Brain Tumor detection

Final Year Project

Session 2020-2024

A project submitted in partial fulfillment of the degree of

BS in Computer Science

Department of Computer Science

Faculty of Computer Science & Information Technology

The Superior University, Lahore

Spring 2024

Type (Nature of project)	[✓] **D**evelopment [ ] **R**esearch [ ] **R&D**
Area of specialization	Research
FYP ID	FYP-BCSM-F23-053

Project Group Members

Sr.#	Reg. #	Student Name	Email ID	*Signature
(i)	Bcsm-f20-068	Haseeb Malik	Bcsm-f20-068@superior.edu.pk	
(ii)	Bcsm-f20-252	Ali Arshad	Bcsm-f20-252@superior.edu.pk	
(iii)	Bcsm-f20-221	Abdul Hanan	Bcsm-f20-221@superior.edu.pk	

*The candidates confirm that the work submitted is their own and appropriate credit has been given where reference has been made to work of others

Plagiarism Free Certificate

This is to certify that, I **Haseeb Malik** S/o Allah Wasaya, group leader of FYP under registration no **FYP-Bcsm-F20-053** at Computer Science Department, The Superior University, Lahore. I declare that my FYP report is checked by my supervisor.

Date: Name of Group Leader: Haseeb Malik Signature: _____

Name of Supervisor: Muhammad Jameel Co-Supervisor: Miss Maria Iqbal

Designation: Lecturer Designation: Lecturer

Signature: _____ Signature: _____

Hod: Dr. Irfan ud din

Signature: _____

Project Report

[Brain Tumor Detection]

Change Record

Author(s)	Version	Date	Notes	Supervisor's Signature
Haseeb, Hanan, Ali	1.0		<change in figures>	
Haseeb, Hanan, Ali	1.0		<change in use cases>	
	1.0		<change in diagrams>	
Ali, Hanan, Haseeb	1.0		<updation in user interface>	
Ali, Haseeb, Hanan	1.0		<documentation adjustments>	

APPROVAL

PROJECT SUPERVISOR

Comments: _____

_____

Name: _____

Date: _____ Signature: _____

PROJECT Manager

Comments: _____

_____

Name: _____

Date: _____ Signature: _____

HEAD OF DEPARTMENT

Comments: _____

_____

Name: _____

Date: _____ Signature: _____

Dedication

We dedicate this project to all those humble beings who have aided us in any way to become what we are today. Whose sacrifices their precious time, especially our parents who have felt pain beyond us and showered us a win and never-ending prayers and support. We deem as a divine scree of inspiration

Acknowledgements

All the praise to all Mighty ALLAH. For bestowing us with the courage. Knowledge, Health and wisdom to carry out this project. We are greatly indebted to our parents. Without their endless financial, moral support, patience and prayer. The wry idea of this project was impossible we would like to pay humble gratitude to our project supervisor Mr. Muhammad Jameel. His encouragement was the main source and strength to stimulate us to complete the project. We are also grateful to all faculty members of SUPERIOR UNIVERSITY for all facilities provided.

Executive Summary

The Brain Tumor Detection System (BTDS) project seeks to create a cutting-edge solution to detect brain tumors with high accuracy and speed. It will utilize a combination of medical imaging techniques, artificial intelligence, and machine learning algorithms to assist healthcare professionals in diagnosing brain tumors in their early stages. By doing so, we aim to improve patient care, reduce treatment costs, and ultimately save lives.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Chapter 1: Introduction

The “Human Brain Tumor Image Classification” is a desktop application, which will help to transform medical one-step towards development and technology. The purpose of this application is to provide a system, which will be able to classify MRI image of human Brain by incorporating Artificial Intelligence. The core benefit of this system is, it will reduce the utilization of resources such as time and money. The accuracy level of our system will be 95%. This system will be built in PyCharm community edition, Jupiter Notebook, and Google Collaboratory by using python language.

1.1. Background

In the realm of medical diagnostics, the detection of brain tumors is a pivotal challenge, often requiring intricate analysis of MRI and CT images. Traditional methods, reliant on manual interpretation, are time- consuming and prone to errors. The evolution of technology, specifically advancements in image processing and artificial intelligence, has spurred the development of a more efficient and accurate brain tumor detection system. Leveraging Python, a versatile programming language, facilitates the integration of sophisticated image processing techniques, including contrast enhancement and resizing. Artificial intelligence, particularly Convolutional Neural Networks (CNNs), plays a pivotal role in automating tumor classification, with pre- trained models offering valuable insights. Evaluation metrics such as accuracy and precision ensure the system's reliability, minimizing false positives. Ethical considerations, particularly regarding patient privacy, guide the system's design, prioritizing confidentiality. The collaborative involvement of medical professionals remains crucial, ensuring the system aligns with clinical standards and societal values, enhancing the overall efficacy of brain tumor detection.

1.2. Motivations and Challenges

There is no system present that perform this type of functionality so it will be very motivating to build such system that would help medical field in a unique way. This bring courage in developing something that is not present before.

1.3. Goals and Objectives

Develop an accurate and reliable brain tumor detection system using machine learning techniques, ultimately contributing to improved diagnostic capabilities in medical imaging.

1.4. Literature Review/Existing Solutions

My STORITM, BRAIN are the existing system that detect the Brain tumor. But it takes a lot of time of disease prediction. Many reasons behind this, first it trains on less datasets and second it develops on old technology. The structure of all these app is very complex. Our System will train on latest technology (Neural Network) which perform well then, the existing system. Existing system are not fully freely available but our system is fully freely available for everyone.

1.5. Gap Analysis

The project has collected brain MRI data for examination and put a simple CNN model for tumor identification into practice. There is a lack of diversity in the data, though, and investigating more datasets might improve the model's capacity to handle various scenarios. Patient data protection has been given top priority in this project; however, model explain ability should be strengthened. The current model is a simple CNN in terms of technology; for maybe improved accuracy, it could be beneficial to investigate more sophisticated architectures. Furthermore, there is space for enhancement to create a more user-friendly interface for clinicians, as it is now somewhat simple. To put it briefly, filling in these gaps will require expanding the variety of data

sources, investigating more complex models, increasing the explain ability of the models, and refining the user interface to create a brain tumor detection system that works better.

1.6. Proposed Solution

Our proposed Brain Tumor Detection System will consist of the following key components: **Input MRI Images:** The system will seamlessly integrate with existing medical imaging equipment, such as MRI and CT scanners, to obtain high-resolution images of the brain.

Image Preprocessing: Prior to analysis, the acquired medical images will undergo preprocessing to enhance image quality and reduce noise. Machine Learning Algorithms: We will employ state-of-the-art machine learning algorithms, including convolutional neural networks (CNNs), to analyze the medical images for tumor presence and characteristics.

User-Friendly Interface: The system will have an intuitive user interface that enables healthcare professionals to easily upload images, initiate the analysis, and receive real-time results. **Accuracy and Speed:** Our system will prioritize both high accuracy and rapid diagnosis, ensuring that patients receive timely and reliable results. Data Security and Privacy: Robust security measures will be implemented to protect patient data, complying with all relevant healthcare regulations.

Continuous Improvement: We will establish a feedback mechanism for healthcare professionals to provide input, ensuring ongoing system refinement and accuracy enhancement.

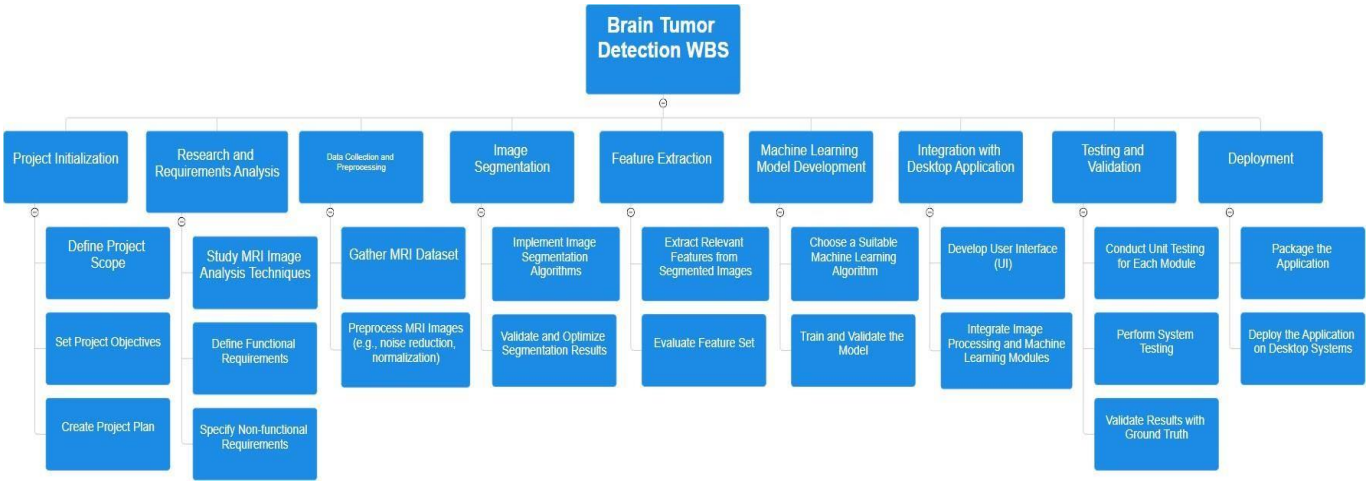
1.7. Project Plan

Develop an accurate and reliable brain tumor detection system using machine learning techniques, ultimately contributing to improved diagnostic capabilities in medical imaging.

Member	Tasks	Task Name	Duration (week)	Dependencies
Haseeb Malik(M1),Ali Arshad(M2),Abdul Hanan(M3)	T1	Requirement Meetings	1	-
Haseeb Malik(M1),Ali Arshad(M2),Abdul Hanan(M3)	T2	Features Discussion	1	T1(M1, M2,M3)
Haseeb Malik(M1),Ali Arshad(M2),Abdul Hanan(M3)	T3	Document the System	2	T1(M1, M2), T2(M1, M2,M3)
Haseeb Malik(M1),Ali Arshad(M2)	T4	Design Database	1	T3 (M1, M2,M3)
Haseeb Malik(M1)	T5	Software Design	2	T3 (M1, M2,M3)
Ali Arshad (M2)	T6	Interface Design	2	T5(M1)
Abdul Hanan (M3)	T7	Create design specification	2	T4 (M1,M2), T5 (M1), T6 (M2)
Haseeb Malik(M1), Ali Arshad(M2),	T8	Develop system module	2	T7(M3)
Haseeb Malik(M1), Ali Arshad(M2), Abdul Hanan(M3)	T9	Integrate system module	2	T8(M1, M2)
Ali Arshad(M2), Abdul Hanan(M3)	T10	Perform initial testing	2	T3(M1,M2,M3), T9(M1, M2)
Haseeb Malik(M1), Ali Arshad(M2),	T11	Perform system testing	2	T10(M2, M3)
Abdul Hanan(M3)	T12	Document issues found	2	T11(M1, M2)
Haseeb Malik (M1)	T13	Correct issues found	2	T12(M3)

Haseeb Malik(M1), Ali Arshad(M2), Abdul Hanan(M3)	T14	Deployment	3	T8(M1,M2), T11(M1, M2)
Ali Arshad(M2), Abdul Hanan(M3)	T15	System maintenance	2	T14(M1, M 2, M3)
Haseeb Malik(M1), Ali Arshad(M2), Abdul Hanan(M3)	T16	Evaluation	3	T13(M1), T14(M1,M 2,M3)

