Image Annotation Tool

By

Aleena Zainab 181081

Rameen Shakeel 181137

Bachelor of Science in Computer Science

Supervisor

Dr. Anabia Sohail

Department of Computer Science

Air University, Islamabad

2022

Type (Nature of project)	[✓] Development [ ] R&D
Area of specialization	[✓] WebApp [ ] Mobile App [ ] AI based [ ] Embedded System
FYP ID	CS-SP-21-23

Project Group Members

Sr.#	Reg. #	Student Name	Email ID	*Signature
(i)	181081	Aleena Zainab	181081@students.au.edu.pk	
(ii)	181137	Rameen Shakeel	181137@students.au.edu.pk	

Plagiarism Certificate

This is to certify that, I **Rameen Shakeel** S/D of **Muhammad Shakeel Wahid**, group leader of FYP under registration no **181137** at Computer Sciences Department, Air University. I declare that my FYP report is checked by my supervisor.

Date:_____

Name of Group Leader: Rameen Shakeel Signature:_____

Name of Supervisor: Dr.Anabia Sohail Signature:_____
Designation: Lecturer

HoD: Dr. Awais Ahmad Signature: _____

Image Annotation Tool

Change Record

Author(s)	Version	Date	Notes	Supervisor's Signature
Rameen Shakeel and Aleena Zainab	1.0	14th May 2022		

APPROVAL

PROJECT SUPERVISOR

Name: Dr. Anabia Sohail

Date: 14th May 2022 Signature:_____

PROJECT MANAGER

Date:_____ Signature:_____

CHAIR DEPARTMENT

Date:_____ Signature:_____

Dedication

I dedicate this report to my loving parents whose prayers, affection and support are always a source of encouragement for me to reach at this destination and a humble icon for others in future. My parents and teachers who give me real eye that help me to lead myself and others in the dark and cruel world.

Acknowledgements

In the Name of Allah, the most gracious, the most merciful. First and foremost, we would like to thank Allah Almighty for giving us the strength, knowledge, ability and opportunity to undertake this research study and to persevere and complete it satisfactorily. Without His blessings, this achievement would not have been possible. Secondly, we are extremely thankful to our supervisor Dr. Anabia Sohail for dedicating her time, support, guidance, ideas and patience. Her proficiency, dedication, honesty and continuous encouragement made it possible for us to complete this project successfully and on time. We are also thankful for her invaluably constructive criticism and friendly advice during the project work. To our respective partners, we simply could not have done this without each other. To our parents, who have supported us throughout the course of our project in every way possible. Our gratitude also goes to the whole CS faculty. We are grateful for their patience and feedback. We wholeheartedly thank the staff of Air University for their assistance. Finally, many thanks to all participants that took part in the study and enabled this research to be possible.

Executive Summary

Image Annotation tool is basically an open source web based application. It is a user friendly tool. In this project, different techniques of Image processing i.e. segmentation and detection are used to develop annotation tool. The tool developed in this work includes python based implementation of detection and segmentation (Thresholding). It includes an image that is going to be uploaded on a tool. Then it gives the option of segmentation and detection to generate the annotation of the dataset. The project includes dataset of more than 3000 Medical images. It generates the annotations in JSON format as the output. The web based Image Annotation tool is beneficial for unlabeled medical dataset. There is no technology transfer or diffusion approach to our project since this is a Free open source web-based tool.

Table of Contents

Image Annotation Tool i
Aleena Zainab 181081 i
Rameen Shakeel 181137 i
Bachelor of Science in Computer Science i
Plagiarism Certificate iii
Dedication vii
Acknowledgements viii
Executive Summary ix
Table of Contents x
List of Figures xiii
List of Tables xiv
List of Abbreviations xv
Chapter 1 1
Introduction & Background 1
1.1. Background 2
1.2. Motivations and Challenges 3
1.3. Goals and Objectives 3
1.4. Literature Review/Existing Solutions 4
1.5. Proposed Solution 5
1.6. Project Plan 5
Work Breakdown Structure 5
1.6.1. 5
1.6.2. Roles & Responsibility Matrix 6
1.6.3. Gantt chart 7
1.7. Report Outline 7
Chapter 2 8
Software Requirement Specifications 8
2.1. Introduction 9
2.1.1. Purpose 9
2.1.2. Document Conventions 9
2.1.3. Intended Audience and Reading Suggestions 10
2.1.4. Product Scope 10
Overall Description 10
2.2. 10
2.2.1. Product Perspective 10
2.2.2. User Classes and Characteristics 10
Operating Environment 10
2.2.3. 10
2.2.4. Design and Implementation Constraints 11
2.2.5. Design and Implementation Constraints 11

2.2.6. Assumptions and Dependencies 11
2.3. External Interface Requirements 11
2.3.1. User Interfaces 11
2.3.2. Hardware Interfaces 14
2.3.3. Software Interfaces 14
2.3.4. Communications Interfaces 14
2.4. System Features 14
2.4.1. Segmentation 14
2.4.1.1. Description and Priority 14
2.4.1.2. Stimulus/Response Sequence 15
2.4.1.3. Functional Requirements 15
2.4.2. Detection 15
2.4.2.1. Description and Priority 15
2.4.2.2. Stimulus/Response Sequences 15
2.4.2.3. Functional Requirements 15
Nonfunctional Requirements 16
2.5 16
2.5.1. Performance Requirements 16
2.5.2. Safety Requirements 16
2.5.3. Security Requirements 16
2.5.4. Usability Requirements 16
2.5.5. Reliability Requirements 16
2.5.6. Maintainability/Supportability Requirements 16
2.5.7. Portability Requirements 16
2.5.8. Efficiency Requirements 16
Chapter 3 17
Use Case Analysis 17
3.1. Use Case Model 18
3.2. Use Cases Description 19
Chapter 4 24
System Design 24
4.1. Architecture Diagram 25
4.2. Domain Model 26
4.3. Entity Relationship Diagram with data dictionary 27
4.4. Class Diagram 28
4.5. Sequence / Collaboration Diagram 29
4.6. Activity Diagram 30
4.7. State Transition Diagram 31
4.8. Data Flow diagram 32
Chapter 5 33
Implementation 33
5.1. Important Flow Control/Pseudo codes 34
5.2. Components, Libraries, Web Services and stubs 38
5.3. Deployment Environment 38
5.4. Tools and Techniques 38
5.5. Best Practices / Coding Standards 39

5.6. Version Control 39
Chapter 6 40
Business Plan 40
Chapter 6: Business Plan 41
6.1 Business Description 41
6.2 Market Analysis & Strategy 41
6.3 Competitive Analysis 41
6.4 Products/Services Description 41
6.5 SWOT Analysis 42
Chapter 7 43
Testing & Evaluation 43
Chapter 7: Testing and Evaluation 44
7.1 Equivalence partitioning 44
7.2 Boundary value analysis 44
7.3 Data flow testing 44
7.4 Unit testing 45
7.5 Integration testing 46
7.6 Performance testing 46
7.7 Pressure Testing 46
Chapter 8 47
Conclusion & Future Enhancements 47
Chapter 8: Conclusion & Future Enhancements 48
8.1 Achievements and Improvements 48
8.2 Critical Review 48
8.3 Lessons Learnt 48
8.4 Future Enhancements/Recommendations 49
Appendices 50
Appendix A: Information / Promotional Material 51

List of Figures

1.1	Image Annotator	5
1.2	Gantt Chart	7
2.1	User Interface (1)	11
2.2	User Interface (2)	12
2.3	Global Thresholding	12
2.4	Fixed Square	13
2.5	Square on canvas' Image	13
3.1	Use Case for Image Annotator	18
3.2	Use Case Diagram for Login	19
3.3	Use case diagram for Upload an image	20
3.4	Use Case diagram for Segmentation	21
3.5	Use Case Diagram for Detection	22
3.6	Use Case Diagram for Save Annotated Images	23
4.1	Architecture Diagram	25
4.2	Domain Model Diagram	26
4.3	ER-Diagram	27
4.4	Class Diagram	28
4.5	Sequence Diagram	29
4.6	Activity Diagram	30
4.7	State transition diagram	31
4.8	Data Flow Diagram	32
5.1	Flow control	34
5.2	Front Page	35
5.3	Segmentation	35
5.4	Binary Thresholding	36
5.5	Otsu Thresholding	36
5.6	Detection	37
5.7	Square On Image	37
7.1	Boundary Value Analysis	44
7.2	Data Flow Testing (1)	45
7.3	Data Flow Testing (2)	45

List of Tables

1.1	Work Breakdown Structure	6
1.2	Roles and Responsibilities	6
3.1	User Login Function	19
3.2	User login Function: Use Case 1	19
3.3	Upload an Image Function	20
3.4	Upload an Image Function: Use Case 2	20
3.5	Segmentation Function	21
3.6	Segmentation Function: Use Case 3	21
3.7	Detection Function	22
3.8	Detection Function: Use Case 4	22
3.9	Save Annotated Image Function	23
3.10	Save Annotated Image Function: Use Case 5	23

List of Abbreviations

1.1	UML	Unified Model Notation
1.3	UI	User Interfaces
1.4	JSON	Java script Object Notation
5.1	VS	Visual Studio
5.3	IDE	Integrated Development Environment
5.3	ML	Machine Learning

Chapter 1

Introduction & Background

Chapter 1: Introduction

This chapter covers an introduction to the project including the context, a description of aims and objectives, a description of what has been achieved, contributions and the structure of the report.

1.1. Background

Annotation is the act of annotating something. Without the annotations, the diagram would be hard to understand. Image annotation is most commonly used to recognize objects and boundaries and to segment images for instance, meaning, or whole-image understanding. For each of these uses, it takes a significant amount of data to train, validate, and test a machine learning model to achieve the desired outcome. There are many commercially available software packages for viewing, editing, and labelling of large number of natural images but are not specifically tailored to deal with histopathological images. Besides, the available histopathology tools, do not allow user to view and annotate images in different modalities as well as generate variants of segmentation masks automatically.

Some of the tools are as follows:

Visual Geometry Group developed a web-based tool named as VGG Image Annotator to create annotations for three different modes of data (images, video and audio) [1]. This tool provides the segment annotation option for audios and video along with drawing of objects shapes in images. This software generates the annotations in the JSON and CSV formats. Torralba et al. presented a web-based annotation tool, LabelMe [2]. It allows to assign semantics to images by defining objects' spatial location along with label. SlideRunner [3] is developed for annotations in whole slide images. This software uses OpenSlide [4] as image loading backend. It allows user to perform annotations as a single point or periphery/boundary of objects. This tool targets gigapixel images but does not accommodate small image patches.

Label studio [5] is a web-based, open source labelling and annotation tool released in late 2019. This tool is simple yet powerful along with the support for multiple bounding box shapes.

Visual Object Tagging tool (VoTT) [6] is also an open source annotation and labelling tool for videos and images, which can be run locally. It can be used easily in any modern browser. VoTT provides the support for windows, Linux and OSX.

Computer vision annotation tool (CVAT) [7] was developed by Intel. CVAT is an open source, web-based annotation tool with support for both images and videos. It is a browser-based application and only works with Google Chrome browser.

1.2. Motivations and Challenges

There are many issues in the publicly provided histopathological datasets. Most often datasets contain information on the count of cells present in each patch. In contrast to this, information regarding cells localization i.e. the bounding box coordinates, and shape of segmentation masks are not provided. Thus, manually defining shape of an object in each sample under a microscope is tiresome and slow, which demands an alternate fast solution. It difficult to generate mask via a single image processing technique. Consequently, there is a need to develop a fast and generalized approach to generate boundary box and segmentation mask. In our work, we aimed to design an open source simple and easy to use tool for pathologists to perform annotations with minimum training.

1.3. Goals and Objectives

Goals:

The goals of the project are as follow:

- Open source web based
- Single transitions of image
- Annotation of medical datasets
- Tagging of multiple images
- User friendly

Objectives:

The objectives of the project are as follow:

- Annotate an image with minimum user involvement
- Perform 2 types of annotation:

1. Segmentation
2. Detection

1.4. Literature Review/Existing Solutions

There are some available tools for image annotations, wcehich allow users to draw bounding boxes or objects shapes in images.

Visual Geometry Group developed a web-based tool named as VGG Image Annotator to create annotations for three different modes of data (images, video and audio) [1]. This tool provides the segment annotation option for audios and video along with drawing of objects shapes in images. This software generates the annotations in the JSON and CSV formats.

Torralba et al. presented a web-based annotation tool, LabelMe [2]. It allows to assign semantics to images by defining objects' spatial location along with label.

SlideRunner [3] is developed for annotations in whole slide images. This software uses OpenSlide [4] as image loading backend. It allows user to perform annotations as a single point or periphery/boundary of objects. This tool targets gigapixel images but does not accommodate small image patches.

Label studio [5] is a web-based, open source labelling and annotation tool released in late 2019. This tool is simple yet powerful along with the support for multiple bounding box shapes.

Visual Object Tagging tool (VoTT) [6] is also an open source annotation and labelling tool for videos and images, which can be run locally. It can be used easily in any modern browser. VoTT provides the support for windows, Linux and OSX.

Computer vision annotation tool (CVAT) [7] was developed by Intel. CVAT is an open source, web-based annotation tool with support for both images and videos. It is a browser-based application and only works with Google Chrome browser.

However, most of the tools for labelling medical imaging datasets are either for commercial use or have high priced subscriptions. In our work, we aimed to design an open source simple and easy to use tool for pathologists to perform annotations with minimum training.

