirements.

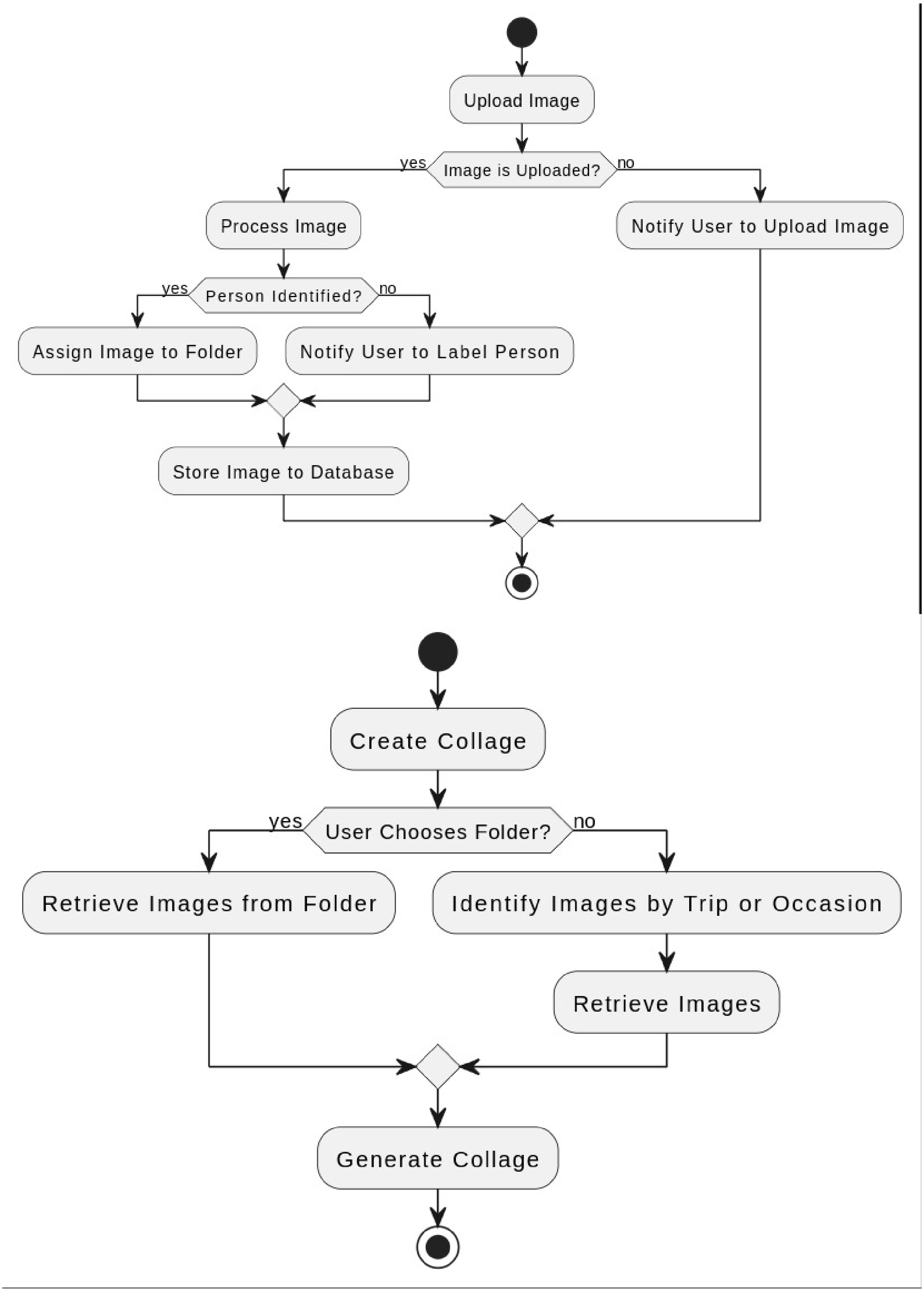


Figure 4.4: Activity Diagram

Chapter 5

Detailed Design Document

5.1 Component Design

5.1.1 Class Diagram

The class diagram illustrates the static structure of the Image Segregation Web App, showing classes, attributes, and relationships.

5.1.2 Sequence Diagram

The sequence diagram depicts the dynamic interactions between objects and components during specific scenarios, such as image upload, analysis, and folder creation.