1.7.1. Work Breakdown Structure

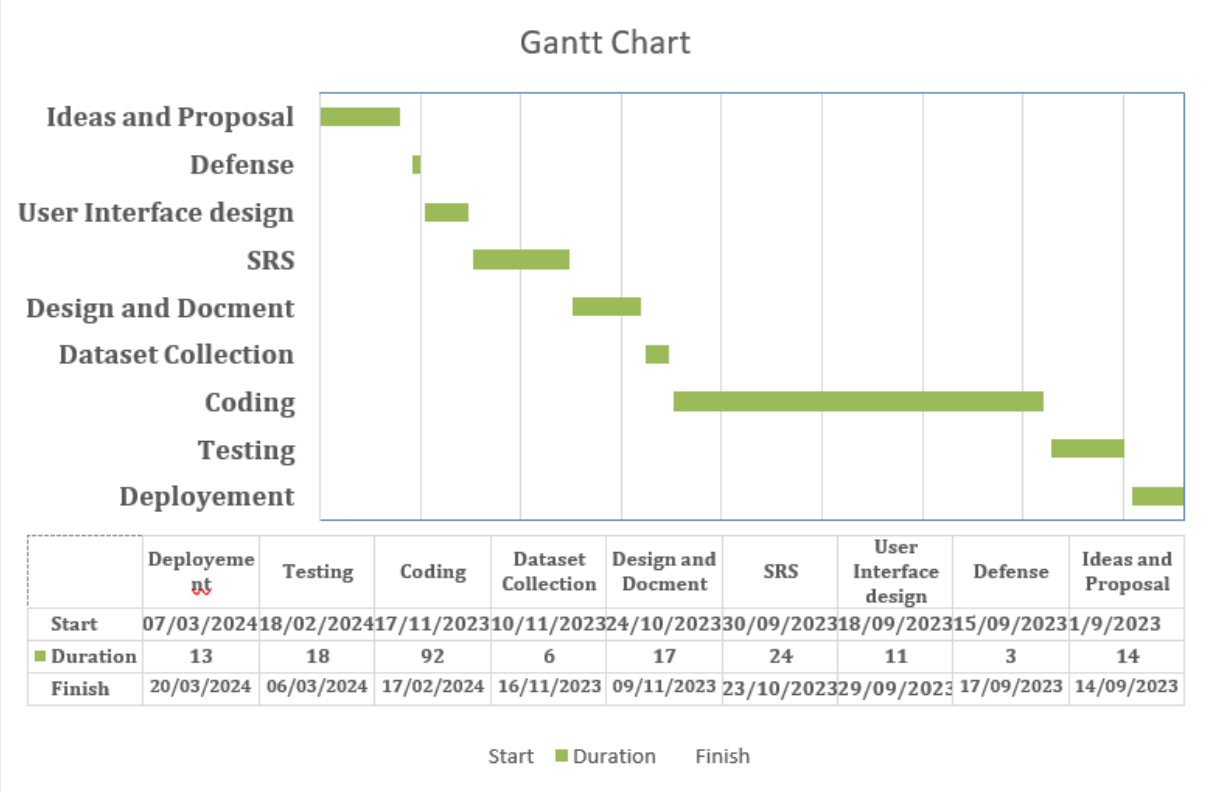


1.7.2. Roles & Responsibility Matrix

Haseeb Malik	**Ali Arshad**	**Abdul Hanan**
Project Initiation and Planning	UI design And Development	UI Design and Development
Dataset collection and Preprocessing on Images	Documentation of Project	Documentation of Project

Development of machine learning model	Contribution to the development of machine learning model	Collect Dataset of MRI Images
Testing and Validation of application	Integrate Modules	Contribution in Integrate Modules

1.7.3. Gantt Chart



	Deployement	Testing	Coding	Dataset Collection	Design and Docment	SRS	User Interface design	Defense	Ideas and Proposal
Start	07/03/2024	18/02/2024	17/11/2023	10/11/2023	24/10/2023	30/09/2023	18/09/2023	15/09/2023	1/9/2023
Duration	13	18	92	6	17	24	11	3	14
Finish	20/03/2024	06/03/2024	17/02/2024	16/11/2023	09/11/2023	23/10/2023	29/09/2023	17/09/2023	14/09/2023

1.8. Report Outline

This document covers all aspects of BRAIN TUMOR DETECTION which is AI based web-based system.

- Introduction
- Goals

Chapter 2

Software Requirement Specifications

Chapter 2: Software Requirement Specifications

2.1. Introduction

In this chapter we will cover all the Software Specification which is used in our project “Brain Tumor Detection”. We have Discuss all functions and non-functionalities used in this project also describe our Documentation conventions and which types of people this project is helpful. We will learn about the product perspective in which environment the project is capable to run and also explain its design and implementation.

2.1.1. Purpose

The product under consideration is the Brain Tumor Detect Web Application, which aims to provide a user-friendly platform for the detection and analysis of brain tumors. This SRS document covers the entire scope of the Brain Tumor Detect Web Application, detailing the functional and non- functional requirements necessary for its successful implementation. Document encompasses the entire system, including all modules and subsystems essential for the proper functioning of the application. The Brain Tumor Detect Web Application is intended to serve as a valuable tool for healthcare professionals, enabling them to upload medical imaging data, such as MRI scans, and receive accurate and timely analysis regarding the presence of brain tumors. The scope of this SRS includes the core functionalities related to image processing, tumor detection algorithms and result presentation.

2.1.2. Document Conventions

The First Letter of my new **paragraph** is **Capital** because it’s showing up the new paragraph starting from here and the only more thing I implemented here. That’s one is **Bold** the word. Where I did **Bold** the word, It means that’s word is more important in this paragraph. Use Calibri Text style and font size is 11.

Text Formatting:

- **Bold Text:** Indicates important keywords, section headings, and emphasized points.
- Italic Text: Used for placeholder names, variables, or to emphasize specific points.

Requirements Notation:

- Functional requirements are expressed using the following format: "FR-<UniqueIdentifier>: Description of the functional requirement."
- Non-functional requirements follow the format: "NFR-<UniqueIdentifier>: Description of the non-functional requirement."
- Use case names are presented as: "Use Case <UseCaseNumber>: Use Case Name."

Lists:

- Bulleted lists are used for items of equal importance.
- Numbered lists are used for sequential steps or priorities.

2.1.3. Intended Audience and Reading Suggestions

This Software Requirements Specification (SRS) document is intended for a diverse audience, each with specific roles and interests in the development and utilization of the Brain Tumor Detect Web Application. The following outlines the different types of readers for whom this document is relevant:

Intended Audience:

- **Developers:** Developers involved in the implementation of the Brain Tumor Detect Web Application will find detailed technical specifications, system architecture, and functional requirements. It provides insights into the software design, coding standards, and interfaces that need to be adhered to during the deveopment process.
- **Project Managers:** Project managers will benefit from understanding the

project scope, resource requirements, and timelines. They will find information about project constraints, risks, and dependencies. The document provides a basis for project planning, scheduling, and tracking.

- **Marketing Staff:** Marketing staff may be interested in understanding the application's features and capabilities, as well as any unique selling points. Insights into the user interface and user experience can assist in the development of marketing strategies. Additionally, any constraints that may affect marketing efforts are outlined.
- **Users:** End-users, including healthcare professionals and medical staff, will gain an understanding of the application's functionalities, user interface, and the overall user experience. This section addresses user authentication, input requirements, and the expected output from the system.
- **Testers:** Testers will find detailed information on test scenarios, test cases, and acceptance criteria. The document outlines the expected behavior of the application under various conditions, aiding in the creation of a comprehensive testing plan.

Sequence for Reading:

- **All Readers:** Start with the Overview section to gain a high-level understanding of the platform's goals and objectives.
- **Developers:** Proceed to the System Architecture section for technical details.
- **Project Managers:** Move on to the Project Timeline and Milestones section for information on project planning.
- **Marketing Staff:** Focus on the Marketing and User Acquisition section to understand strategies for promoting the platform.

- **Users:** Explore the User Experience and Onboarding section for guidance on using the platform effectively.

- **All Readers:** Examine the Security Features section for insights into the platform's security measures.

- **Testers:** Refer to the Testing Procedures section for details on testing processes.
- **Documentation Writers:** Finally, review the Documentation Guidelines section to understand how to create effective user documentation.

Organization of the Software Requirements Specification (SRS):

The SRS is organized into several sections, each serving a specific purpose:

- **Introduction -** Provides an overview of the document, its purpose, and the intended audience.

- **Overall Description -** Offers a high-level description of the Brain Tumor Detect Web Application, including product perspective, features, and user characteristics.
- **Specific Requirements -** Details functional and non-functional requirements, including system features, external interfaces, and performance requirements.
- **System Models -** Provides visual representations of the system through use case diagrams, activity diagrams, and other relevant models.
- **User Interface Design** - Describes the look and feel of the user interface, including mock-ups and design considerations. Section

2.1.4. Product Scope

MRI images are classified in the past with heavy systems and time-consuming methods. But in our system, MRI would be classified in real time with low processing power. Our project uses the latest technologies to help researchers in MRI classification within a cell.

- Web Based Application

  It will allow user to access the interface to analyze MRI images.

- Classification of human Brain

  MRI image classification of human brain would be done with image processing using deep neural network Convolution Neural Networks (scikit learn, TensorFlow, koras).

- Disease Prediction

  The system will be able to predict specific disease on the bases of MRI image that will be detected/classified by image classification using deep neural networks.

2.1.5. References

1. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC10453020/
2. https://www.cancer.net/cancer-types/brain-tumor/diagnosis
3. https://www.hindawi.com/journals/jhe/2022/2693621/
4. https://www.mdpi.com/2075-1729/13/7/1449

2.2. Overall Description

2.2.1. Product Perspective

The Brain Tumor Detect Web Application is designed as a standalone product, representing a new and innovative solution in the domain of medical imaging and healthcare. It is not a replacement for any existing system but rather introduces a novel approach to facilitate the

detection and analysis of brain tumors through advanced image processing techniques. Context and Origin: Standalone Product: The Brain Tumor Detect Web Application is conceived as an independent software solution with a specific focus on brain tumor detection. It is not part of an existing product family but serves as a unique and dedicated tool for healthcare professionals. Innovative Solution: Originating from the increasing demand for advanced medical imaging technologies, the Brain Tumor Detect Web Application leverages cutting-edge algorithms to enhance the accuracy and efficiency of brain tumor diagnosis. Its development is driven by the need for a reliable, user-friendly, and accessible platform for medical practitioners.

2.2.2. User Classes and Characteristics

The Brain Tumor Detect Web Application caters to diverse user classes with distinct roles, responsibilities, and characteristics. Understanding the varied user needs is crucial for tailoring the application to provide an optimal experience. The identified user classes are as follows:

Healthcare Professionals:

- **Characteristics:**
  - High frequency of use.
  - Extensive medical expertise and knowledge.
  - Regular interaction with medical imaging data.

- **Roles:**
  - Radiologists, neurologists, and oncologists.
  - Responsible for interpreting and diagnosing brain imaging data.

System Administrators:

- **Characteristics:**
  - Moderate to high technical expertise.
  - Responsible for system configuration and maintenance.
  - May have access to system logs and user management features.
- **Roles:**
  - IT professionals managing the deployment and maintenance of the application.

Data Entry Personnel:

- **Characteristics:**
  - Moderate technical expertise.
  - Responsible for uploading patient data and medical images.
  - May have basic medical knowledge for data validation.
- **Roles:**
  - Administrative staff entering patient information into the system.

End Users (Medical Staff):

- **Characteristics:**
  - Varied levels of technical expertise.
  - Use the application for specific tasks such as viewing reports.

- Limited interaction with advanced features.

- **Roles:**
  - Nurses and other medical staff who need access to diagnostic reports.

Patients (Limited Access):

- **Characteristics:**
  - Limited technical expertise.
  - Access the application to view personal medical reports.
  - Minimal interaction with diagnostic tools.
- **Roles:**
  - Patients who have been granted limited access for personal health monitoring.

2.2.3. Operating Environment

The system operates on the anaconda prompt, windows operating system, visual studio code Running the front end on HTTP server.