1.5. Proposed Solution

We developed a web based open source annotation tool to make the nature of tool more generic to other datasets and problems as well. It will be easily accessible and useable. At the end we annotate medical images of datasets using Image processing techniques of segmentation and detection. Our output annotations are saved in JSON format.

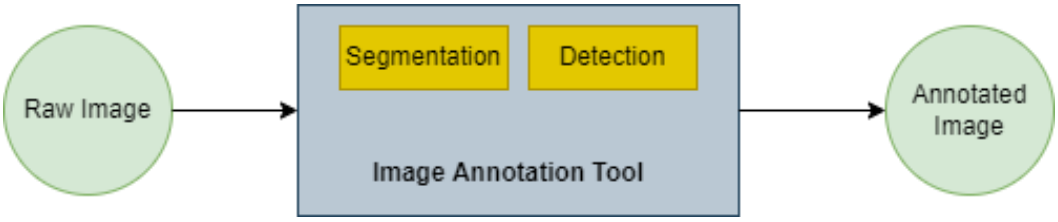


Figure 1.1: Image Annotator

Different tools generate different sets of data. The web based Image Annotation tool is beneficial for Medical use as it includes segmentation and detection. There is no technology transfer or diffusion approach to our project since this is a Free open source web-based tool.

1.6. Project Plan

A set of project milestones and deliverables are produced illustrating the project breakdown to create a management plan to adhere to throughout the project lifecycle, however there may be slight changes in each iteration as some requirements may require modification. The steps required in each iteration are outlined to meet each project deliverable. Produce risk management plan and a project plan displayed through the use of a Gantt chart showing the projects tasks for each semester.

1.6.1. Work Breakdown Structure

No.	Elapsed time from start(in months) of the project	Milestone	Deliverables

1.	3 months	Learning phase	Learning of web development tools,basic image processing techniques, segmentation, detection.
2.	6 months	Backend development	Backend start(segmentation)
3.	9 months	Frontend development	Frontend start(UI designs)
4.	12 months	ML model	Backend continue(remaining)
5.	15 months	Integration	Integration of backend and frontend
6.	18 months	Deployment	Deployment of application

Table 1.1: Work Breakdown structure

1.6.2. Roles & Responsibility Matrix

Responsibilities	**Role 1** – Aleena Zainab	**Role 2** – Rameen Shakeel
Project Proposal	√	√
Market Analysis	√	√
Frontend Designing	√	√
AI Model Training	√	√
FYP-1 presentation + Report	√	√
FYP-2 presentation + Report	√	√
FYP-3 presentation + Report	√	√

Table 1.2: Roles & Responsibilities

1.6.3. Gantt chart

ID	Task Name	Start	Finish	Pre-ce	3 Months	6 Months	9 Months	12 Months	15 Months	18 Months
1	Project Plan	1st May 2021	30th May 2021							
2	Project Discussion	1st June 2021	30th June 2021	1						
3	Documentation	1st July	30th August							
4	Prototype	1st September 2021	30th October 2021							
5	Coding	1st November 2021	30th December 2021	2						
6	Testing	1st January 2022	31st January 2022	1						
7	Debugging	1st February 2022	31st March 2022	3						
8	Deployment	1st April 2022	31st April 2022	1						
9	Submission Of Final Project	May 2022	May 2022	1						

Figure 1.2: Gantt chart

1.7. Report Outline

In the report outline the work we had done for completion of our project has been mentioned. The goals and objectives we require to fulfill for the final implementation of our application are also highlighted.

- Software Requirement Specifications
- Use case Analysis
- System Design
- Implementation
- Business Plan
- Test & Evaluation
- Conclusion & Future Enhancement

Chapter 2

Software Requirement Specifications

Chapter 2: Software Requirement Specifications

2.1. Introduction

Image Annotation is the tool of labeling or classifying an image using annotation tool, to show data features. In this project we are performing the techniques of Image segmentation (Thresholding) and detection.

2.1.1. Purpose

There are many issues in the publicly provided histopathological datasets. Information regarding Tumor localization i.e. the bounding box coordinates, and shape of segmentation masks are not provided. Manually defining shape of an object in each sample under a microscope is tiresome and slow, which demands an alternate fast solution. Consequently, there is a need to develop a fast and generalized approach to generate boundary box and segmentation mask. We developed an annotation tool for histopathological images. The developed tool is used to annotate images. In addition, it can also develop bounding boxes and annotation masks in standard JSON format.

2.1.2. Document Conventions

All the text in this document is 12pt Times New Roman.

- Chapter numbers are 48pt Times New Roman
- Chapter Names are 36pt Bold Times New Roman
- Section Titles are 16pt Bold Times New Roman
- Sub Section are 14pt Bold Times New Roman
- All Sections and subsections are numbered using X.X.X... format. Where X represents numbers.
- Document text spacing is 1.5.

2.1.3. Intended Audience and Reading Suggestions

This document is to target the computer scientists and everybody with an interest in image annotation, ranging from non-professional end-users that are annotating their personal digital photos to professionals working with digital pictures in medical image.

2.1.4. Product Scope

Image annotation is most commonly used to recognize objects and boundaries and to segment images for instance, meaning, or whole-image understanding. For each of these uses, it takes a significant amount of data to train, validate, and test a machine learning model to achieve the desired outcome.

2.2. Overall Description

2.2.1. Product Perspective

We developed an annotation tool for medical images. In addition, it can also develop bounding boxes and annotation masks in standard JSON format. Further we developed an open source web-based annotation tool. The tool is performing the Image processing techniques of Segmentation and Detection to the Brain Tumor dataset.

2.2.2. User Classes and Characteristics

Users of the software system include researchers, practicing engineers, and software developers. All users should be familiar with web technology. Users are knowledgeable of software-development processes or manufacturing processes but may not be knowledgeable of both. However, users should have a good understanding of the tasks, activities, and artifacts of either process in which they may be interested.

2.2.3. Operating Environment

This product will be available as a web application. Any browser that can be opened using windows, or any Operating system that supports browser can access this tool.

2.2.4. Design and Implementation Constraints

This project should have some important things to operate perfectly. Training and query image file format should be .jpg, .png, or others and not .txt, .doc, .pdf or others. Dataset images should be clear to obtain better results. It should not be blurry or corrupt.

2.2.5. Design and Implementation Constraints

This project should have some important things to operate perfectly. Training and query image file format should be .jpg, .png, or others and not .txt, .doc, .pdf or others. Dataset images should be clear to obtain better results. It should not be blurry or corrupt.

2.2.6. Assumptions and Dependencies

If the internet connection is down, then user is unable to interact with the tool.

2.3. External Interface Requirements

2.3.1. User Interfaces

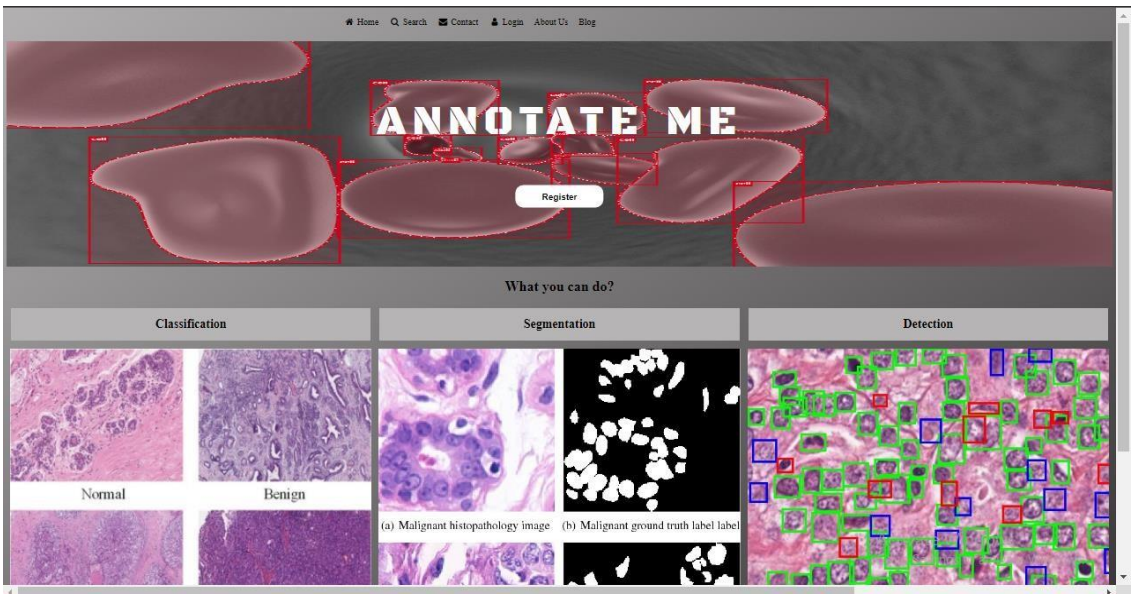


Figure 2.1: User interface (1)

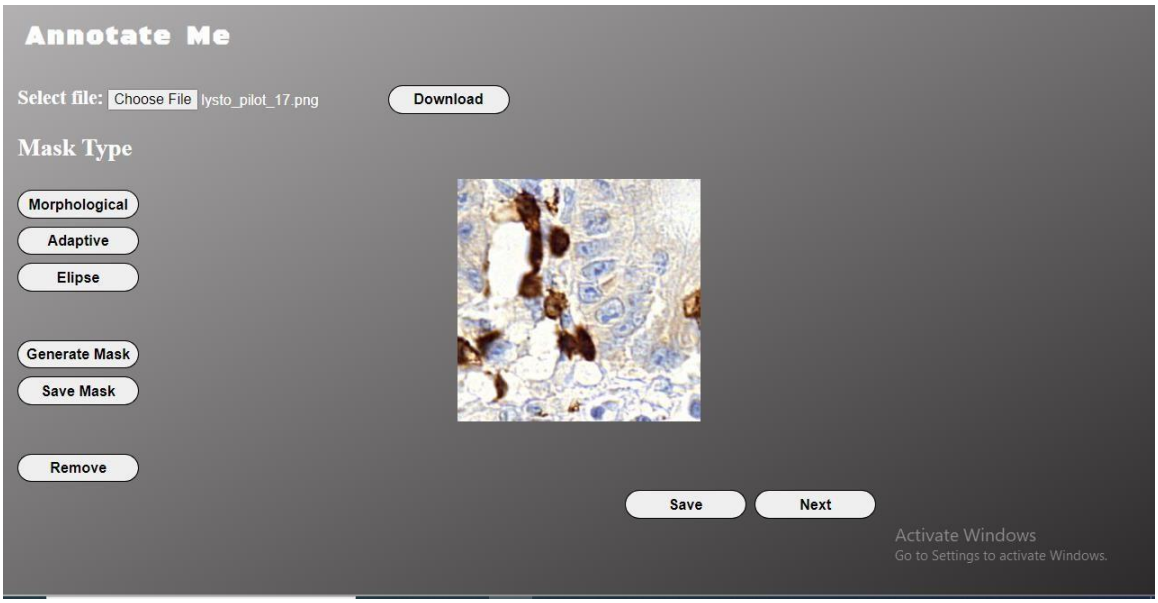


Figure 2.2: User Interface (2)

Segmentation:

1. Global Thresholding:

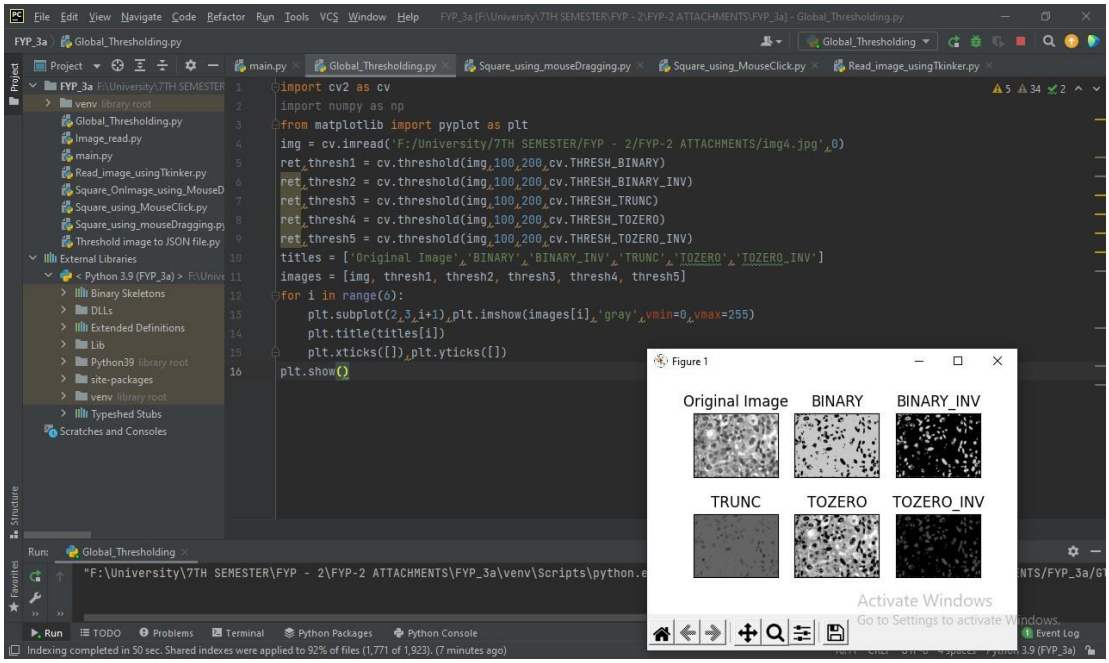


Figure 2.3: Global Thresholding

Detection:

1. Fixed square:

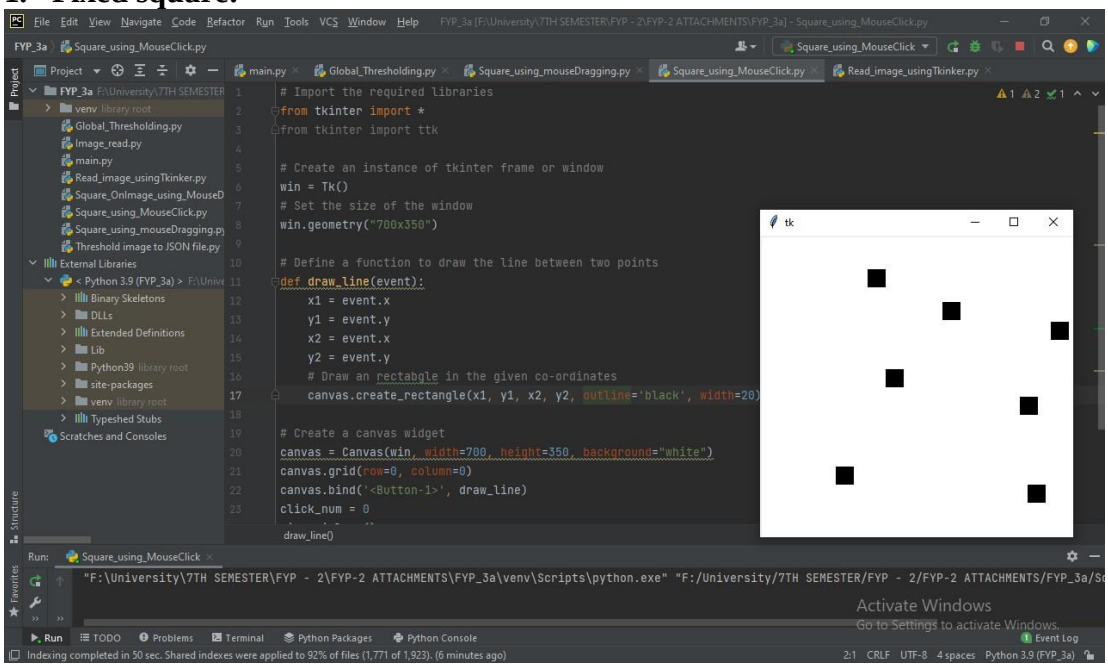


Figure 2.4: Fixed square

2. Square on Canvas' Image:

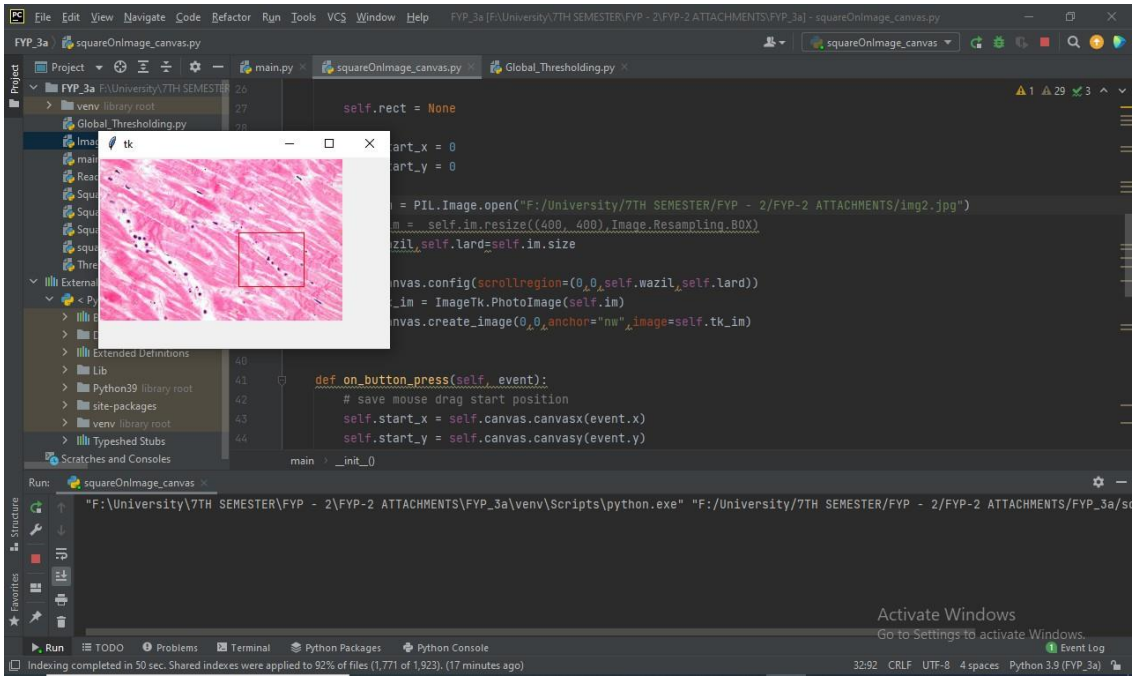


Figure 2.5: Square on Canvas' Image

2.3.2. Hardware Interfaces

There is no direct hardware interface specially for our system. This tool will run on a web server.

- Windows
- A browser

2.3.3. Software Interfaces

Software interfaces are:

- **Operating System:** Windows OS will be used for smooth running.
- **Google Colab:** Image segmentation model to be trained on google colab.
- **PyCharm:** Python code for Image processing techniques.
- **VS Code:** For front-end implementation

2.3.4. Communications Interfaces

The communication interfaces are:

- Internet

2.4. System Features

Our project main features is to apply segmentation and detection techniques on medical images.

2.4.1. Segmentation

This is the feature of segmentation in which different segmentation techniques can be applied.

2.4.1.1. Description and Priority

Priority**:** High.

Description**:** Given and uploaded image will be segmented. Techniques of thresholding like Adaptive, Binary, Otsu can be applied in order to get the desired result. A machine learning model is used for Image segmentation.

2.4.1.2. Stimulus/Response Sequence

- User opens the Image Annotator.
- User choose an image and upload it.
- Images will be segmented.

2.4.1.3. Functional Requirements

REQ-SF2-1: System takes an image in order to give result. Images can be in .jpg, jpeg, .jpg or any other format.

REQ-SF2-2: A machine learning model is used in image segmentation.

REQ-SF2-3: System shows the segmented image to the user on web-based application's UI.

2.4.2. Detection

This is the feature of detection in which we can detect the objects from images.

2.4.2.1. Description and Priority

Priority**:** High.

Description**:** A bounding box/square will be generated on uploaded image.

2.4.2.2. Stimulus/Response Sequences

- User opens the Image Annotator.
- User choose an image and upload it.
- User can draw a square of any size on image in order to identify and locate object.

2.4.2.3. Functional Requirements

REQ-SF2-1: System takes an image in order to give result. Images can be in .jpg, jpeg, .jpg or any other format.

REQ-SF2-2: A bounding box will be formed on image and then the system shows the results to the user on web-based application's UI.

2.5. Nonfunctional Requirements

2.5.1. Performance Requirements

Our performance requirement is to annotate medical images better than the other tools. That's the main purpose of developing this tool. It should be having good results.

2.5.2. Safety Requirements

We will make our tool more secure than the other tools. But there are no such safety issues to which we should focus on now.

2.5.3. Security Requirements

There are no such security issues to which we should focus on now.

2.5.4. Usability Requirements

The interface is good enough for the user to learn how to use the tool in few minutes. The user should have a web browser in order to access the tool.

2.5.5. Reliability Requirements

User should provide correct images files in formats like .jpeg, .jpg, .png, .bmp etc. in order to get the desired results and get the most reliable system output.

2.5.6. Maintainability/Supportability Requirements

The tool should be easy to maintain. There should be a clear separation between the interface and logic code.

2.5.7. Portability Requirements

Tool is web-based application which can be used in both mobiles and PCs. Both mobiles and PCs browsers will be able to access the tool hence it increases probability.

2.5.8. Efficiency Requirements

The system should respond to the user within 3 seconds.

Chapter 3
Use Case Analysis

Chapter 3: Use Case Analysis

The use cases of Image annotation tool are:

1. Login
2. Upload an image
3. Segmentation
4. Detection
5. Save Annotated Images

3.1. Use Case Model

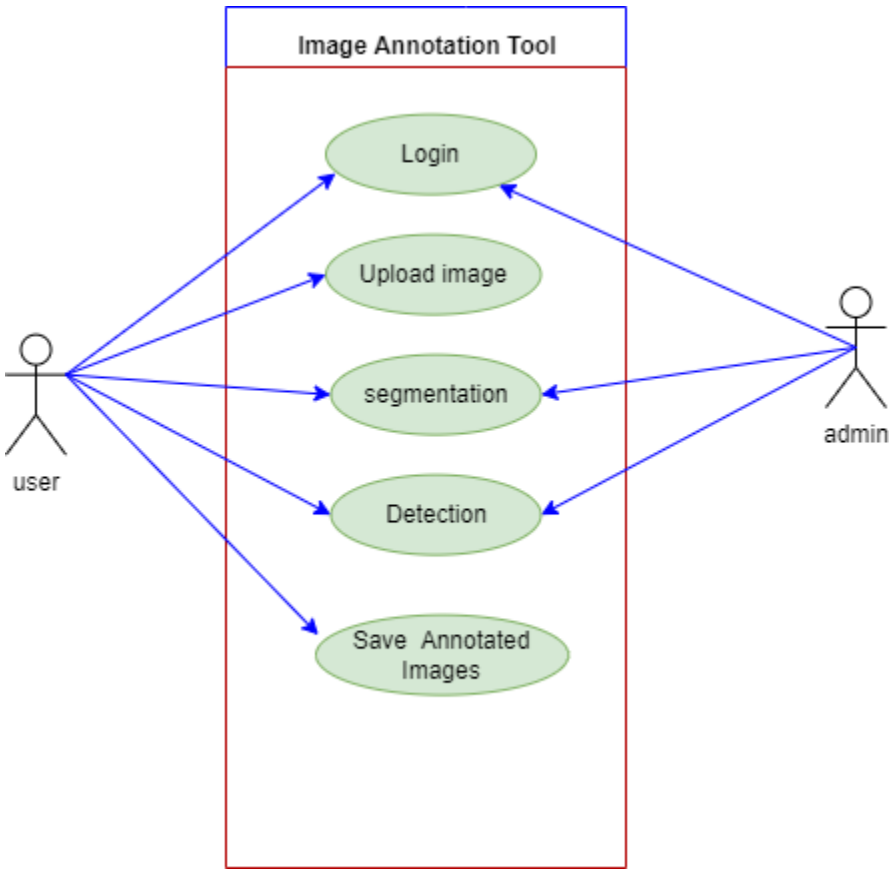


Figure 3.1: Use Case for Image Annotator

3.2. Use Cases Description

3.2.1. User Login Function

Title	Allow user and admin to sign in
Requirement	User must be registered through admin
Rational	Login to the system
Restriction or Risk	No Wrong logins
Dependency	Pc, Internet connection
Priority	Safety, timing

Table 2.1: User Login Function

Use case 1

Login
Actor
• Admin & User
Preconditions
• Must be register through Admin and user
Basic flow
• User wants to sign in
Alternate flows
• User don't want to login
Post Condition
• User must Sign Out

Table 3.2: User Login Function: Use Case 1

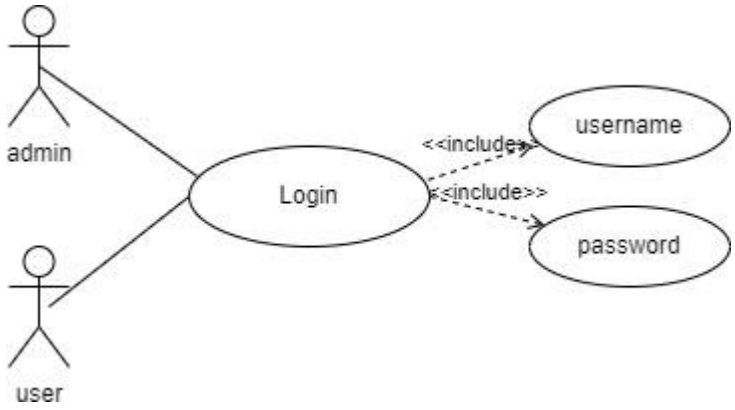


Figure 3.2: Use case diagram for login

3.2.2. Upload an image Function

Title	Upload an image
Requirement	User must be registered through admin
Rational	Upload an image to the system
Restriction or Risk	Wrong or correct image format not accepted.
Dependency	Pc, Internet connection
Priority	Safety, timing

Table 3.3: Upload an image Function

Use case 2

Upload an Image
Actor
• User
Preconditions
• User must be given access to use the system and user must be login to the system
Basic flow
• Sign in to the system
Alternate flows
• User don't want to upload an image
Post Condition
• User must Sign Out

Table 3.4: Upload an image Function: Use Case 2

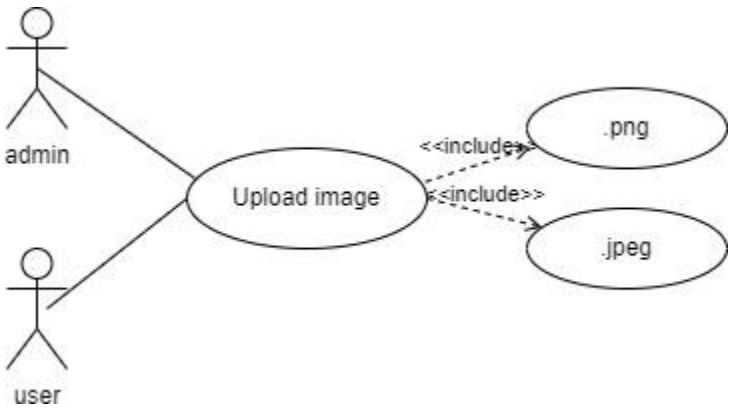


Figure 3.3: Use case diagram for Upload an image

3.2.3. Segmentation Function

Title	Segmentation
Requirement	There must be dataset of images in .png, .jpeg format and more.
Rational	Upload an image to the system
Restriction or Risk	Wrong or correct image format not accepted.
Dependency	Pc, Internet connection
Priority	Safety, timing