5.1.3 Component Diagram

The component diagram illustrates the high-level components and their interactions within the Image Segregation Web App, including frontend, backend, and external services.

5.1.4 Deployment Diagram

The deployment diagram visualizes the deployment architecture of the app, showing the distribution of components across physical or virtual servers and devices.The deployment

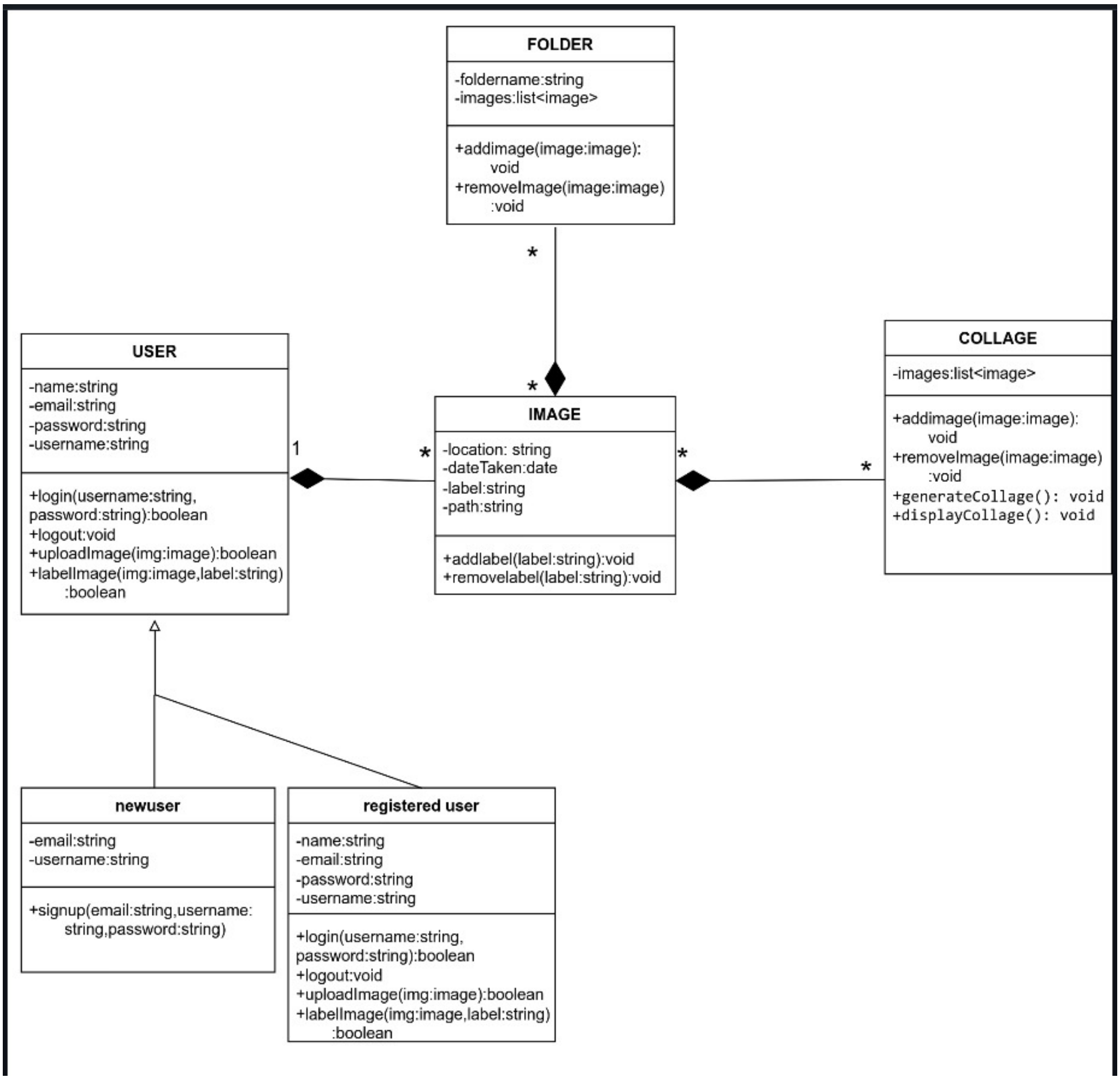

FOLDER
-foldername:string
-images:list<image>
+addimage(image:image): void
+removeImage(image:image) :void
*
*
USER
-name:string
-email:string
-password:string
-username:string
+login(username:string, password:string):boolean
+logout:void
+uploadImage(img:image):boolean
+labelImage(img:image,label:string) :boolean
1
*
IMAGE
-location: string
-dateTaken:date
-label:string
-path:string
+addlabel(label:string):void
+removelabel(label:string):void
*
*
COLLAGE
-images:list<image>
+addimage(image:image): void
+removeImage(image:image) :void
+generateCollage(): void
+displayCollage(): void
newuser
-email:string
-username:string
+signup(email:string,username: string,password:string)
registered user
-name:string
-email:string
-password:string
-username:string
+login(username:string, password:string):boolean
+logout:void
+uploadImage(img:image):boolean
+labelImage(img:image,label:string) :boolean

Figure 5.1: Class

diagram visualizes the deployment architecture of the app, showing the distribution of components across physical or virtual servers and devices.