2.2.4. Design and Implementation Constraints

The Brain Tumor Detection System, utilizing ResNet50 AI architecture, aims to provide an efficient tool for early tumor diagnosis. Constraints include computational resource limitations, the need for diverse labeled datasets, and adherence to regulatory and ethical standards. Challenges in model interpretability, real-time processing, scalability, and integration with existing systems must be addressed. The system's timeline highlights key milestones, and risk mitigation strategies are identified. Ultimately, the system aims to enhance medical diagnostics through the integration of AI in neuroimaging.

2.2.5. Assumptions and Dependencies

Assumptions:

- **Availability of External Databases:** Assumption: The application assumes that external databases, if integrated, are accessible and maintained with updated and accurate medical information. Changes in the availability or structure of these databases may affect data retrieval.
- **User Internet Connectivity:** Assumption: End-users are assumed to have reliable internet connectivity when accessing the web application. Limited or unreliable internet access may impact the user experience and data upload/download processes.
- **Regulatory Compliance:** Assumption: The development and deployment of the application assume adherence to relevant healthcare data privacy and security regulations. Any changes in regulatory requirements may necessitate adjustments to the application. Integration with External

Dependencies:Third-Party Libraries and Tools: Dependency: The project relies on specific third-party libraries and tools for image processing and analysis. Changes or discontinuation of these tools may necessitate adjustments to the application's algorithms.

- **Web Browser Compatibility:** Dependency: The application is dependent on the compatibility of web browsers with its frontend components. Changes in browser behavior may impact user experience.
- **Server Infrastructure:** Dependency: The reliable operation of the application is dependent on the availability and proper configuration of the server infrastructure. Downtime or configuration issues may affect the application's accessibility.
- **Operating System Updates:** Dependency: The application's server-side

operation is dependent on the continued support and compatibility of the chosen operating system. Updates or changes to the operating system may require corresponding updates to the application.

- **Regulatory Changes:** Dependency: The project is dependent on compliance with healthcare regulations. Changes in regulations may necessitate modifications to the application to ensure continued compliance.

2.3. External Interface Requirements

2.3.1. User Interfaces

The user interface for the Brain Tumor Detection System is web-based and developed using the Flask framework. Users access the system through an intuitive dashboard that provides a comprehensive overview of system status and recent activities. The interface facilitates the seamless upload of medical imaging data, employing preprocessing tasks for optimal compatibility. Tumor detection results are presented through annotated images, probability scores, and relevant metrics, allowing users to interact with and analyze the outcomes. Secure user authentication and authorization are implemented, and administrators can manage system configuration through a dedicated interface.

2.3.2. Hardware Interfaces

The Brain Tumor Detection System requires standard computing hardware with multi-core processors, GPUs, and ample RAM for optimal performance of the ResNet50 AI architecture. It is adaptable to various storage solutions, including SSDs. External interfaces support seamless integration with medical imaging devices like MRI scanners and PACS. The hardware is selected to efficiently process large medical imaging datasets during training and inference, ensuring accurate and timely brain tumor detection.

2.3.3. Communications Interfaces

The Brain Tumor Detection System relies on HTTP and HTTPS protocols for secure communication through a web-based interface. Email notifications update users on progress, and network server protocols handle data transfer. Encryption (SSL/TLS) secures sensitive data during transmission. Electronic forms ensure secure user authentication. The system prioritizes efficient data transfer rates and employs synchronization mechanisms for reliable interactions.

2.4. System Features

2.4.1. System Feature 1

Input MRI Image: Enable users to Input MRI Image.

Image Segmentation: Apply image segmentation on image using neural network approaches.

Feature Extraction: System Perform feature Extraction on segmentation image using ResNet Model.

Classification Images: Combine the extracted features from both the ResNet model and the neural network to create a fused feature representation for robust and accurate classification

Disease Prediction: Utilize the integrated machine learning model to predict the presence of brain tumors based on the combined feature vectors, leveraging the power of both ResNet and neural network architectures.

Report Generation: Dynamically generate comprehensive diagnostic reports summarizing the classification results, including features identified by both ResNet and the neural network.

2.4.1.1 System Features Description and Priority

1. **Input MRI Image:**
    a) **Description:** Authorized users, predominantly healthcare professionals, can securely upload MRI images into the system for analysis.
    b) **Priority:** High

2. **Image Segmentation:**
    a. **Description:** Employ state-of-the-art image segmentation algorithms, including deep learning techniques, to accurately identify and isolate regions of interest within the MRI images..
    b. **Priority:** High

3. **Feature Extraction:**
    a. **Description:** Integrate a ResNet model to effectively extract hierarchical features from segmented MRI images, capturing complex patterns and spatial relationships within the data.
    b. **Priority:** High

4. **Image Classification:**
    a. **Description:** Combine the extracted features from both the ResNet model and the neural network to create a fused feature representation for robust and accurate classification.
    b. **Priority:** High

5. **Predict Disease:**
    a. **Description:** Utilize the integrated machine learning model to predict the presence of brain tumors based on the combined feature vectors, leveraging the power of both ResNet and neural network architectures.
    b. **Priority:** High

6. **Show Result:**
    a. **Description:** Dynamically generate comprehensive diagnostic reports summarizing the classification results, including features identified by both ResNet and the neural network.
    b. **Priority:** High

2.4.1.2 Stimulus/Response Sequences

1. **Stimulus: User Uploads MRI Image**
   - **Response:** The system receives the uploaded MRI image.
   - **Sequence:**
     - **Stimulus:** Authorized healthcare professionals initiate the image upload process.
     - **Response:** The system validates the image format, ensuring compatibility with the required standards (e.g., DICOM).
     - **Response**: The uploaded image is securely stored within the system.

2. **Stimulus: Automated Image Segmentation**
   - **Response:** The system performs automated image segmentation.
   - **Sequence:**
     - **Stimulus:** The segmented MRI image is input into the ResNet model and neural network.
     - **Response:** The system employs advanced segmentation algorithms to identify and differentiate brain tissues and potential tumor regions.
     - **Response:** Segmentation results are generated, highlighting regions of interest for further analysis.
3. **Stimulus: Feature Extraction Using ResNet and Neural Network**
   - **Response:** The system extracts features from the segmented MRI image using ResNet and a neural network.
   - **Sequence:**
     - **Stimulus:** Feature extraction involves utilizing the hierarchical features learned by the ResNet model and neural network.
     - **Response:** The system creates comprehensive feature vectors, capturing nuanced patterns and spatial relationships within the MRI data.
     - **Response:** Extracted features from both models are combined to form a fused feature representation.

4. **Stimulus: Image Classification and Disease Prediction**
   - **Response:** The system classifies the MRI image and predicts the presence of a brain tumor.
   - **Sequence:**
     - **Stimulus:** The fused feature vector is input into the integrated machine

learning model.

- **Response:** The system leverages the combined features for accurate classification, utilizing the power of both ResNet and the neural network.
- **Response:** The predicted disease status, along with probability scores, is generated.

5. **Stimulus: Result Presentation to User**
   - **Response:** The system presents diagnostic reports and visual representations to the user.
     - **Sequence:Stimulus: Healthcare professionals access the diagnostic results through the user interface.**
     - **Response:** The system dynamically generates comprehensive diagnostic reports, summarizing classification results and identified features.
     - **Response:** Visual representations, such as annotated MRI images and heatmaps, are presented to aid in the evaluation process.

2.1.1.1. Functional Requirements

1. MRI Image Input

Functional Requirements:

1.1. **User Uploads MRI Image:**

- The system shall provide a secure and user-friendly interface for authorized healthcare professionals to upload MRI images.

1.2. **Image Format Validation:**

- The system shall validate uploaded MRI images to ensure they adhere to standardized formats, such as DICOM, for compatibility and accurate processing.

Image Segmentation

Functional Requirements:

2.1. **Automated Segmentation:**

- The system shall employ advanced image segmentation algorithms

to automatically identify and isolate regions of interest within the MRI images.

2.2. **Tissue Differentiation:**

- The system shall differentiate brain tissues through segmentation, distinguishing normal tissues from potential tumor regions using neural network-based approaches.

3. Feature Extraction

Functional Requirements:

3.1. **ResNet Model Integration:**

- The system shall integrate a ResNet model to extract hierarchical features from segmented MRI images.

3.2. **Neural Network Feature Extraction:**

- The system shall utilize a neural network architecture to enhance feature extraction, allowing the model to learn and represent intricate patterns indicative of tumor characteristics.

3.3. **Feature Vector Generation:**

- The system shall formulate feature vectors containing high-level representations derived from both ResNet and the neural network for subsequent classification.

4. Image Classification

Functional Requirements:

4.1. **Machine Learning Model Fusion:**

- The system shall combine extracted features from both the ResNet model and the neural network to create a fused feature representation for robust and accurate classification.

4.2. **Disease Prediction:**

- The system shall utilize the integrated machine learning model to predict the presence of brain tumors based on the combined feature vectors.

5. Result Presentation

Functional Requirements:

5.1. **Diagnostic Report Generation:**

- The system shall dynamically generate comprehensive diagnostic reports summarizing the classification results, including features identified by both ResNet and the neural network.

5.2. **Visual Representation:**

- The system shall present visual representations, such as annotated MRI images or heatmaps, to highlight areas of interest and support clinicians in their evaluation.

5.3. **Transparent Probability Scores:**

- The system shall provide probability scores associated with the classification results, offering transparency regarding the confidence of the prediction derived from both ResNet and the neural network.

Requirements Trace-ability Matrix

Sr #	No.	Requirements	Builds	Use-Case-Name	Categories
1	1.0	A System "Shall" allow the user to TURN_ON	B1	UC-Turn-On	Business

2	2.0	A system "Shall" allow the user to input MRI images.	B2	UC-Input-Image	Medical
3	3.0	The system "shall" do segmentatio non image	B3	UC-Image-Segmentation	Medical
4	4.0	The system "shall" extract the feature from Images.	B4	UC-Feature-Extraction	Medical
5	5.0	The system "shall" predict the disease.	B5	UC-Predict-Disease	Medical
6.0	6.0	The system "shall" classify The Brain Tumor.	B6	UC-Classify-Images	Medical
7	7.0	The system "Shall" visual the result.	B7	UC-Visual-Result	Medical
8	8.0	The system "shall" show result on the screen in real Time.	B8	UC-Show-Result	Medical
9	*9.0*	The System "Shall" show Essential instruction for disease.	B9	UC-Show-Instruction	Medical

10	10.0	A system "Shall" allow the user to TURN_OFF	B10	UC-Turn-Off	Business

2.5. Other Nonfunctional Requirements

No.	Name	Non-Functional	Description	Actor
1.	Privacy	Yes	Ensure user privacy is maintained securely	Admin
2.	Validity	Yes	Handle errors effectively for a robust system	Admin
3.	Performance	Yes	Optimize application performance for efficiency	Admin
4.	Usability	Yes	Ensure the platform is user-friendly for everyone	Admin
5.	Reliability	Yes	Build trustworthiness in the system for users	Admin
6.	Security	Yes	Implement robust security measures for data protection	Admin
7.	Scalability	Yes	Design the platform to scale with increasing users and data	Admin
8.	Availability	Yes	Ensure high availability for uninterrupted service	Admin
9.	Compatibility	Yes	Make the platform compatible with various devices and browsers	Admin
10.	Compliance	Yes	Adhere to legal and regulatory compliance standards	Admin

2.5.1. Performance Requirements

The performance of the Brain Tumor Detect Web Application is critical to its effectiveness in facilitating accurate and timely brain tumor detection. The following performance requirements have been established to ensure a responsive, scalable, and efficient user experience:

Response Time:

- Rationale: Quick response times are essential for healthcare professionals using the application to efficiently review patient data and diagnostic results. The

specified timeframes aim to optimize user productivity and maintain an effective workflow.

Throughput:

- Requirement: The application should support a minimum of 100 concurrent users without a degradation in performance.
- Rationale: As the number of users accessing the application concurrently may vary, the system needs to be scalable to accommodate increasing user demands. Ensuring consistent performance under load is essential to meet the diverse needs of healthcare professionals.