Table 3.5: Segmentation Function

Use case 3

Segmentation
Actor
• User and Admin
Preconditions
• System must be active
Basic flow
• sign in to the system
Alternate flows
• User don't want to perform segmentation
Post Condition
• Administrator must log out the system

Table 3.6: Segmentation Function: Use Case 3

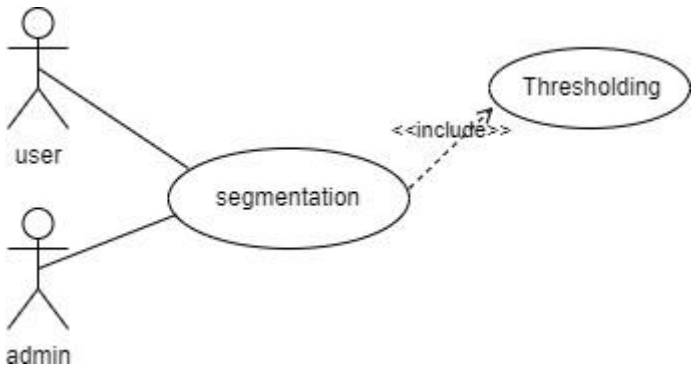


Figure 3.4: Use case diagram for Segmentation

3.2.4. Detection Function

Title	Detection
Requirement	There must be dataset of images in .png, .jpeg format and more.
Rational	Upload an image to the system
Restriction or Risk	Wrong or correct image format not accepted.
Dependency	Pc, Internet connection
Priority	Safety, timing

Table 3.7: Detection Function

Use case 4

Detection
Actor
• Admin and user
Preconditions
• System must be active
Basic flow
• Sign in to the system
Alternate flows
• User don't want to perform detection
Post Condition
• Administrator must log out the system

Table 3.8: Detection Function: Use Case 4

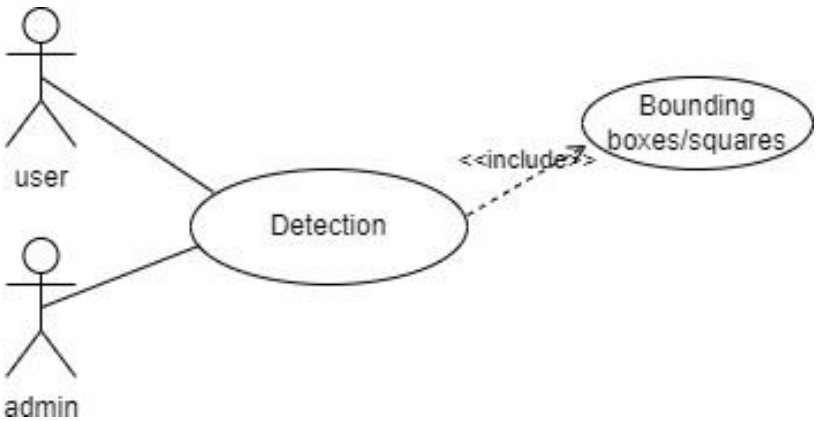


Figure 3.5: Use case diagram for Detection

3.2.5. Save Annotated images Function

Title	Save annotated image
Requirement	There must be dataset of images in .png, .jpeg format and more.
Rational	Upload an image to the system
Restriction or Risk	Wrong or correct image format not accepted.
Dependency	Pc, Internet connection
Priority	Safety, timing

Table 3.9: Save Annotated images Function

Use case 5

Save Annotated images
Actor
• User
Preconditions
• Images must be annotated
Basic flow
• sign in to the system
Alternate flows
• User don't want to save images.
Post Condition
• User must log out the system

Table 3.10: Save Annotated images Function: Use Case 5

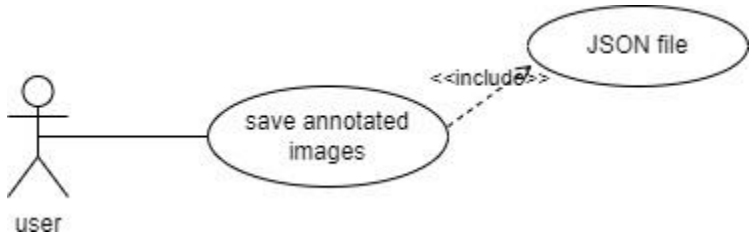


Figure 3.6: Use case diagram for Save Annotated images

Chapter 4
System Design

Chapter 4: System Design

This chapter outlines the design of the system in the form of diagrams as follow:

4.1. Architecture Diagram

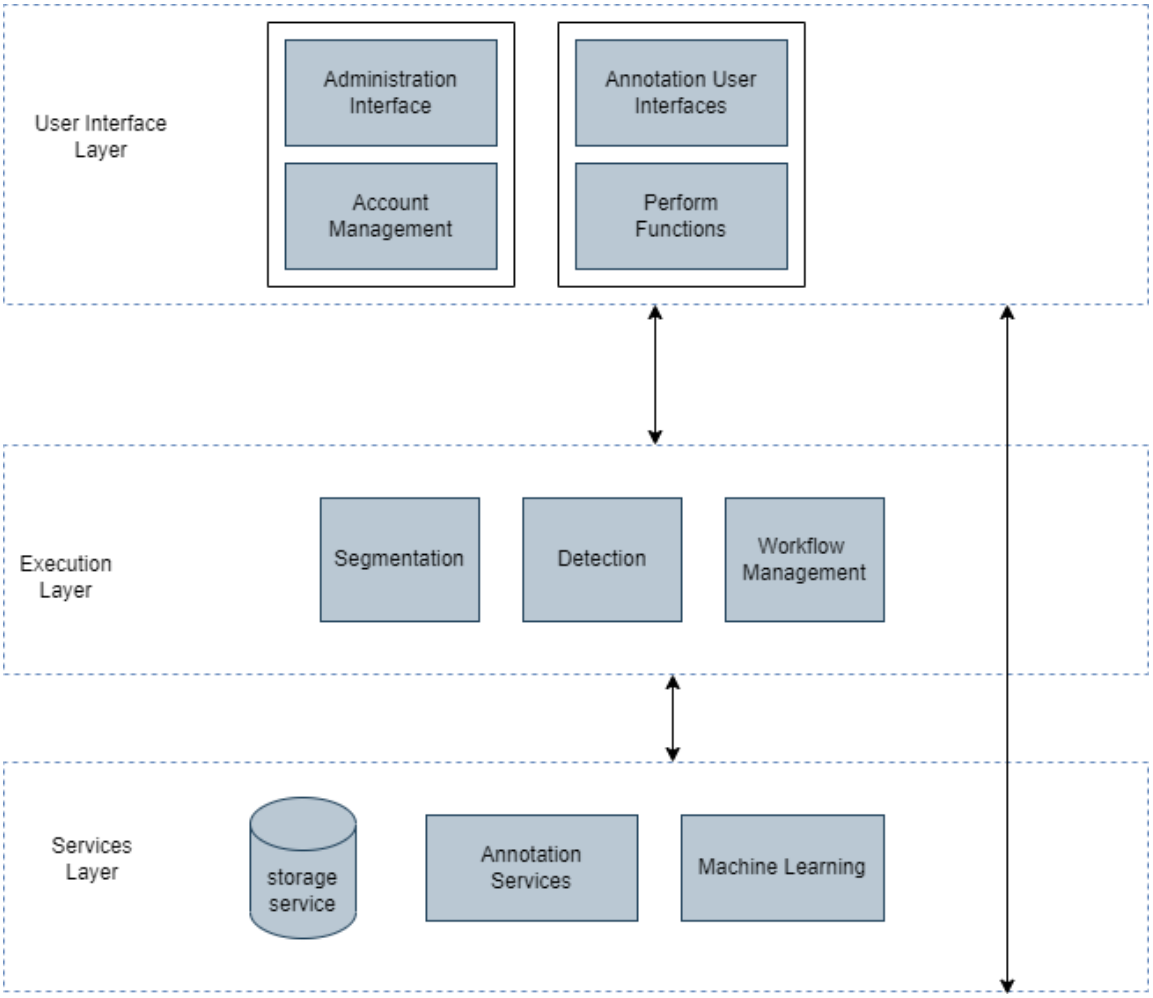


Figure 4.1: Architecture Diagram

4.2. Domain Model

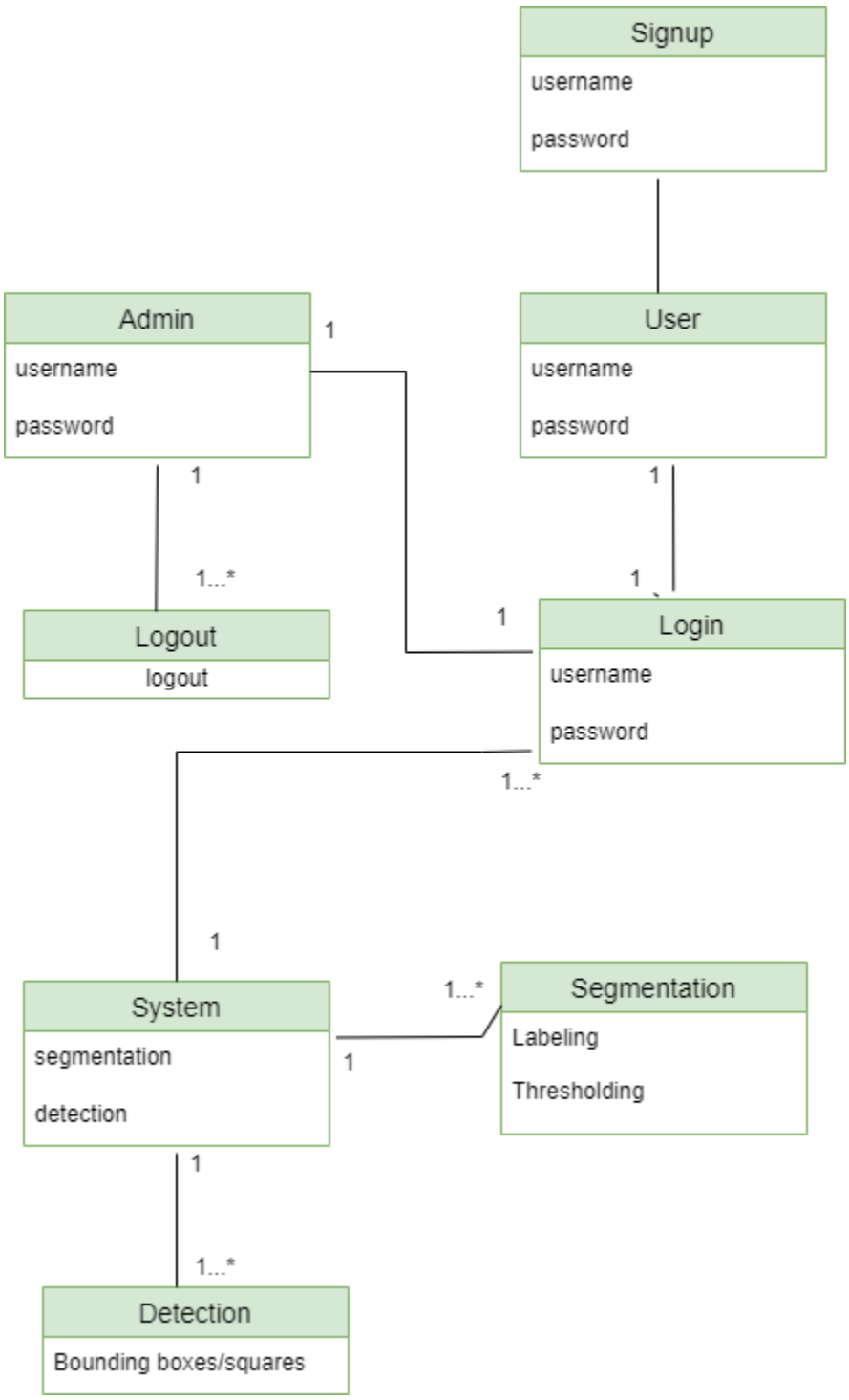


Figure 4.2: Doman Model Diagram

4.3. Entity Relationship Diagram with data dictionary

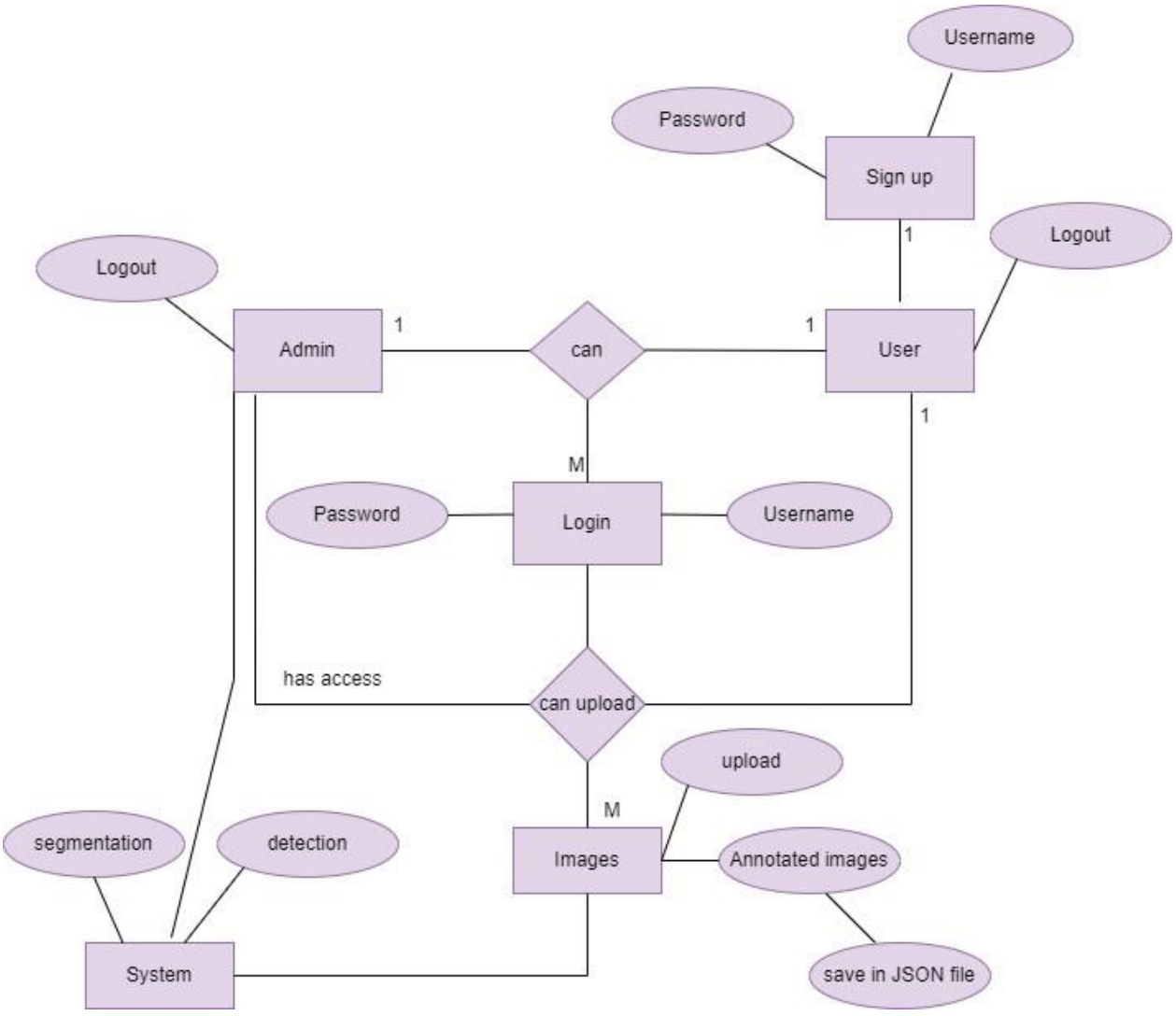


Figure 4.3: ER-Diagram

4.4. Class Diagram

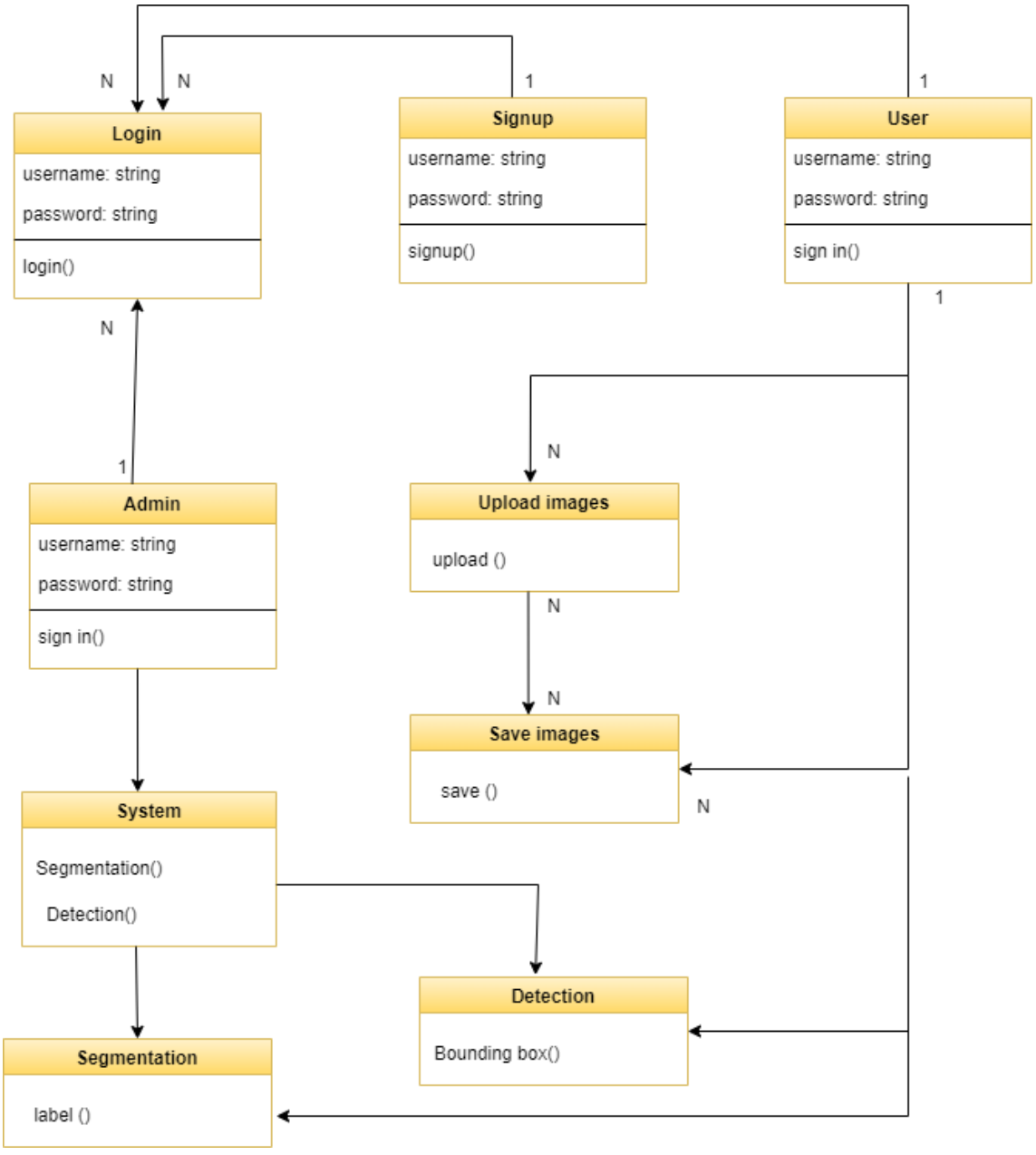


Figure 4.4: Class Diagram

4.5. Sequence / Collaboration Diagram

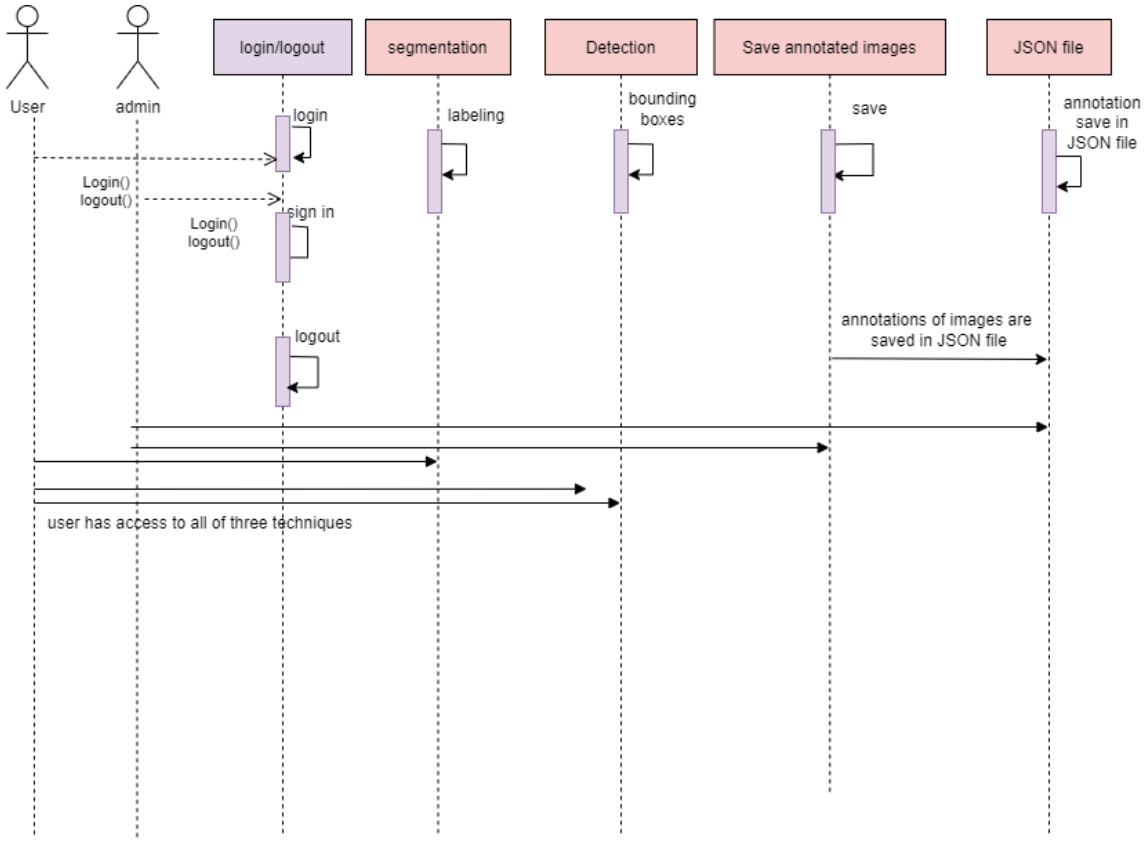


Figure 4.5: Sequence Diagram

4.6. Activity Diagram

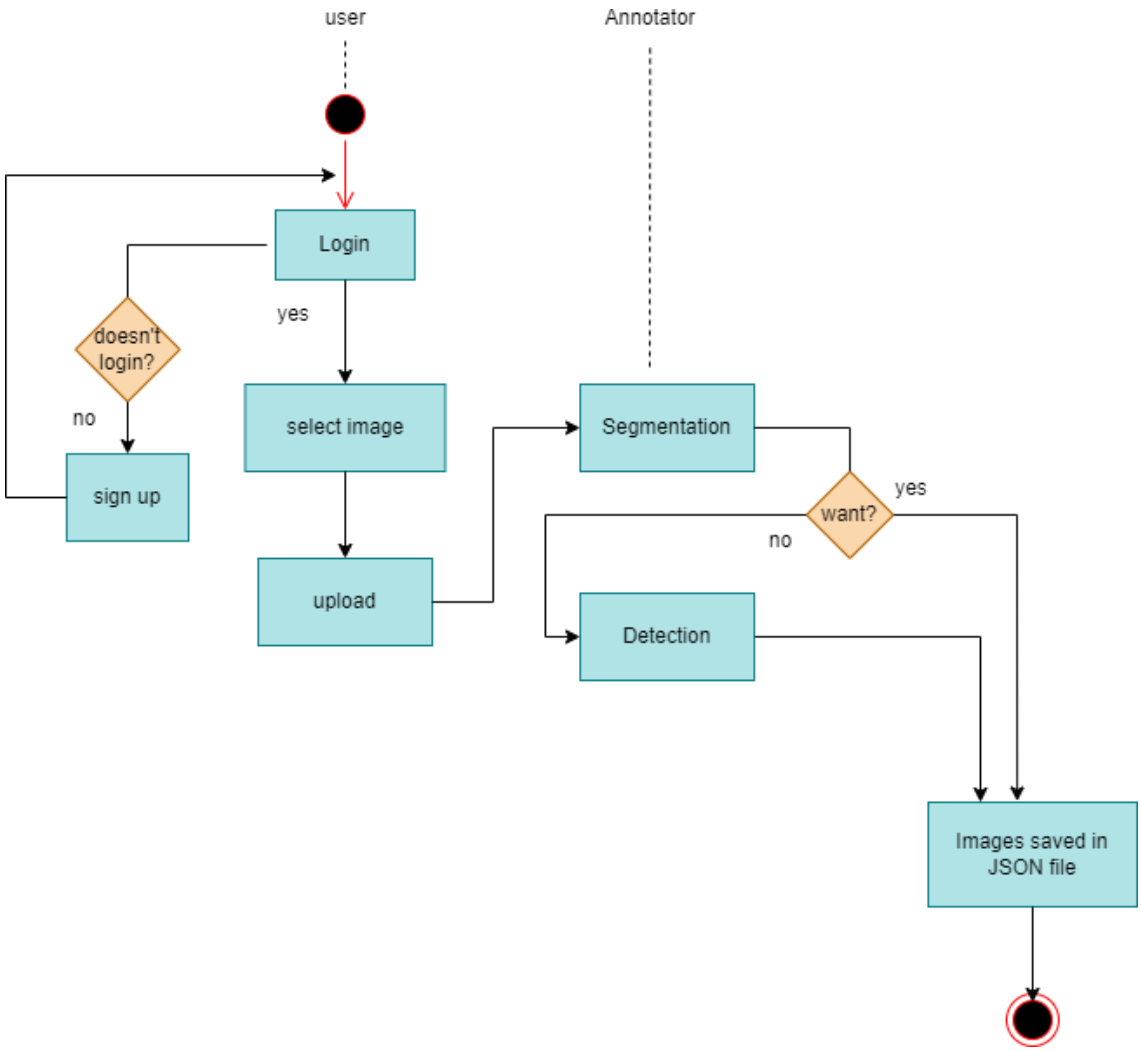


Figure 4.6: Activity Diagram