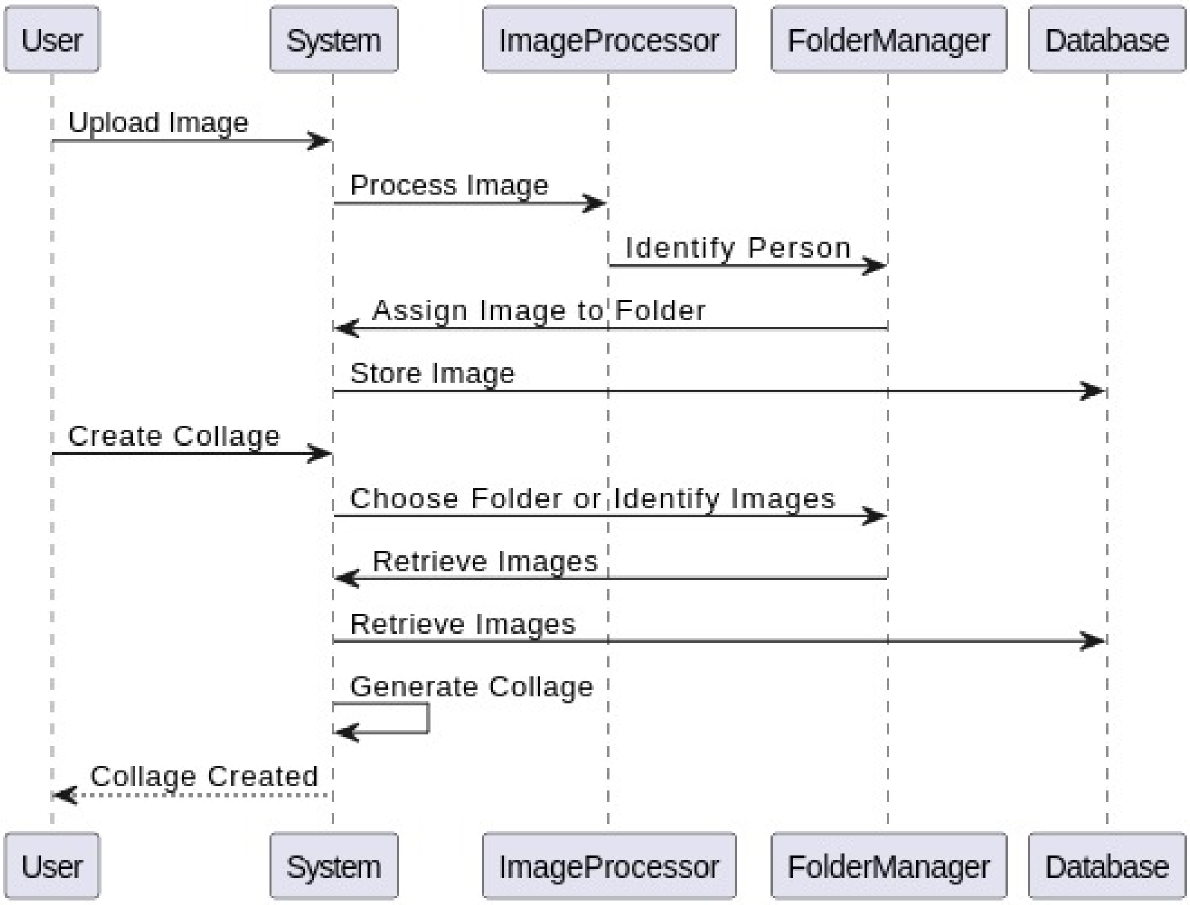


Figure 5.2: Sequence Diagram

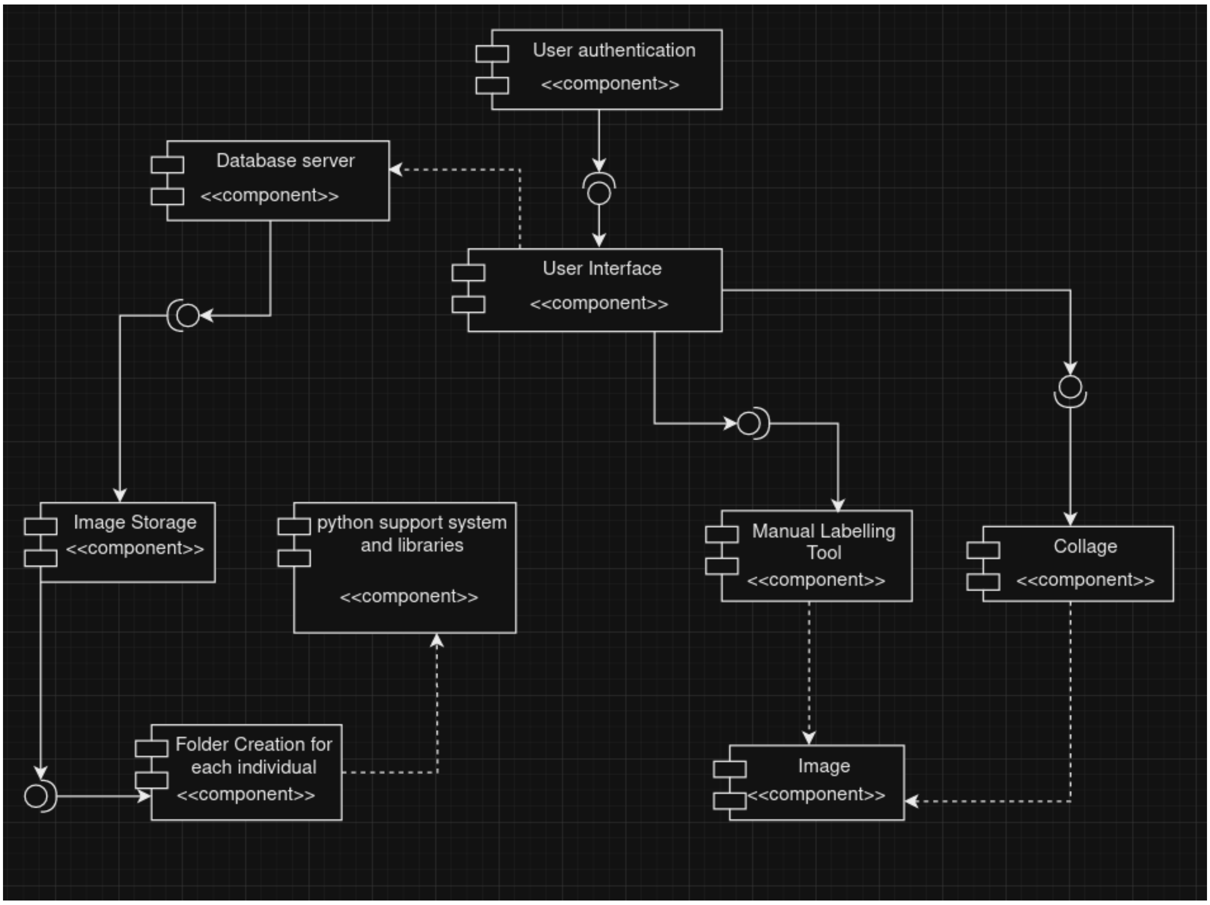


Figure 5.3: Component Diagram