Scalability:

- Requirement: The system architecture must be designed to scale horizontally to handle an increasing number of users and a growing dataset of medical images.
- Rationale: Scalability is crucial for accommodating the potential growth in user base and medical data. A scalable architecture ensures that the application remains responsive and efficient as usage and data volume increase over time.

Real-Time Requirements:

- Requirement: For real-time image processing tasks, the system must adhere to the specified response time requirements, taking into account the time-sensitive nature of medical diagnoses.
- Rationale: Real-time processing is vital for time-sensitive tasks such as brain tumor detection. Meeting the specified real-time requirements is critical to ensuring that medical professionals receive timely and accurate results for prompt decision-making.

2.5.2. Usability Requirements

- **User Interface Consistency:**

  The user interface should maintain consistency across different sections of the application, promoting a seamless and intuitive user experience.

2.5.3. Reliability Requirements

The reliability requirements for the Brain Tumor Detect Web Application are designed to ensure a robust and uninterrupted user experience for healthcare professionals relying on the platform. One of the primary requirements is a high level of availability, mandating that the application must maintain an uptime of at least 99.5% over any given month. This reliability standard is crucial for healthcare practitioners who depend on constant access to patient records and diagnostic outcomes. In addition to availability, the application is required to demonstrate fault tolerance by gracefully handling unexpected errors or failures without compromising the integrity of patient data. This resilience ensures that the system remains operational even in the face of unforeseen issues, minimizing disruptions in service delivery. Data integrity is another key aspect of reliability, emphasizing the implementation of robust checks to detect and prevent data corruption or loss. Maintaining the accuracy of patient data is fundamental to the application's reliability, ensuring that diagnostic results presented to healthcare professionals are trustworthy

2.5.4. Maintainability/supportability Requirements

The application's code adheres to best coding scripts, promoting ease future advancement. Comprehensive validation, including code comments and user manuals, is maintained to facilitate the understanding of the system by developers, administrators, and end-users.

2.5.5. Portability Requirements

The portability requirements of the Brain Tumor Detect Web Application focus on ensuring adaptability and seamless deployment across diverse computing environments. The application is designed to be platform- independent, allowing it to run consistently on various operating systems, including Linux, Windows, and macOS. This flexibility ensures that healthcare professionals can access the application on their preferred devices, regardless of the underlying operating system. To enhance accessibility, the application must also exhibit browser compatibility. It should perform optimally on popular web browsers such as Google Chrome, Mozilla Firefox, Safari, and Microsoft Edge. This requirement ensures a consistent user experience across different browser platforms, enabling healthcare practitioners to utilize the application on their browsers of choice.

2.6. Efficiency Requirements

Efficiency is a crucial aspect of the Brain Tumor Detect Web Application to ensure that computational resources are utilized optimally, resulting in a streamlined and responsive user experience. The following efficiency requirements have been identified:

- **Algorithmic Efficiency:**

  Requirement: The image processing algorithms employed for brain tumor detection must be optimized for efficiency, minimizing computational time while maintaining high accuracy.

  Rationale: Optimized algorithms are essential for timely analysis of medical imaging data. The efficiency of these algorithms directly impacts the overall responsiveness of the application, influencing user satisfaction and clinical decision-making.

- **Resource Utilization:**

Requirement: The application should effectively utilize server resources, including CPU and memory, to handle image processing tasks efficiently and support concurrent user interactions.

Rationale: Efficient resource utilization is critical for maintaining system responsiveness under varying workloads. The application should dynamically allocate resources to prioritize real-time image processing tasks while ensuring a smooth user experience for all users.

- **Data Storage Efficiency:**

  Requirement: The storage and retrieval of medical imaging data should be optimized for efficiency to minimize latency in accessing patient records.

  Rationale: Efficient data storage and retrieval are essential for quick access to patient information. This requirement aims to minimize delays in accessing medical records, enhancing the overall speed and efficiency of the application.

- **Caching Mechanism:**

  Requirement: Implement a caching mechanism to store frequently accessed data, such as diagnostic results and patient records, to reduce the need for redundant computations.

  Rationale: Caching commonly accessed data reduces the need for repetitive processing, improving response times and decreasing the computational load on the server.

- **User Interface Rendering:**

  Requirement: The user interface components, including dashboards and reports, must be optimized for rendering speed to provide a seamless and responsive user experience.

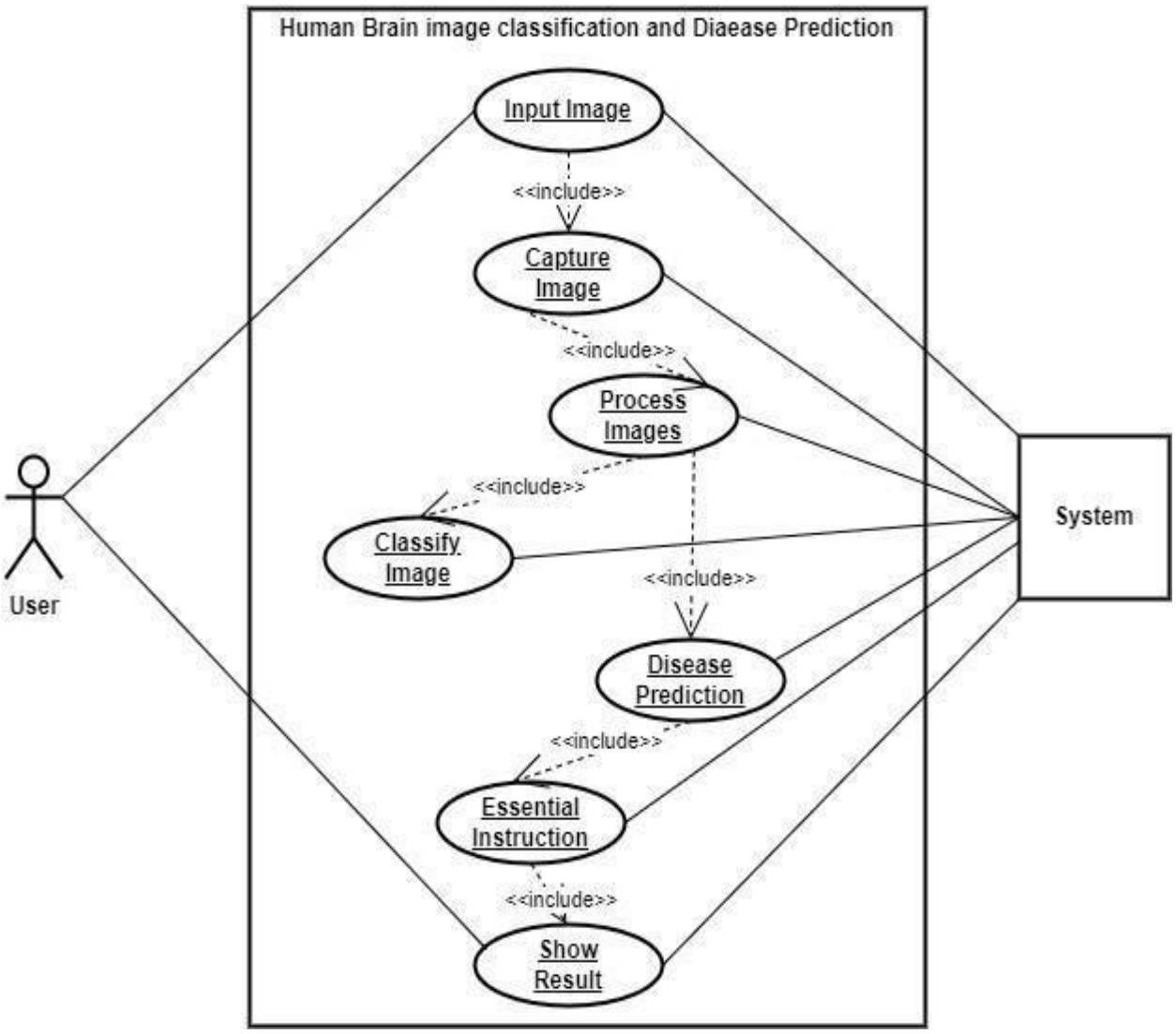
Rationale: A visually responsive interface is crucial for effective user interaction. Optimizing the rendering speed of user interface elements ensures a smooth and efficient experience for healthcare professionals accessing diagnostic information.

Chapter 3

Use Case Analysis

Chapter 3: System Analysis

3.1. Use Case Model



3.2. Use Case Descriptions

USE CASE Description	
Name	**Input Image**
Brief Description	Input the image and get permissions for the camera's from theSystem.
Actors	User, System
Pre-Condition	System should be active
Post Condition	System starts accepting Images

USE CASE Description	
Name	**Capture Images**
Brief Description	System take image from user and took apart into images
Actors	System
Pre-Condition	Must have the image
Basic Flow	Get the image Breakdown image in to frames Each frame is segmented by respective technique according toimage edges
Post Condition	Segmented frames send to Feature extraction

USE CASE Description	
Name	**Classify Images**
Brief Description	System classifies the images either is Brain Tumor or not.
Actors	System
Pre-Condition	Must have Processed Images
Basic Flow	System gets the Processed Images displays the detected Result
Post Condition	Frames send to disease prediction.

USE CASE Description	
Name	**Disease Prediction**
Brief Description	System predicts the disease base on input images.
Actors	System
Pre-Condition	Must have Processed Images
Basic Flow	System gets the Processed Images System displays the detected result.System show result to the user.

Post Condition	frame send to disease essential instruction

USE CASE Description	
Name	**Essential Instruction**
Brief Description	Essential instruction for the Brain tumor disease.
Actors	System
Pre-Condition	Must have Processed Images
Basic Flow	System gets the Processed Images System displays the detected result. System displays the instruction to the user.
Post Condition	frames send to final output.

USE CASE Description	
Name	**Show Result**
Brief Description	System shows result of the disease prediction and essential instruction.
Actors	User, System
Pre-Condition	System must have the results to show user.

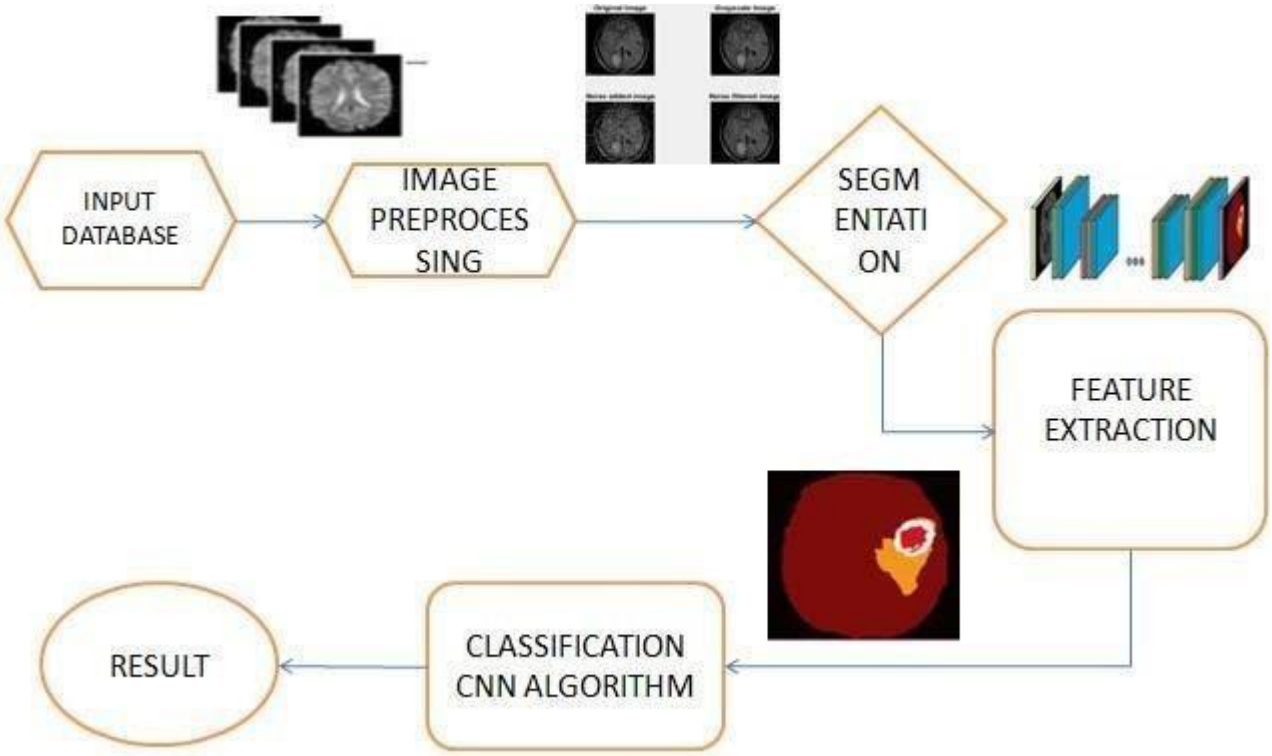
Basic Flow	System gives the results
Exception Flow	System gives the error notification for respective features