4.7. State Transition Diagram

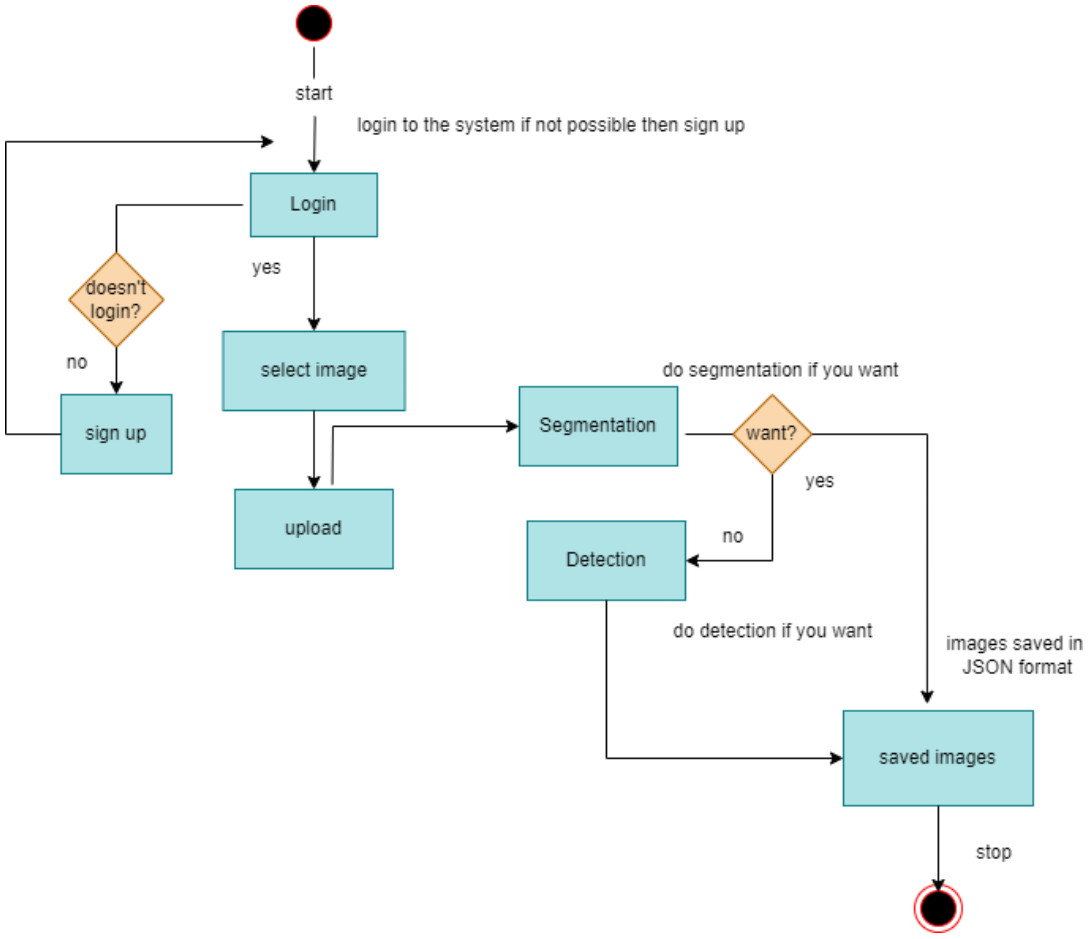


Figure 4.7: State transition diagram

4.8. Data Flow diagram

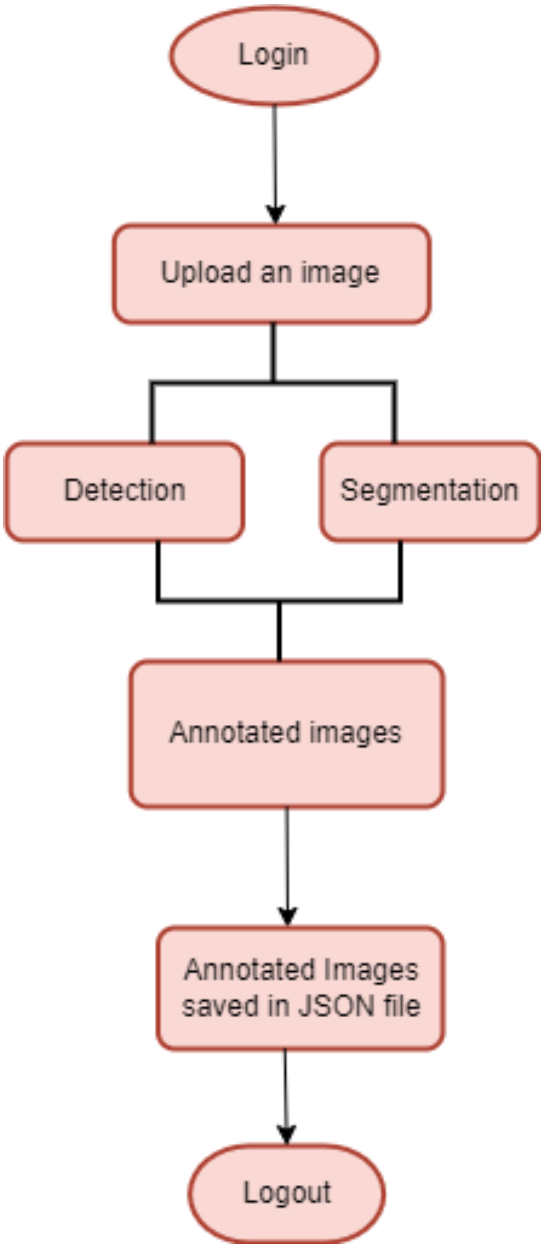


Figure 4.8: Flow Diagram

Chapter 5
Implementation

Chapter 5: Implementation

This chapter is all about what technical things we used in our system like language, tools, source, scope, and environment we used to achieve our main goal.

5.1. Important Flow Control/Pseudo codes

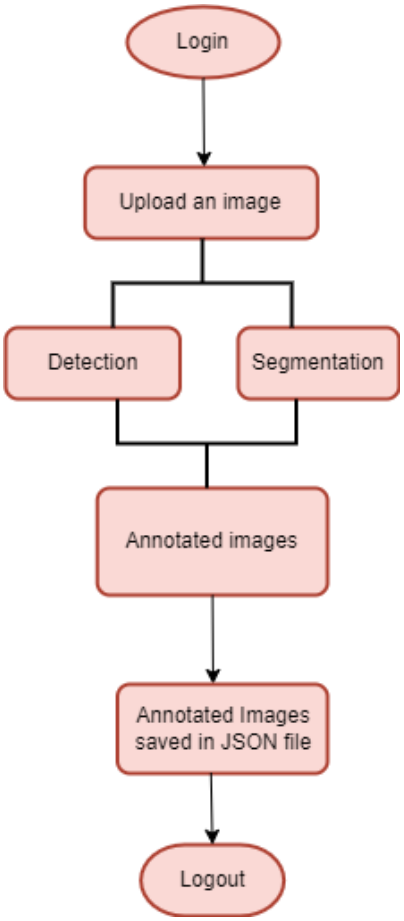


Figure 5.1: Flow control

Front page:

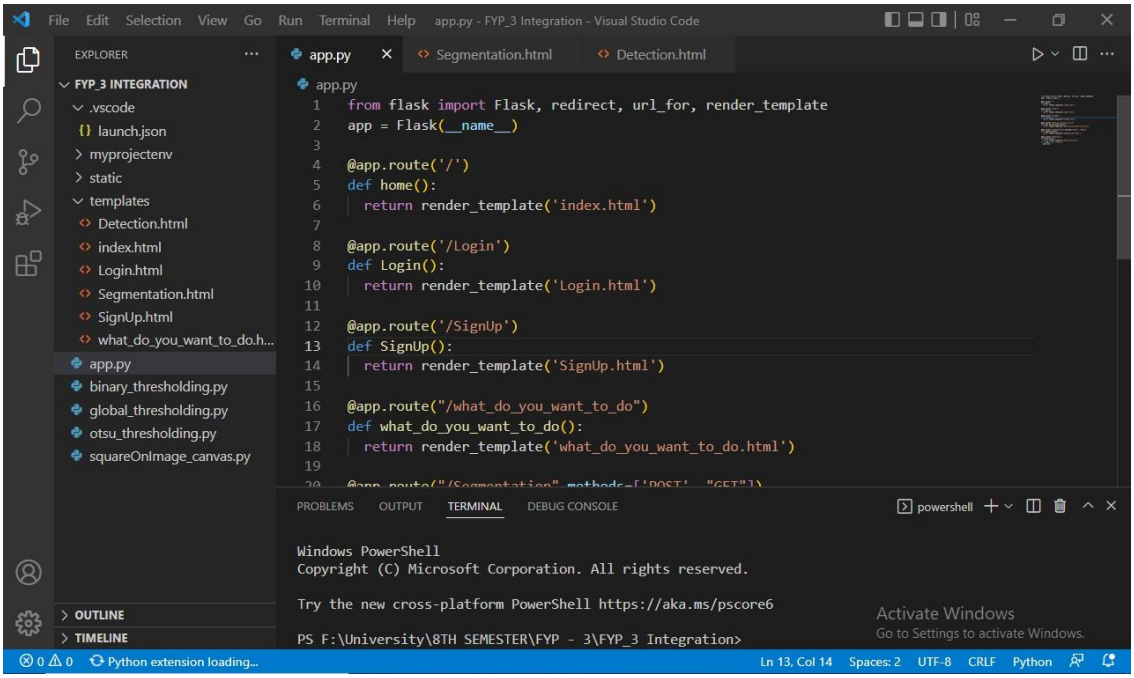


Figure 5.2: Front page

Segmentation:

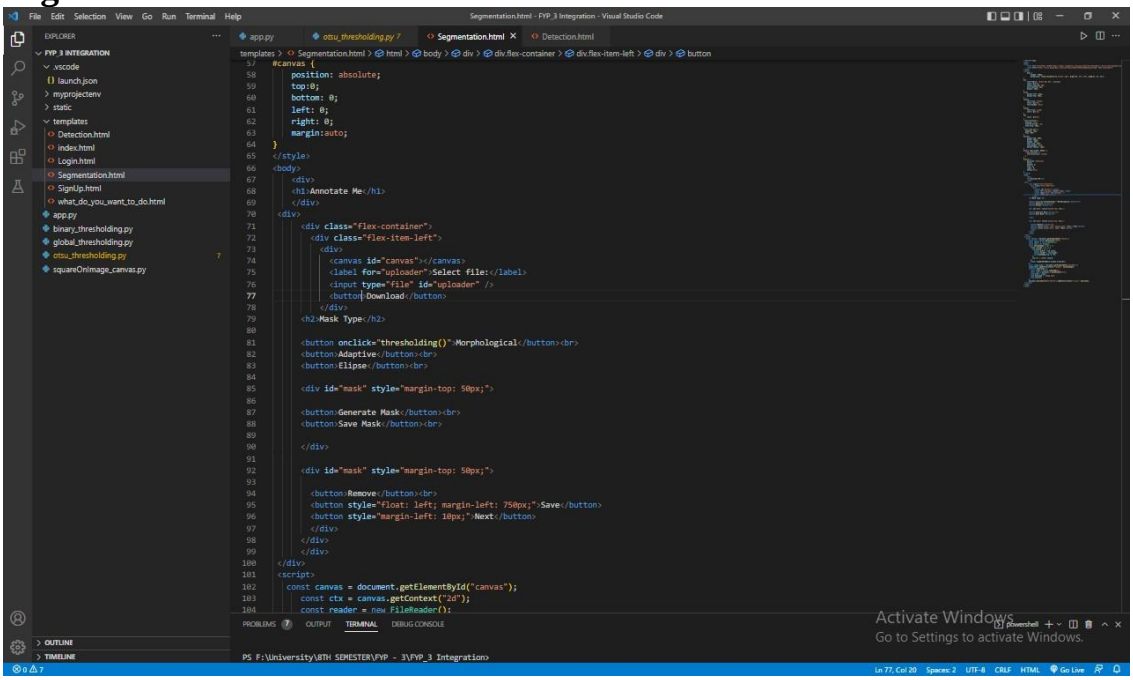


Figure 5.3: Segmentation

1. Binary Thresholding:

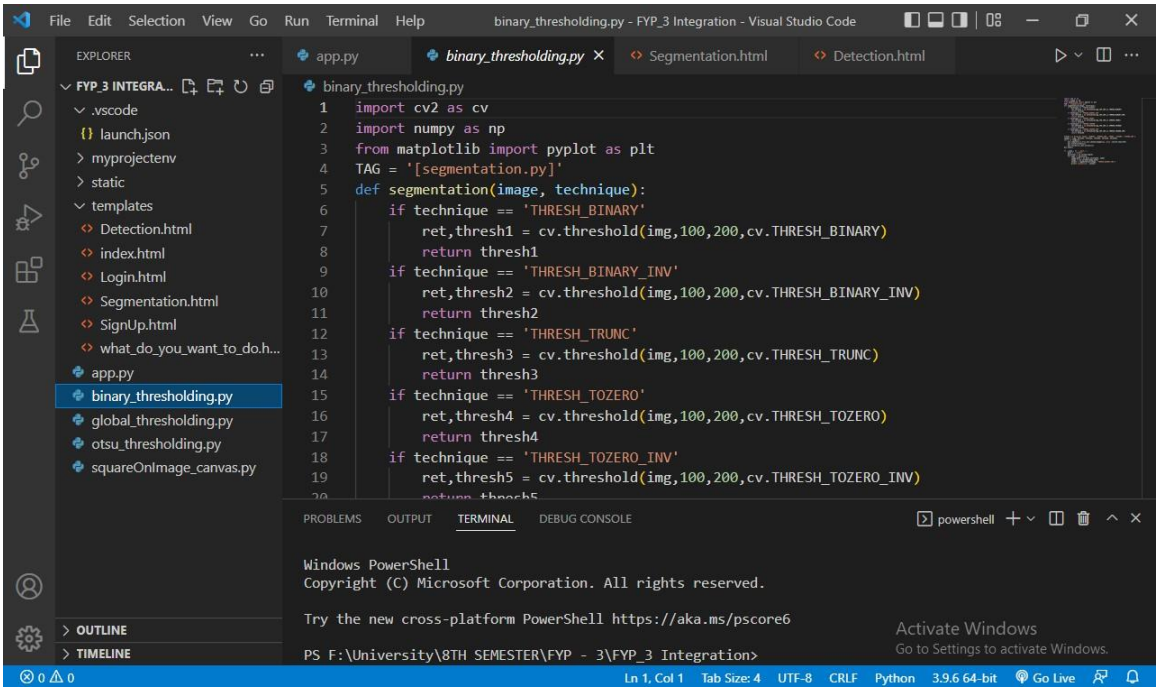


Figure 5.4: Binary Thresholding

1. Otsu thresholding

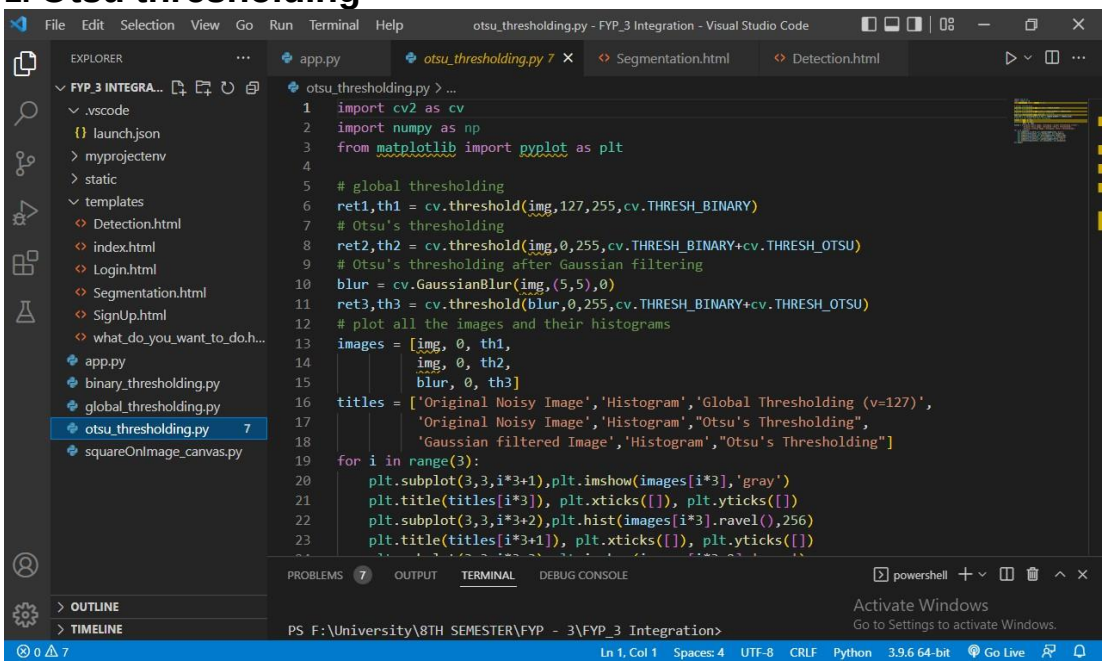


Figure 5.5: Otsu Thresholding

Detection:

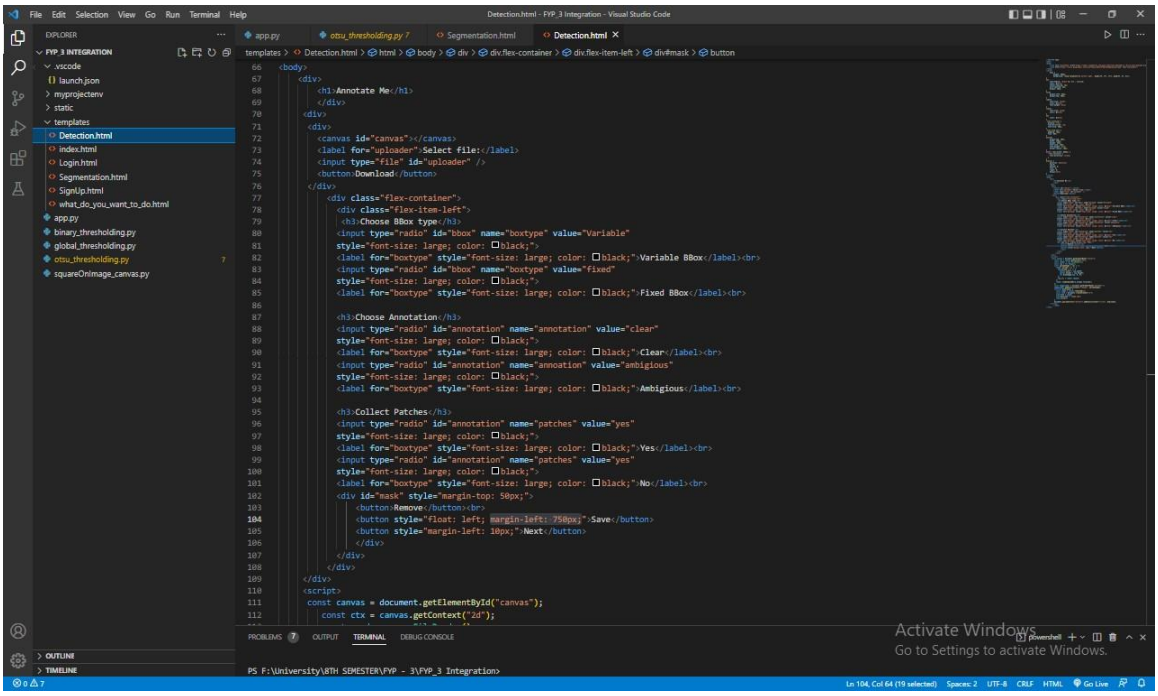


Figure 5.6: Thresholding

1. Square On Image

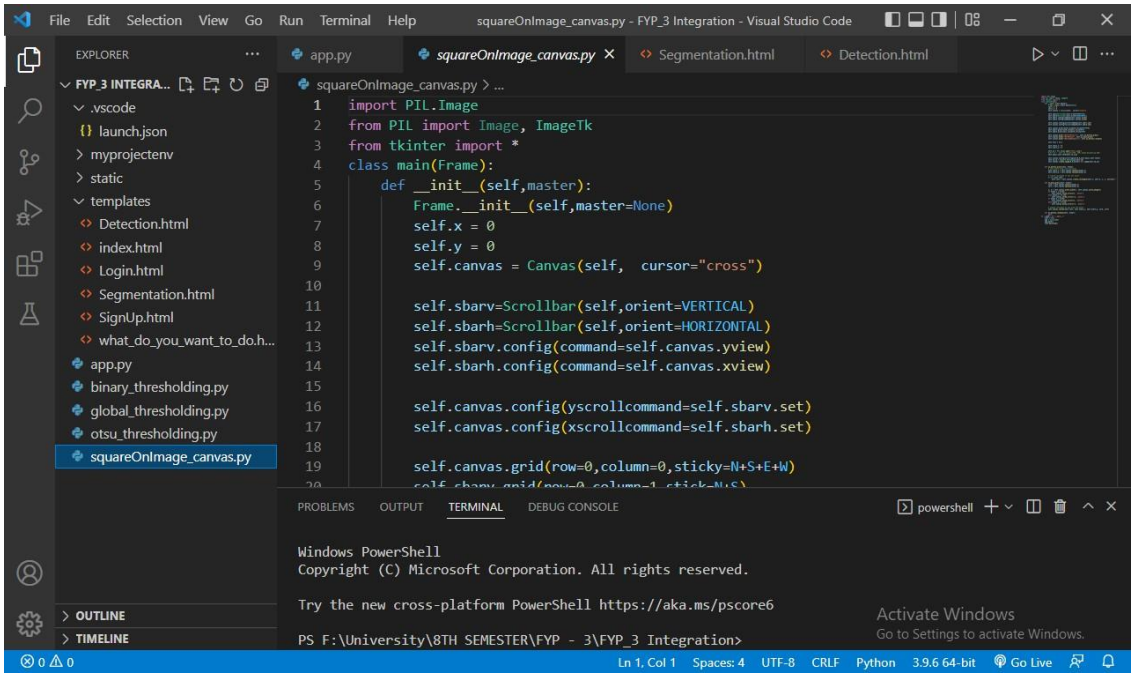


Figure 5.7: Square On Image

5.2. Components, Libraries, Web Services and stubs

- Html, CSS and JavaScript framework for frontend
- Image annotations saved into json file.
- PyCharm, Google Colab or VS-Code for backend.
- OpenCv, numpy, Pandas, pillow, matplotlib. Scikit learn, scikit, Flask image libraries are used.

5.3. Deployment Environment

We have deployed our web application on internet.

Visual studio code: The software used for designing the frontend ot the tool is "VScode".It is a source code editor which is used with a number of programming languages. This includes Javascript, HTML, CSS and PYTHON. It is used to develop websites, apps, web services and mobile apps.

Flask: Flask is a small and lightweight Python web framework that provides useful tools and features that make creating our web application in Python easier.

Canvas: The Canvas display various graphics on the application. In our tool, first of the image is being uploaded on canvas and then detection and segmentation operations are performed.

Google colab / PyCharm: It is easy to use, interactive data science IDE, in our project it will be used for training ML model in python language.

5.4. Tools and Techniques

Tools:

- Visual Studio Code
- PyCharm

- Google colab

Technologies:

- Python
- Flask
- Canvas
- Image processing techniques
- Git Version Control

5.5. Best Practices / Coding Standards

- Had a plan and purpose behind the development.
- To made the system accessible freely to users.
- User Interface built keeping in mind the user's experience.
- Kept code useful and concise.
- System compatibility.
- Our code (both backend and frontend) is reusable as out tool is an open source. Our model is scalable as we can train it on more images from time to time.

5.6. Version Control

Administrator has access to the system modification and updating the system i.e. update system features etc.

Chapter 6
Business Plan

Chapter 6: Business Plan

In this section, we will be discussing the business plan of our project. We will be discussing that how our project can be better than existing products and how it can bring change in the market.

6.1 Business Description

This Image Annotator could be used in market as it will be delivering more accurate results and it is specifically used for medical dataset which to accelerate the development of automated systems.

6.2 Market Analysis & Strategy

In the market, most of the image search engines are using old feature extraction techniques. We aim to improve the existing results by using Machine Learning model in our product as it is computationally efficient.

6.3 Competitive Analysis

There are some websites on internet which are annotating any random image and consume a little more time in as compared to our tool that only consume 3 seconds. And it is specifically used for medical dataset. As this tool will be of great help to people who require Image Annotation Tool and researchers who plan to work in this field in future so we develop an open source tool which will be free of cost.

6.4 Products/Services Description

This tool is a web based open source application that can perform Image processing techniques of Segmentation and Detection. An uploaded image will be annotated according to the user's requirement and need. The tool will be helpful for hospital for annotating medical images.

6.5 SWOT Analysis

Strength:

- It generates bounding box for detection problems and segmentation mask for segmentation problems.
- It can draw single box (rectangle).
-

Opportunity:

- It generates annotation in JSON format.
-

Weakness:

- It can work image by image.
- It is not fully automated.
-

Threat:

- Incase if any hospital will use this tool then it will get security issues as it is an open source tool.

Chapter 7
Testing & Evaluation

Chapter 7: Testing and Evaluation

7.1 Equivalence partitioning

User can upload .png, .jpg or .jpeg type image. The results will be same.

7.2 Boundary value analysis

When a user will upload any other file instead of an image from dataset, the file will be uploaded but it will not show any results and the file will not be uploaded on canvas.

Figure 7.1: Boundary Value Analysis

7.3 Data flow testing

User can annotate the images by accessing the tool online via internet. The image will be uploaded by user and then according to the user's choice the action will be performed.