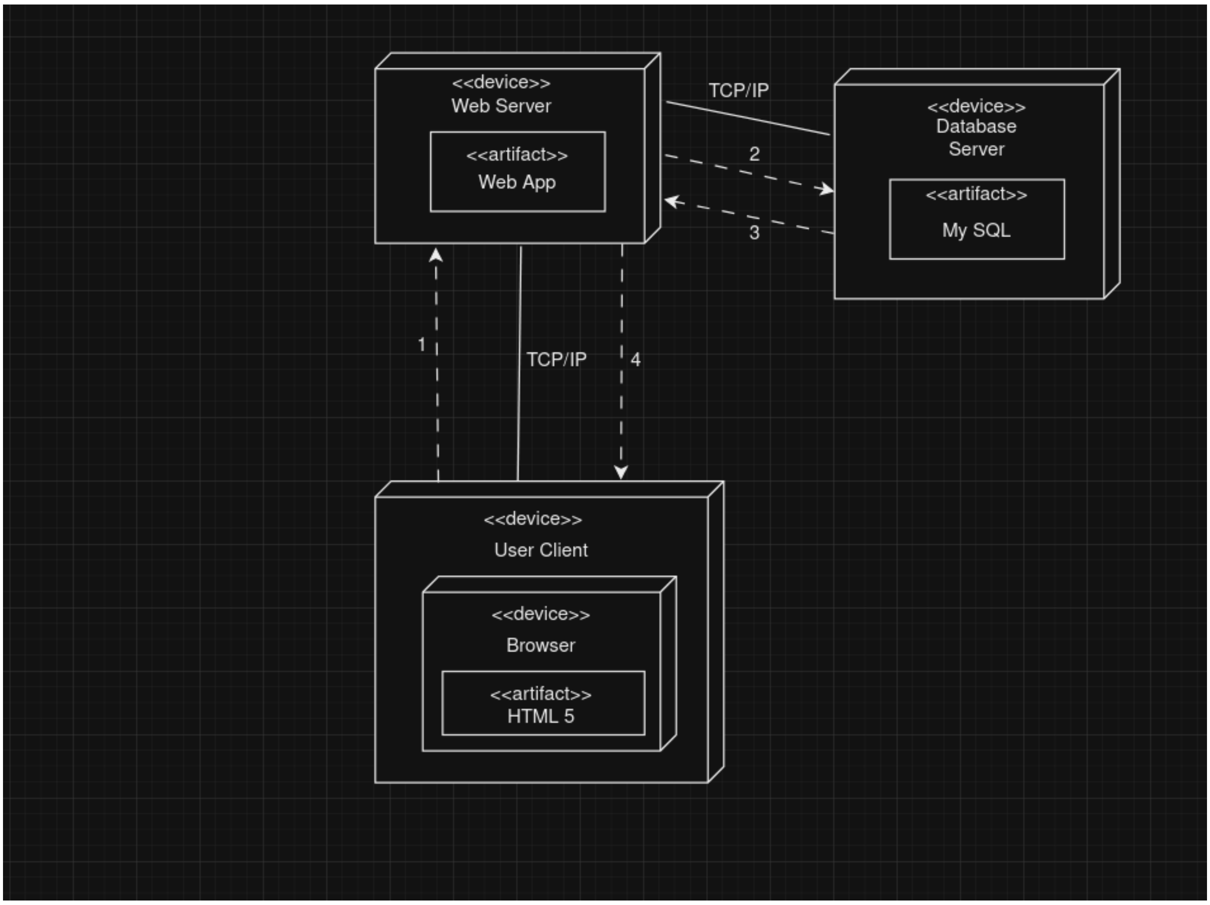


Figure 5.4: Deployment Diagram

5.2 Navigation Flow

The navigation flow defines the sequence of user interactions and navigation paths within the app, including:

- Uploading images
- Viewing segregated folders
- Labeling images
- Navigating between different app sections