Chapter 4

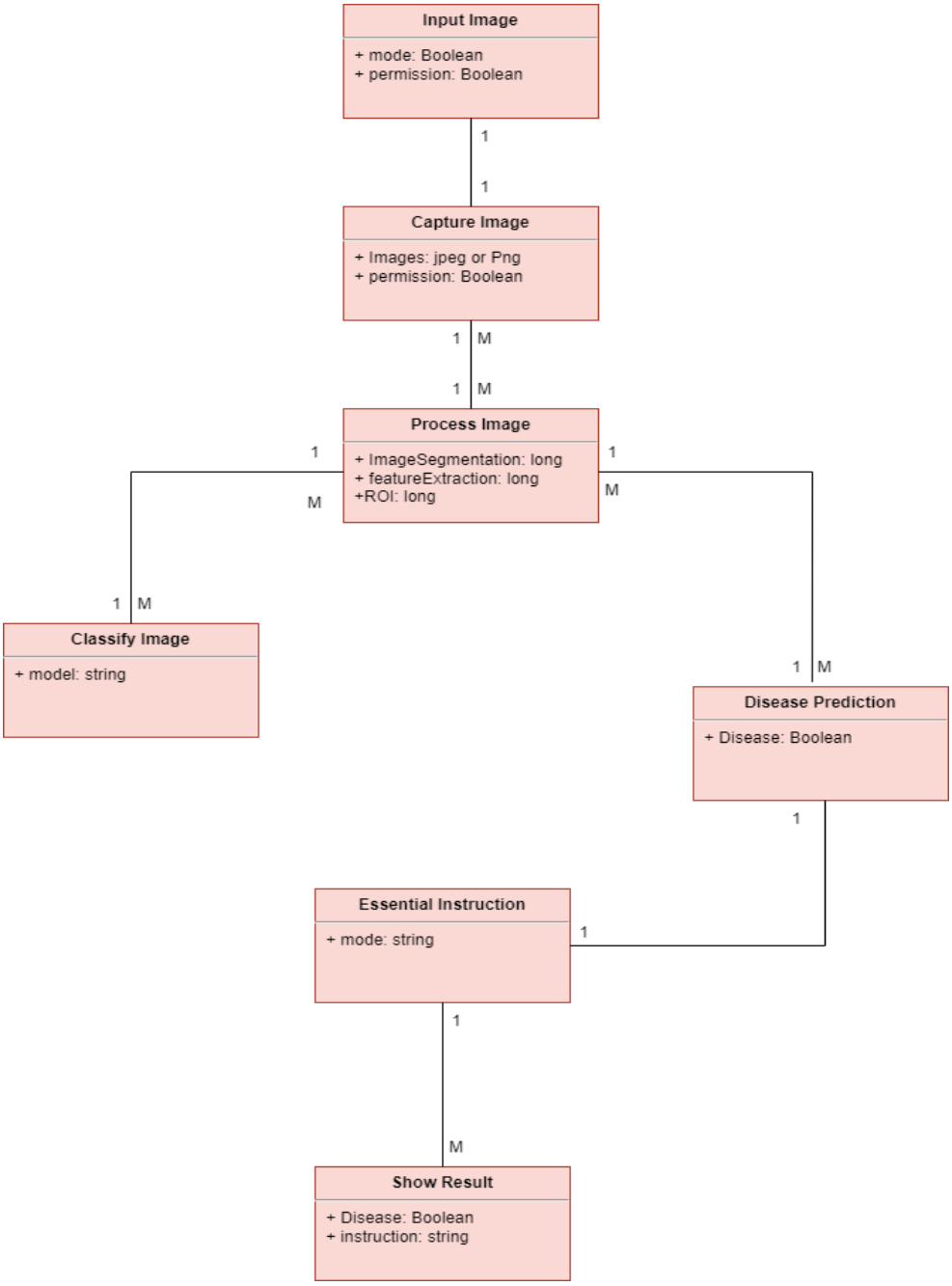
System Design

Chapter 4: System Design

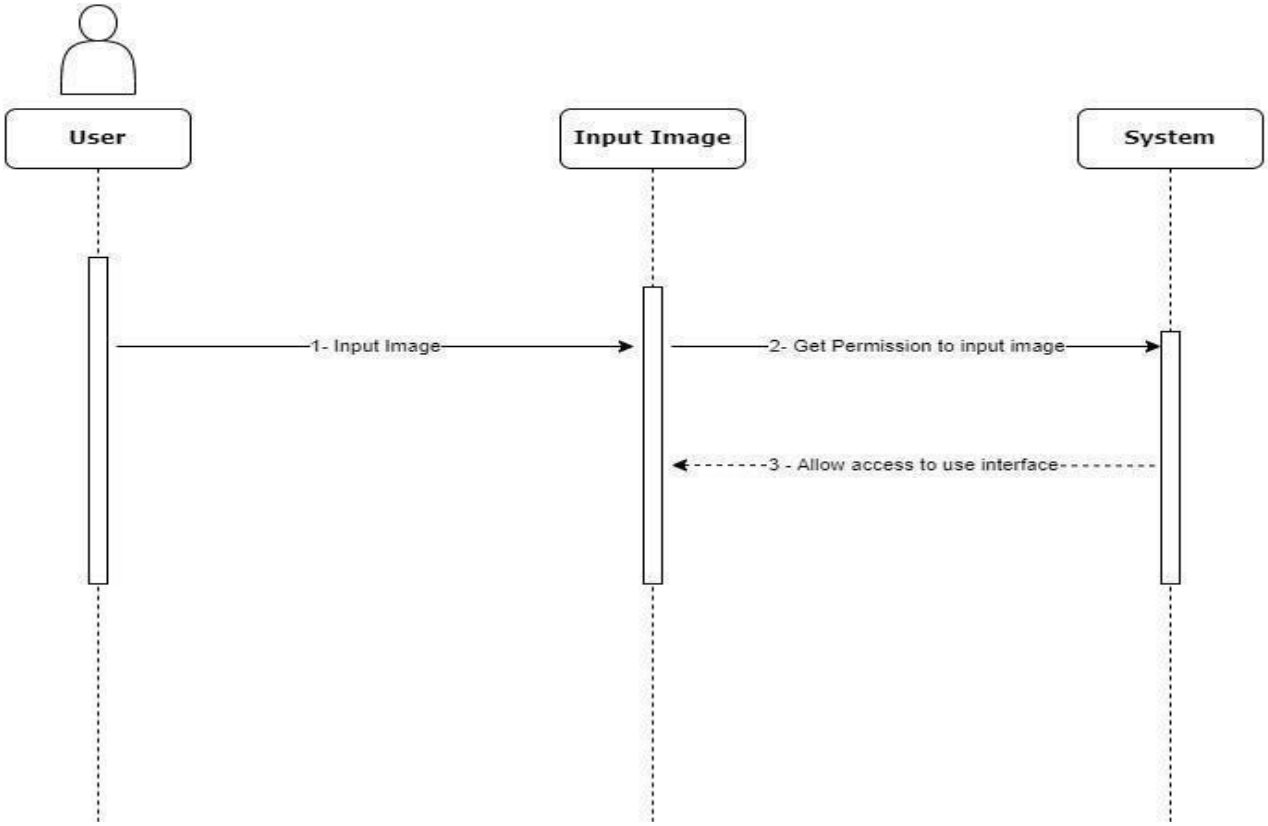
4.1. Architecture Diagram



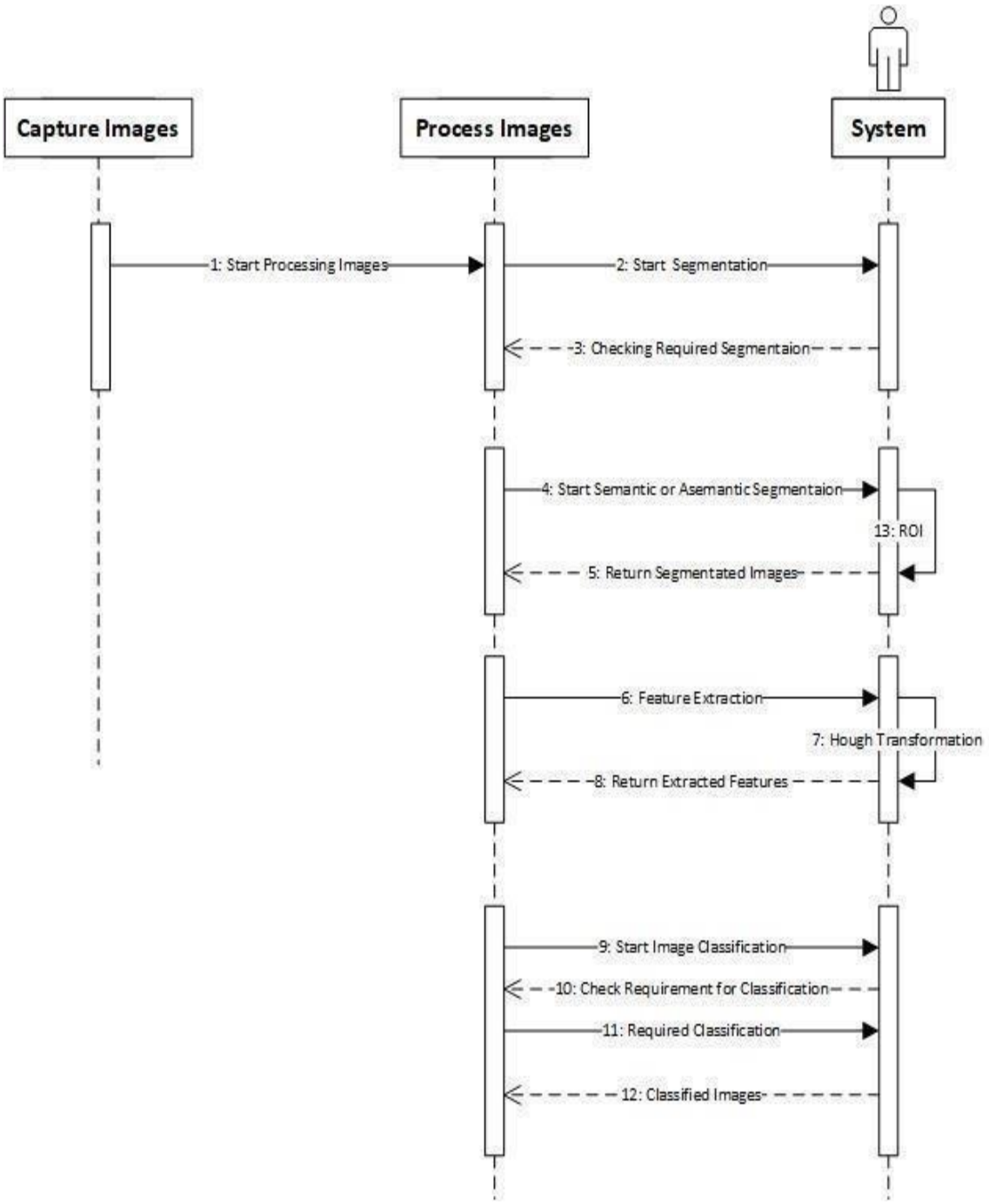