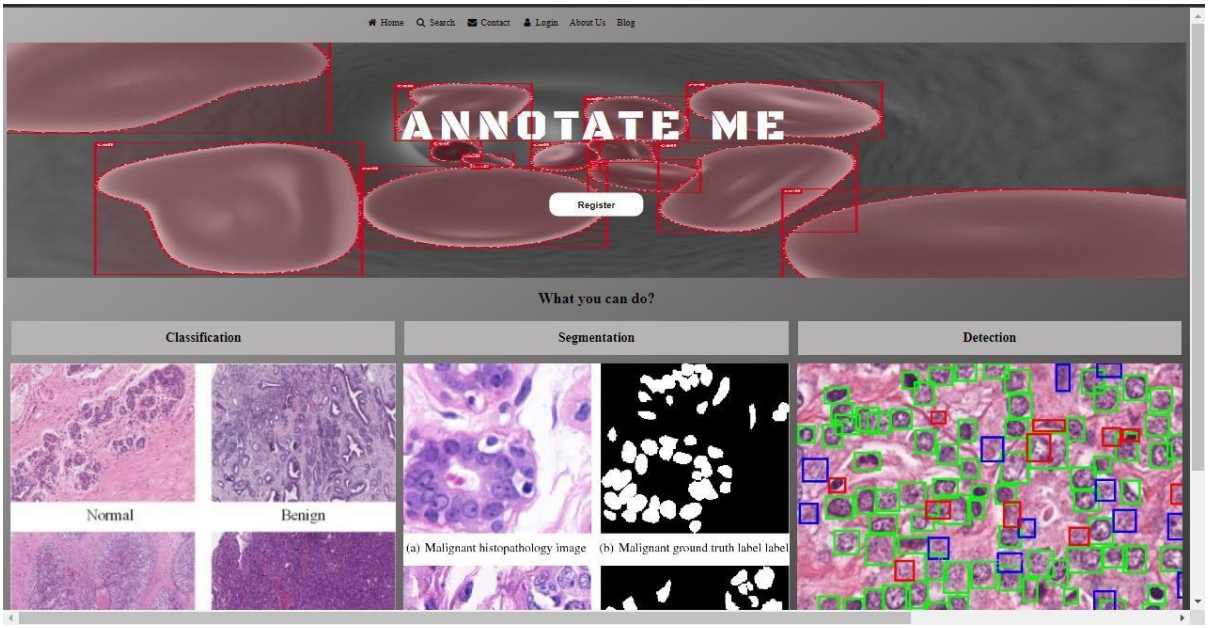


Figure 7.2: Data Flow Testing (1)

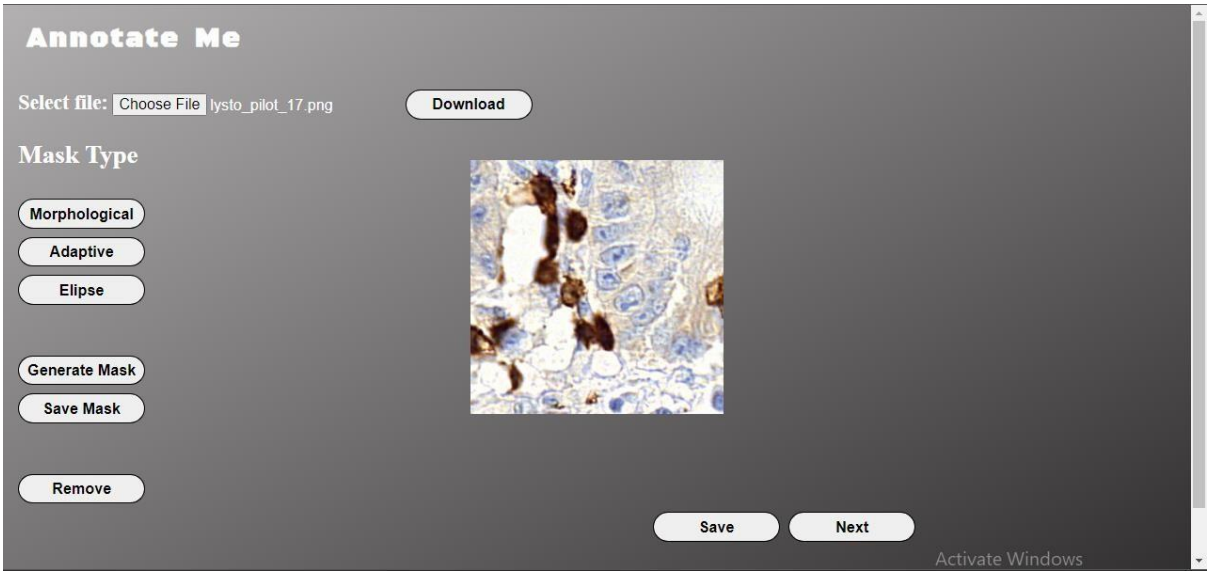


Figure 7.3: Data Flow Testing (2)

7.4 Unit testing

In unit testing we tested all the components of our project separately so that defects can be fixed nearly in development stage. On our user interface we have Register, Segmentation

and Detection buttons, on clicking these buttons new windows will open i.e. login or signup windows, and by clicking segmentation and detection buttons, a window of different operations will appear where you can upload an image and then perform different techniques. After this you can download your annotated image. The last module tested was that whether our tool is giving right output or not and was functioning correctly and was giving right annotations on different input images.

7.5 Integration testing

After unit testing, we performed integration testing to make sure that all of the Image Annotation Tool modules are working together. These modules include backend and frontend. In frontend we have modules of Sign up, Sign in, Segmentation and Detection buttons. All of them were functioning together as a whole and were giving the desired output.

7.6 Performance testing

The speed, response time, stability, resource usage reliability and scalability of our project under particular workload was tested.

7.7 Pressure Testing

The user cannot upload more than one image at a time. The image with more than 2MB size will not be uploaded for annotation.

Chapter 8

Conclusion & Future Enhancements

Chapter 8: Conclusion & Future Enhancements

8.1 Achievements and Improvements

We have achieved the desired results of Segmentation and Detection in this project. This tool will annotate images within 3 seconds. This is improvement from other tools. Through our project Image Annotation Tool, we have learned a lot. Firstly, we studied Machine Learning in detail, its basic concepts and technologies which we were not familiar with at all. This gave us a huge insight in this field. We learned about different tools and technologies of ML and applied them successfully. Other than this, we got hands-on experience on HTML, CSS and Flask by making our front-end. Also we have learned Canvas. One of the huge learnings through this project is how to work in a team and how to manage and complete tasks in a specific period of time.

8.2 Critical Review

While reviewing our project the problem with our current application is that it is not covering all the techniques of Image processing. The annotated images are not saved in multiple formats. A better approach for an image annotator is that it should save results in multiple formats. While comparing with other tools we concluded that this tool is easy to use and it is open source web based application.

8.3 Lessons Learnt

We have learnt following lesson by working on this project. We got the knowledge of **python language** as our backend was purely based on python as we train the model of machine learning. We got to know how to create the frontend using **HTML, CSS** and how to integrate it using the **FLASK**. We also came to know about the usage of **CANVAS**. All the time period we have spent on this project was quite useful for us. We faced a lot of errors but get over them within the time period that helped us in adopting the quality of Time Management. During this project we got to know how to work as a team.

8.4 Future Enhancements/Recommendations

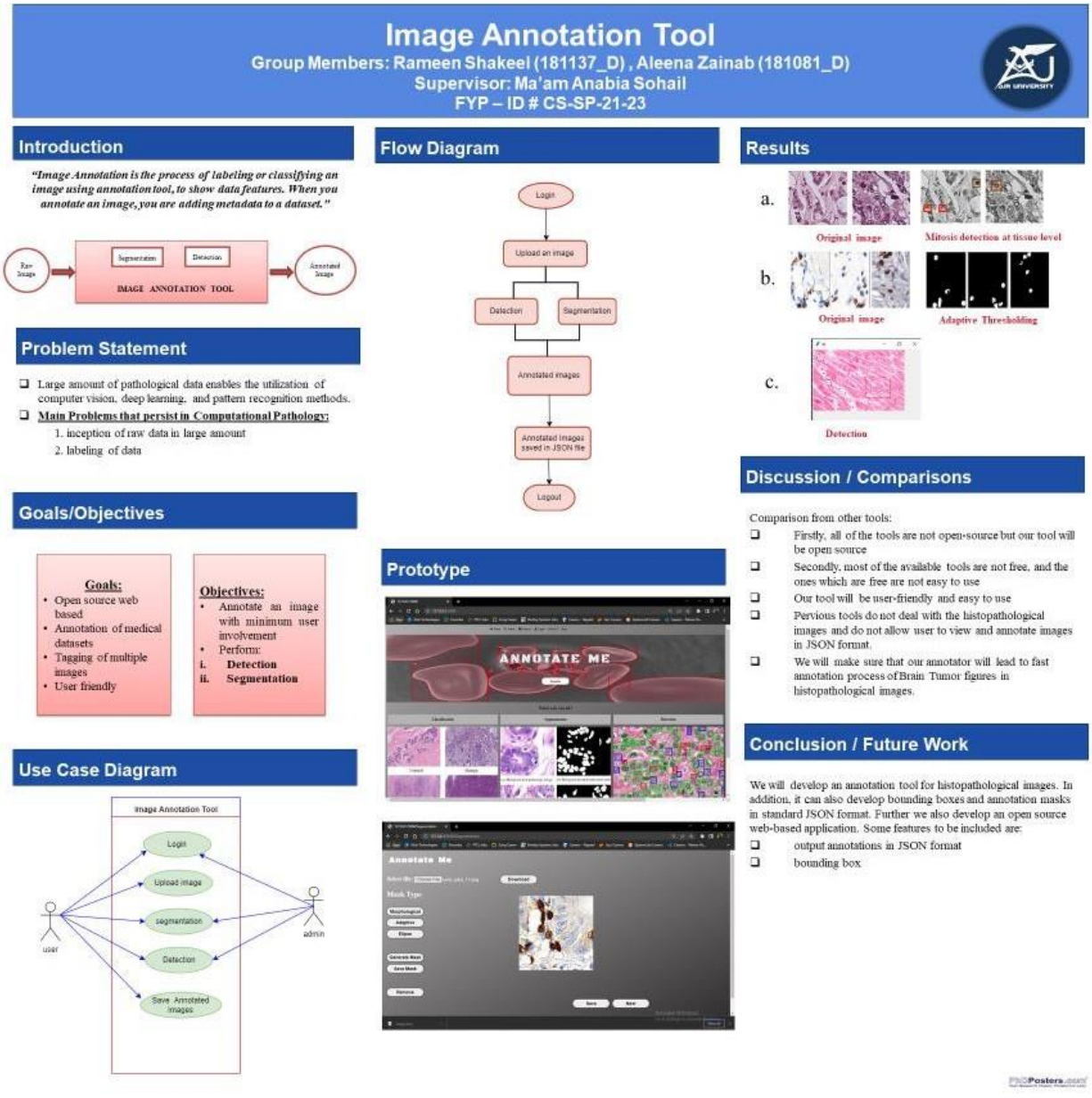
In future we will add more functionalities to our project and we will also add these functionalities in our premium version so that it can be extended to commercial product that will be of high quality with low price. And we will try to add more techniques in this tool. We can also make our frontend responsive so that it can be used on mobiles and tablets as a proper web application.

Appendices

Appendix A: Information / Promotional Material

This chapter is about how we will promote our project in market.

A.1. Poster



Reference and Bibliography

[1] A. C. Ruifrok and D. A. Johnston, “Quantification of histochemical staining by color deconvolution,” 2001.

[2] A. Dutta and A. Zisserman, “The VIA Annotation Software for Images, Audio and Video,” MM 2019 - Proc. 27th ACM Int. Conf. Multimed., pp. 2276–2279, Apr. 2019, doi: 10.1145/3343031.3350535.

[3] A. Torralba, B. C. Russell, and J. Yuen, “LabelMe: Online image annotation and applications,” in Proceedings of the IEEE, 2010.

[4] M. Aubreville, C. Bertram, R. Klopfleisch, and A. Maier, “SlideRunner: A tool for massive cell annotations in whole slide images,” in Informatik aktuell, Feb. 2018, no. 211279, pp. 309– 314, doi: 10.1007/978-3-662-56537-7_81.

[5] M. Satyanarayanan, A. Goode, B. Gilbert, J. Harkes, and D. Jukic, “OpenSlide: A vendor-neutral software foundation for digital pathology,” J. Pathol. Inform., vol. 4, no. 1, p. 27, 2013, doi: 10.4103/2153-3539.119005.

[6] M. Tkachenko, M. Malyuk, N. Shevchenko, A. Holmanyuk, and N. Liubimov, “Label Studio: Data labeling software,” 2020. https://github.com/heartexlabs/label-studio (accessed Oct. 09, 2020).

[7] “microsoft/VoTT: Visual Object Tagging Tool: An electron app for building end to end Object Detection Models from Images and Videos.” https://github.com/microsoft/VoTT (accessed Oct. 09, 2020).

[8] “openvinotoolkit/cvat: Powerful and efficient Computer Vision Annotation Tool (CVAT).” https://github.com/openvinotoolkit/cvat (accessed Oct. 09, 2020).

[9] F. Ciompi, Y. Jiao, and J. van der Laak, “Lymphocyte Assessment Hackathon (LYSTO),” Oct. 2019, doi: 10.5281/ZENODO.3513571.

[10] S. Van Der Walt et al., “Scikit-image: Image processing in python,” PeerJ, vol. 2014, no. 1, 2014, doi: 10.7717/peerj.453.

[11] P. Marquez-Neila, L. Baumela, and L. Alvarez, “A morphological approach to curvature-based evolution of curves and surfaces,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 36, no. 1, pp. 2–17, Jan. 2014, doi: 10.1109/TPAMI.2013.106.

[12] T. Chan and L. Vese, “An active contour model without edges,” in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 1999, vol. 1682, pp. 141–151, doi: 10.1007/3-540-48236-9_13.

[13] P. Getreuer, “Chan-Vese Segmentation,” Image Process. Line, vol. 2, pp. 214–224, Aug. 2012, doi: 10.5201/ipol.2012.g-cv.

[14] A. Khan, A. Sohail, U. Zahoora, and A. S. Qureshi, “A survey of the recent architectures of deep convolutional neural networks,” Artif. Intell. Rev., vol. 53, no. 8, pp. 5455–5516, Apr. 2020, doi: 10.1007/s10462-020-09825-6