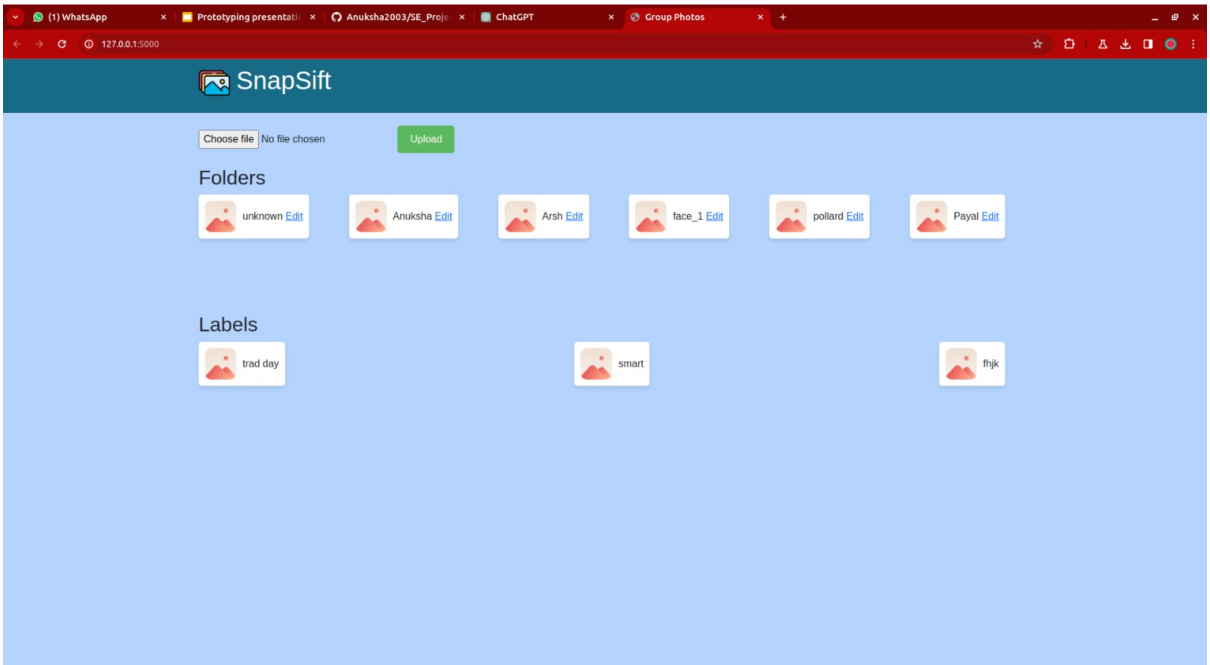


Figure 5.5: Main page

The navigation flow ensures a seamless and intuitive user experience throughout the app.

Chapter 6

Summary and Conclusion

The development of the Image Segregation Web App represents a significant advancement in automating the organization and management of digital image collections. This project aimed to address the challenges associated with manual image sorting and labeling by leveraging machine learning techniques and modern web technologies.

Throughout the project lifecycle, several key objectives were achieved:

- **Automated Image Analysis:** Implemented a robust image analysis module using python libraries to recognize individuals in uploaded images.
- **Folder Segregation:** Developed a folder segregation mechanism to automatically organize images into individual folders based on recognized individuals, enhancing accessibility and organization.
- **User Interface Design:** Created an intuitive user interface that allows users to upload images, view segregated folders, and manually label images for better categorization.
- **Backend Development:** Implemented backend components for data management, storage, and retrieval to support the app's functionalities.

- **Testing and Validation:** Conducted rigorous testing and validation to ensure the app's performance, accuracy, and reliability meet high standards.

The successful completion of the Image Segregation Web App project contributes to the field of digital image management by providing a practical and efficient solution for individuals and professionals dealing with large image collections.

In conclusion, the Image Segregation Web App project has demonstrated the feasibility and effectiveness of using machine learning and web technologies to automate image organization. The app's deployment offers users an innovative tool to streamline image management processes and enhance user experience.

Moving forward, future enhancements and iterations of the app could include:

- Integration with cloud storage services for seamless image synchronization.
- Enhanced image recognition capabilities for improved accuracy and efficiency.
- Support for collaborative image organization features, allowing multiple users to contribute to folder creation and labeling.

Overall, the Image Segregation Web App project marks a significant milestone in digital image management, paving the way for innovative solutions to address the evolving challenges of image organization in the digital age.

Bibliography

- **OpenCV Python Package**: `https://pypi.org/project/opencv-python/` [Online; accessed April 2024]
- **Face Recognition Python Package**: `https://pypi.org/project/face-recognition/` [Online; accessed April 2024]