4.2. Domain Mod



4.3. Sequence Diagram



User input



Process Image

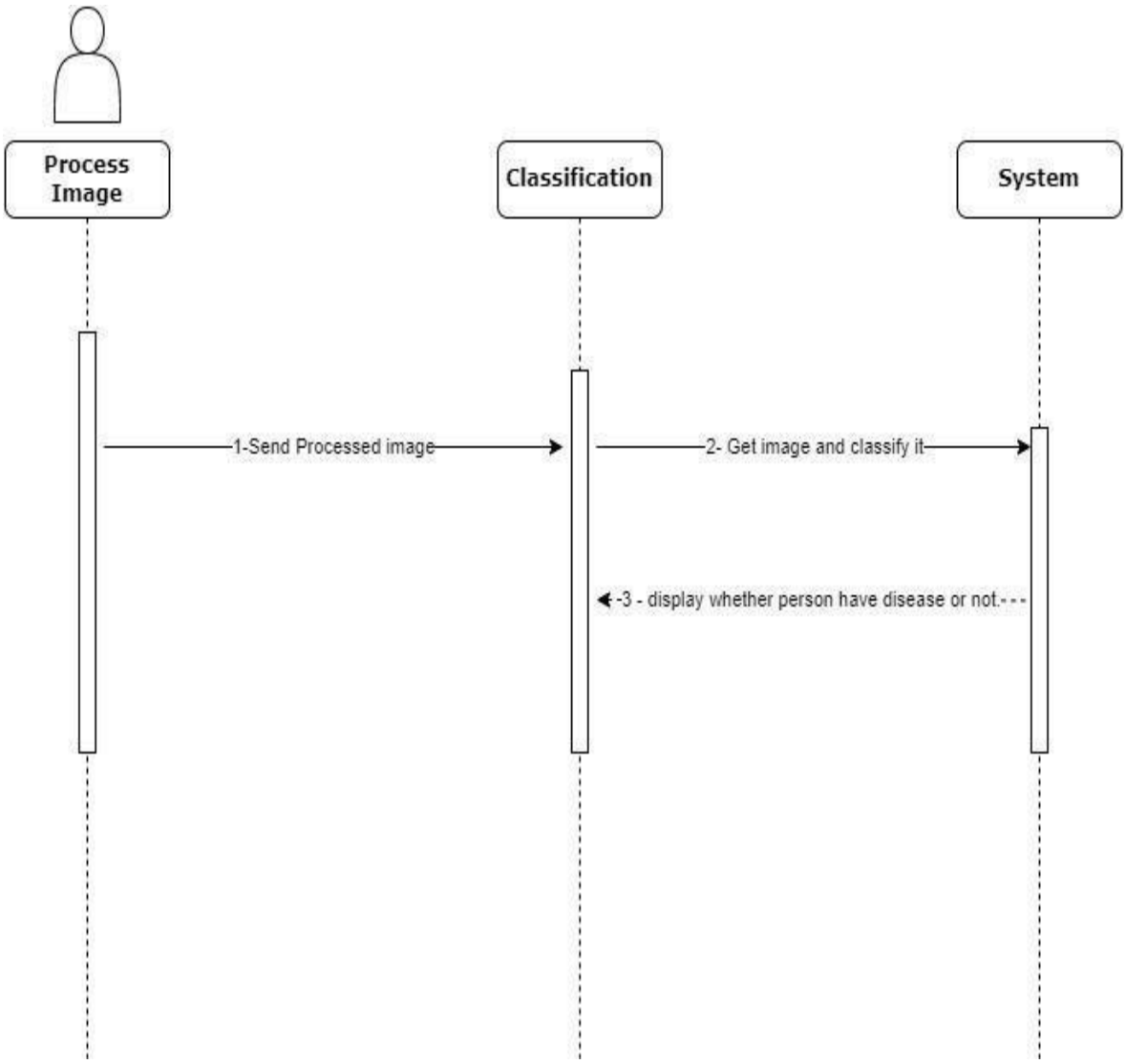
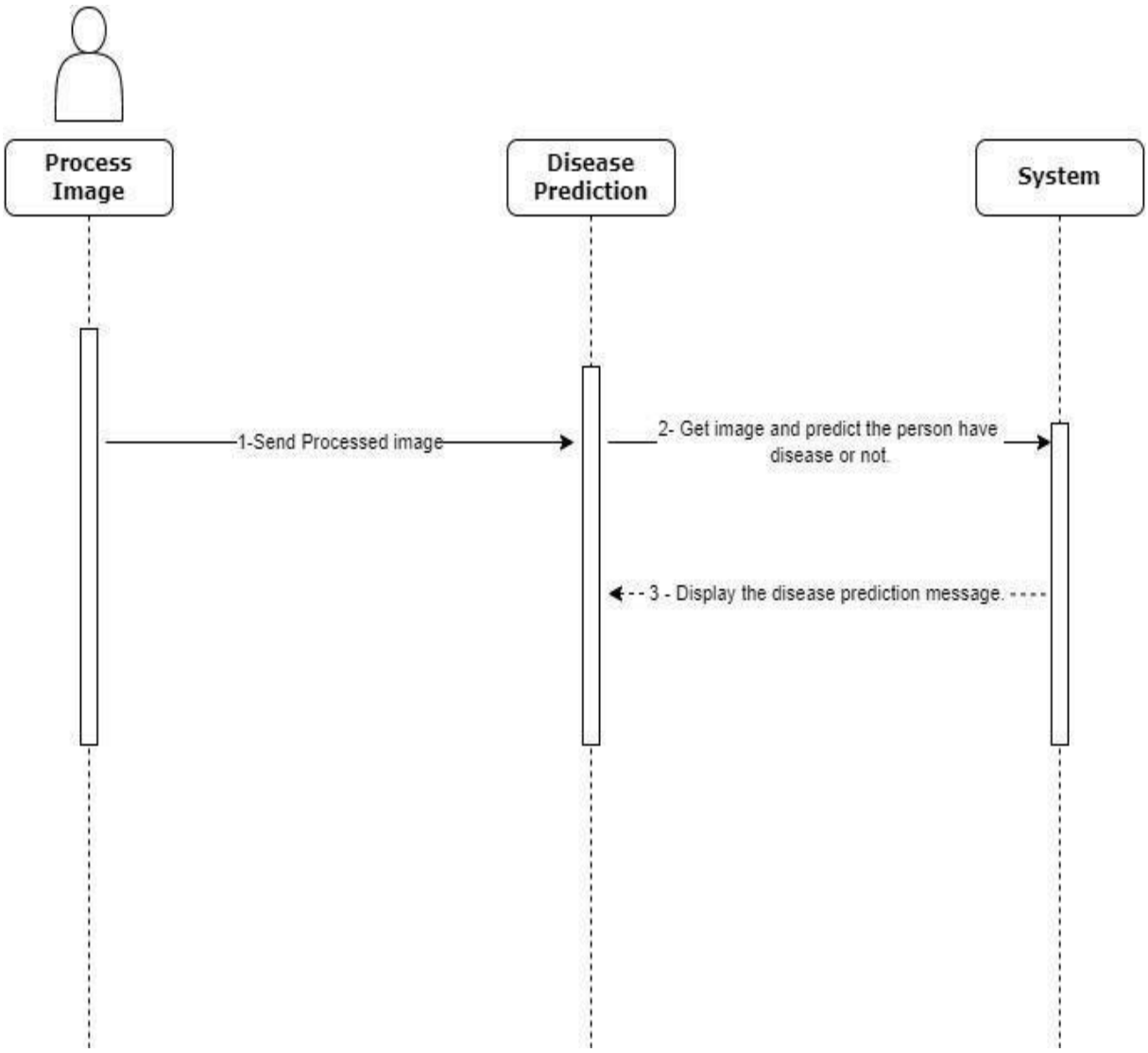
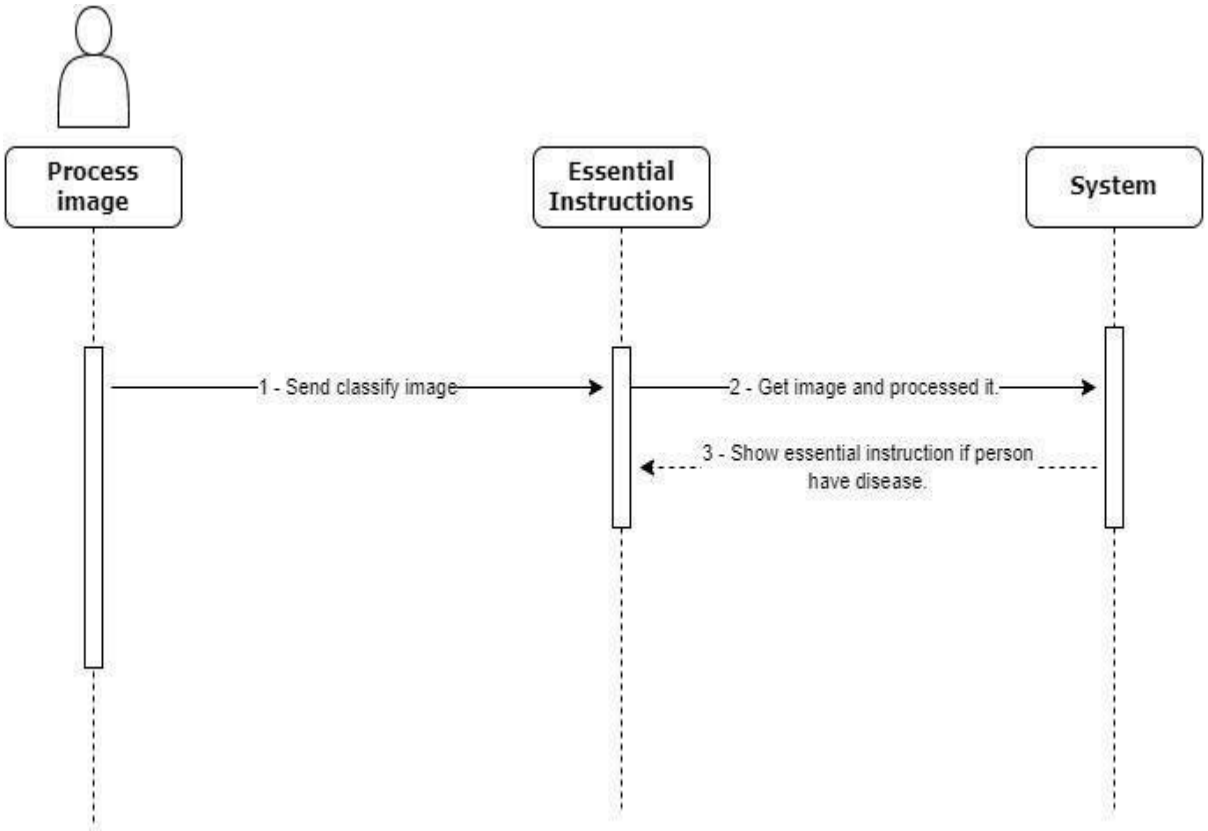


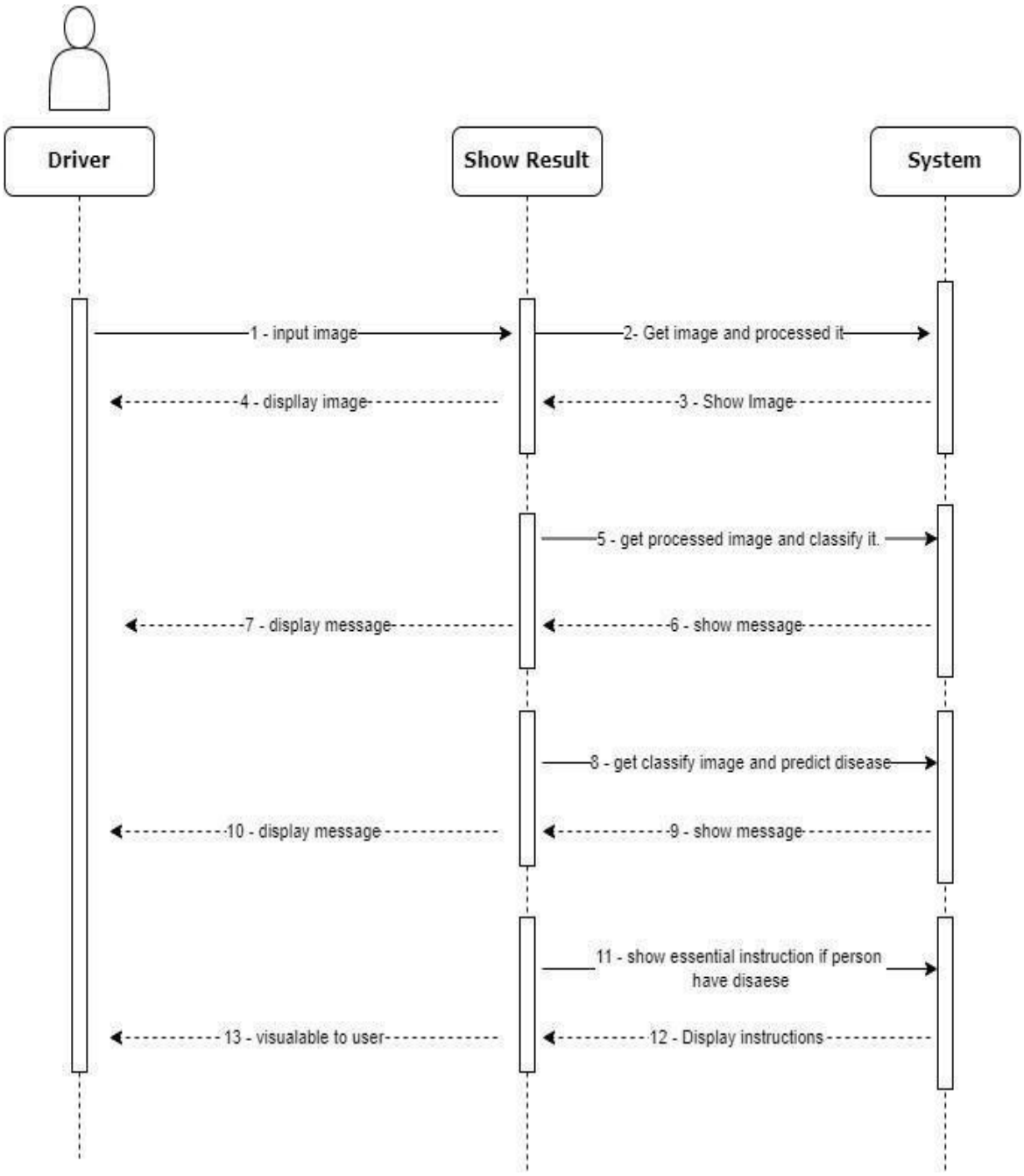
Image Classification



Disease Prediction

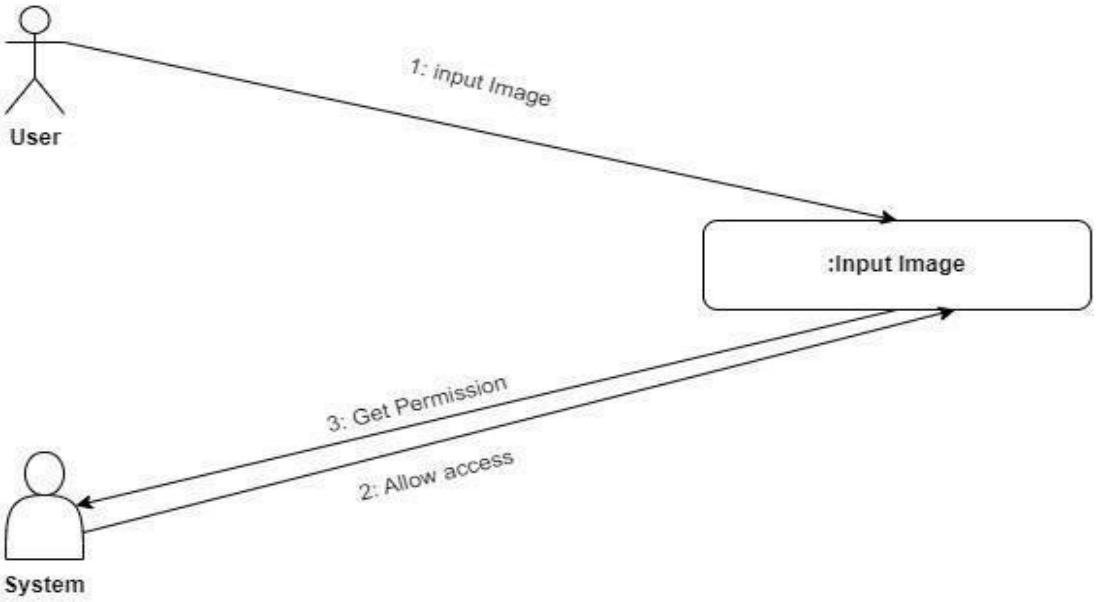


Essential Instructions

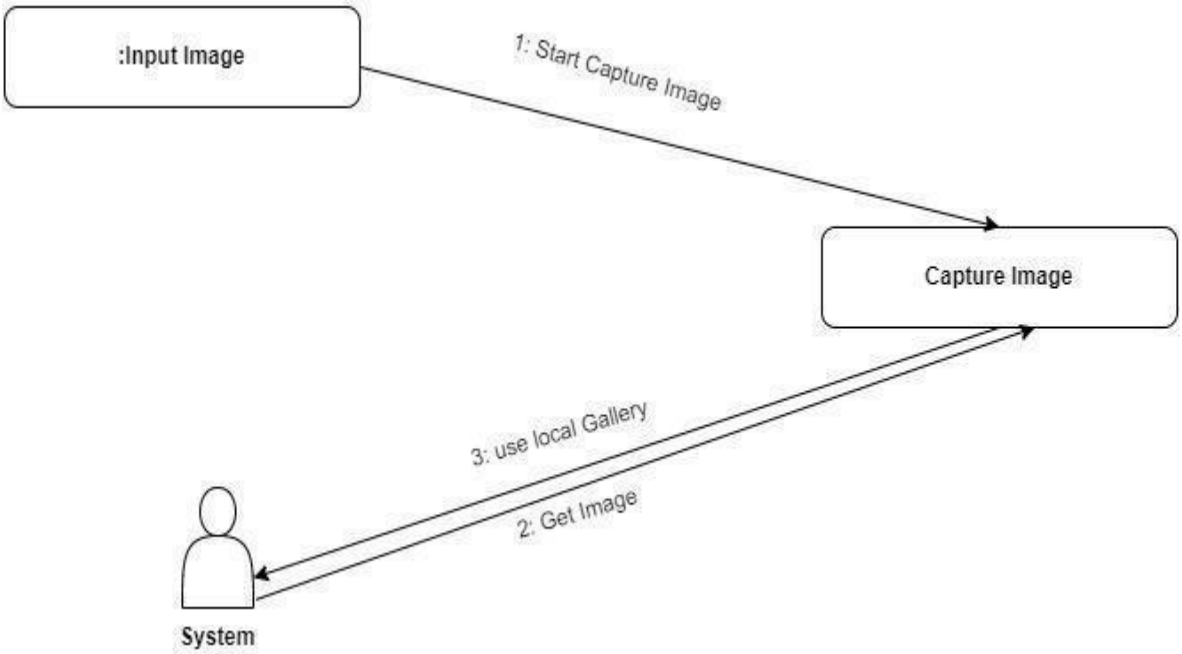

Driver
Show Result
System
1 - input image
2- Get image and processed it
4 - displlay image
3 - Show Image
5 - get processed image and classify it.
7 - display message
6 - show message
8 - get classify image and predict disease
10 - display message
9 - show message
11 - show essential instruction if person have disaese
13 - visualable to user
12 - Display instructions

4.4. Collaboration Diagram

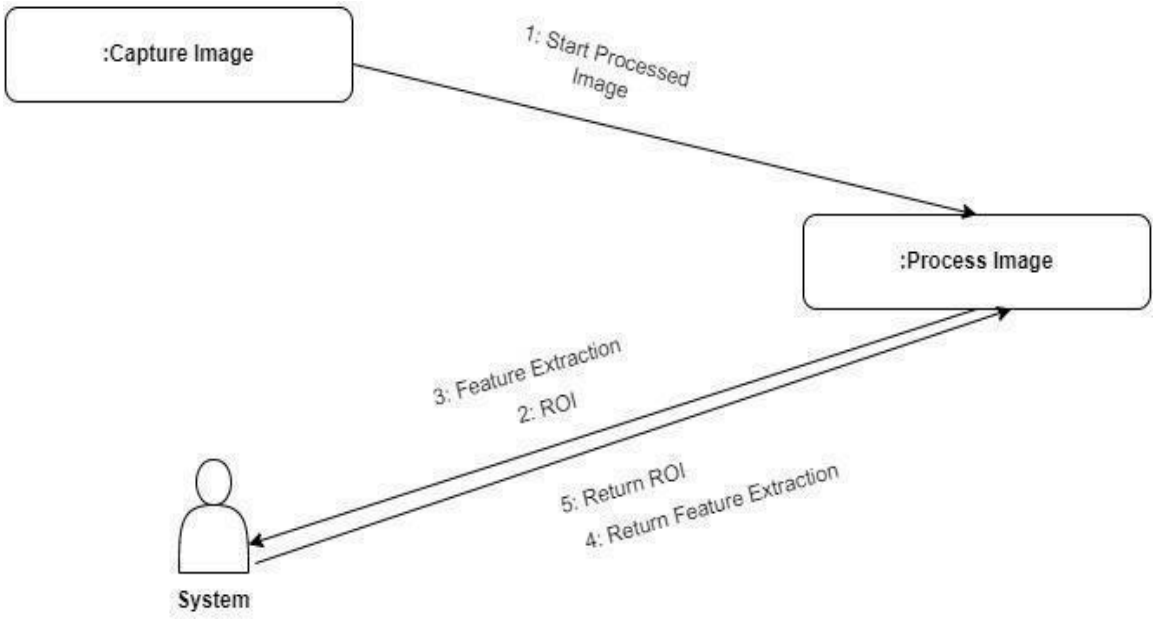
Input Image



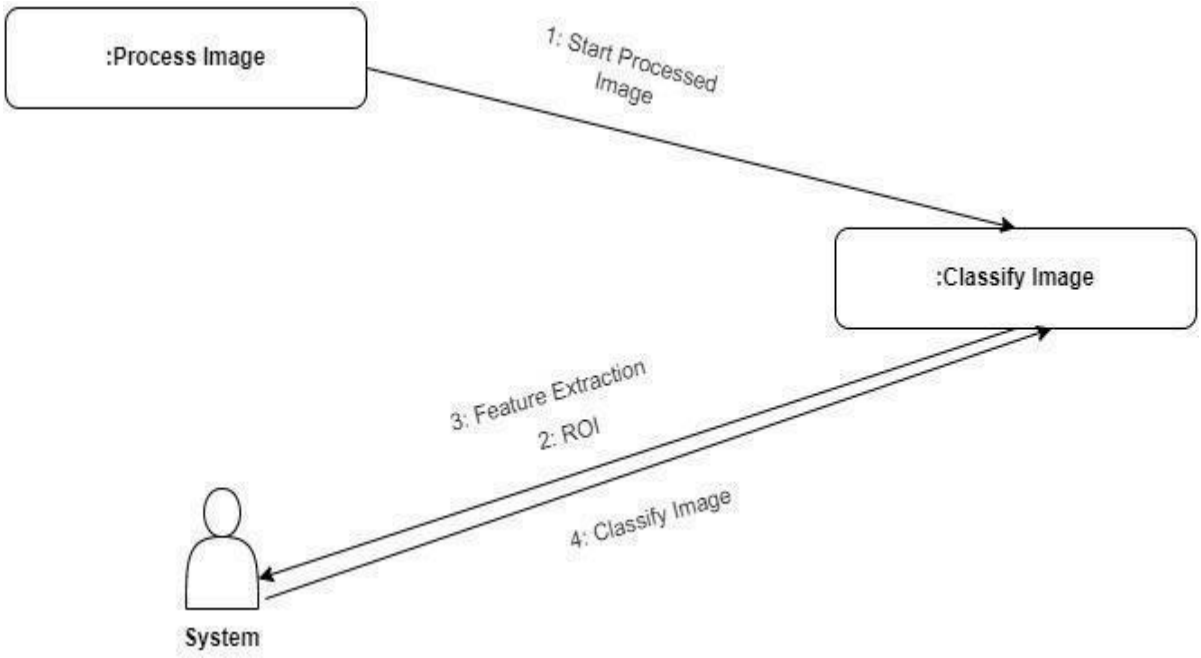
Capture Image



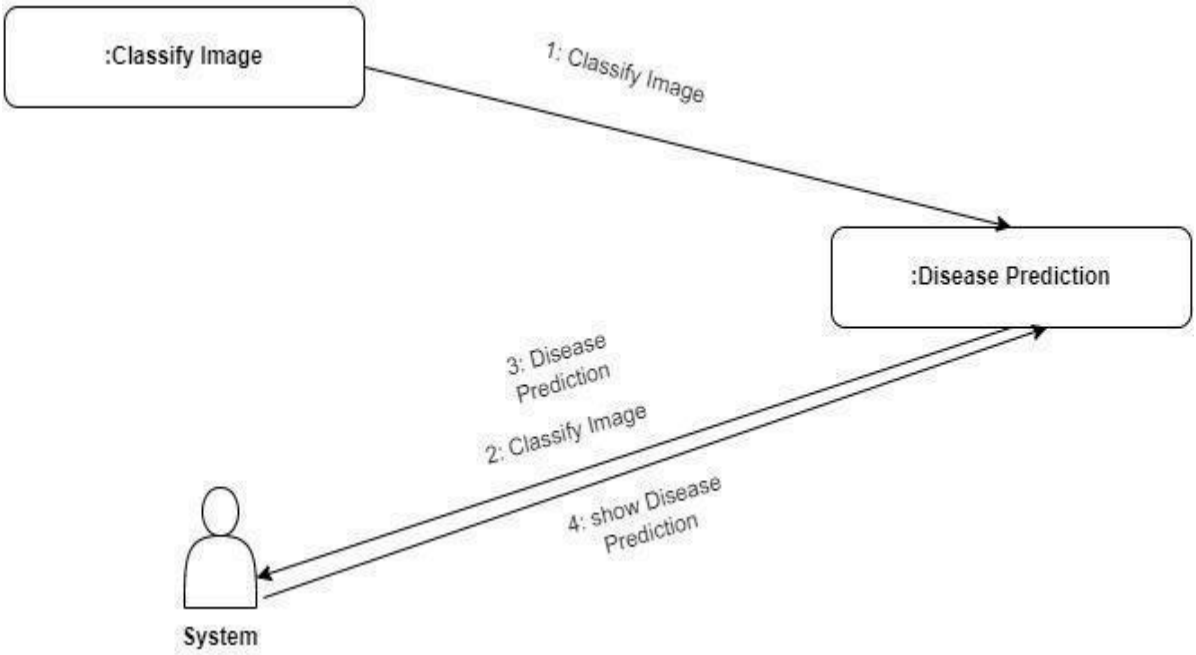
Process Image



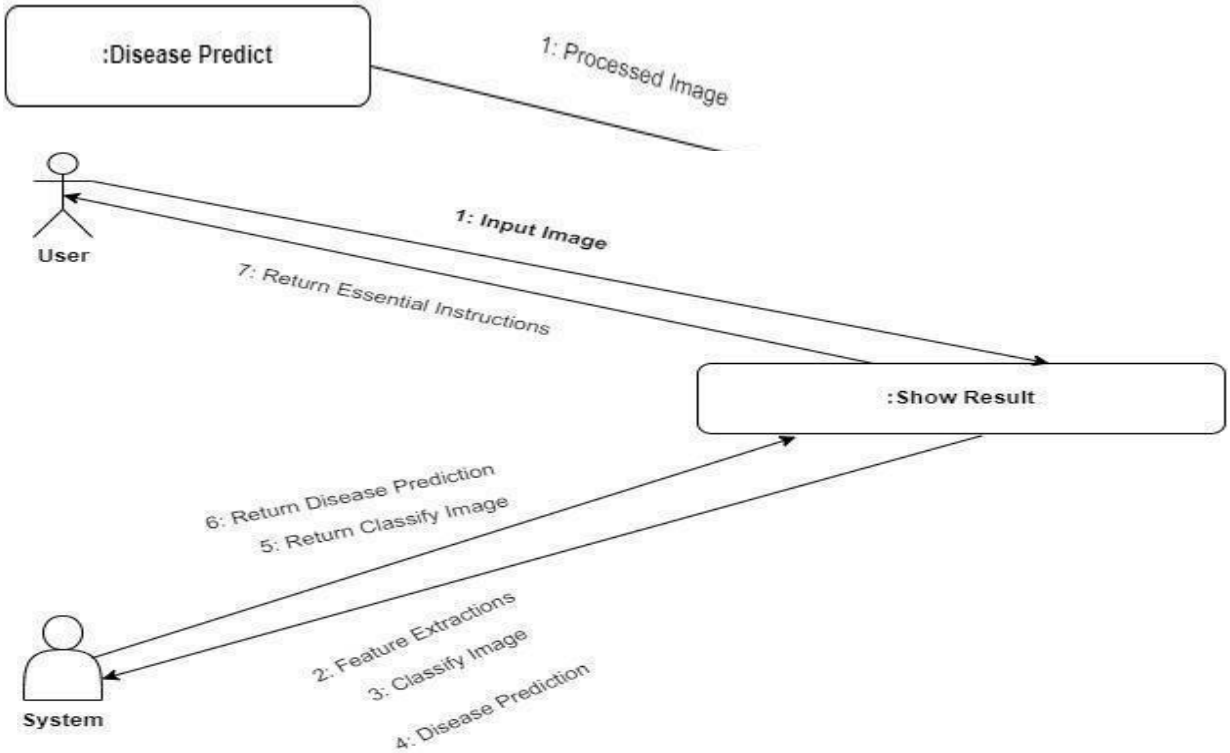
Classify Image



Disease Prediction:



Essential Instructions



4.5. Operation contracts

Contract co1: Input Image

Operation	System On (Input Image)
Responsibility	User Input image into the system
Cross reference	use case: UC_Input_Image
Pre-condition	User visit the system
Post condition	User input the required MRI image. System processed the image further.

Contractco2: Image Segmentation

Operation	Image Segmentation
Responsibility	System accepts the user image and do segment on it.
Cross reference	use case: UC_Image_Segmentation
Pre-condition	Image must be enhanced first.
Post condition	Distinguish the objects from images.

Contract-co3: Feature Extraction

Operation	Feature Extraction
Responsibility	System extract feature from the uploaded image
Cross reference	use case: UC_Feature_Extraction
Pre-condition	Image must be segmentedfirst.
Post condition	Distinguish the image object features.

Contract-co4: Predict Disease

Operation	Predict Disease
Responsibility	System predict disease based on feature extraction from the uploadedimages.
Cross reference	use case: UC_ Predict Disease
Pre-condition	Have required processed image.
Post condition	Predict Brain Tumor disease.

Contract-co5: Image Classification

Operation	Image Classification
Responsibility	To classify the input image of MRI in terms of localization
Cross reference	use case: UC_ Image Classification
Pre-condition	Have required processed image.
Post condition	Classify the objects and attributes.

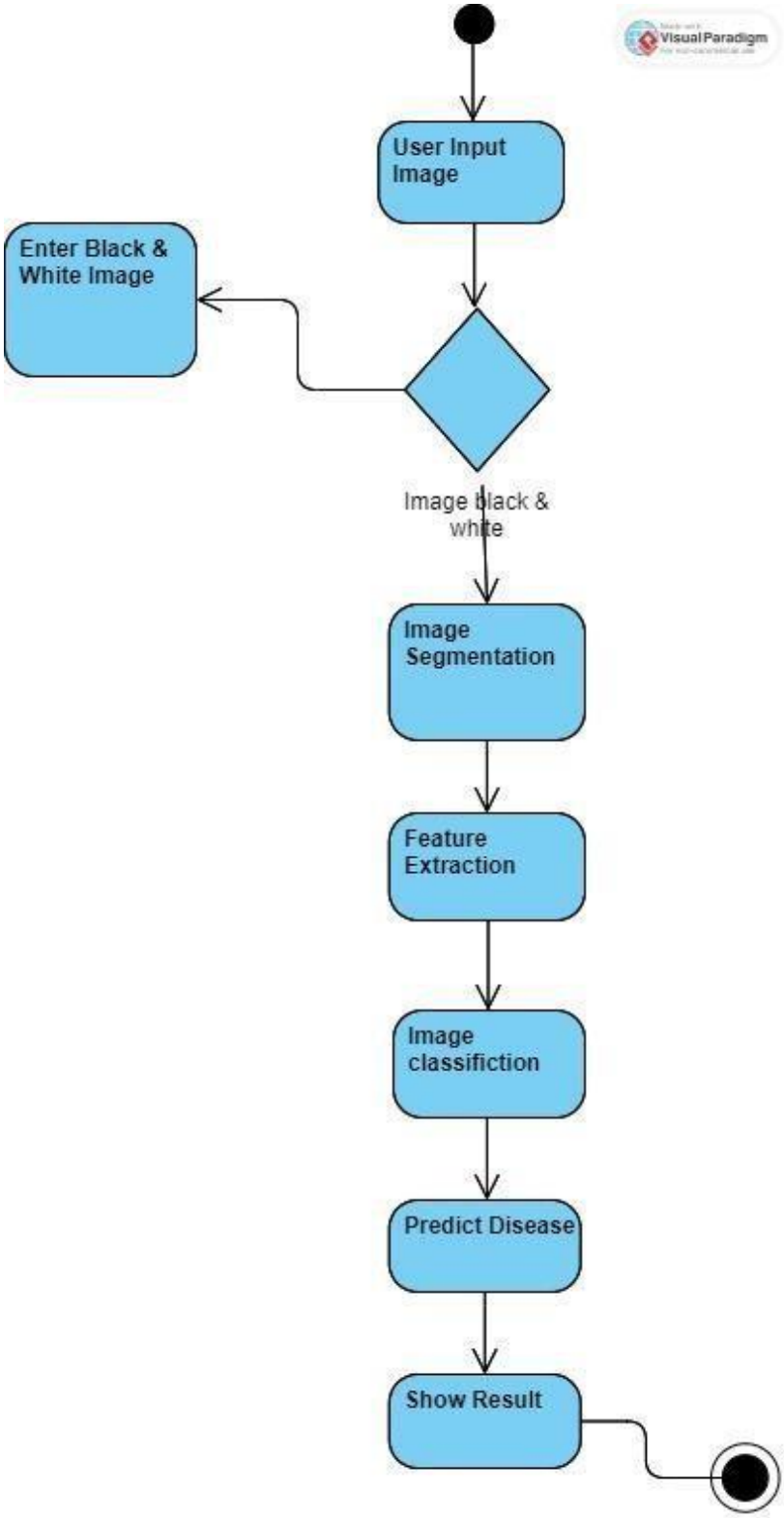
Contract-co6: Essential Instruction

Operation	Essential Instruction
Responsibility	System show essential instruction for user to prevent disease if diseaseprediction is true.
Cross reference	use case: UC_ Essential Instruction
Pre-condition	Disease Prediction True.
Post condition	Show result.

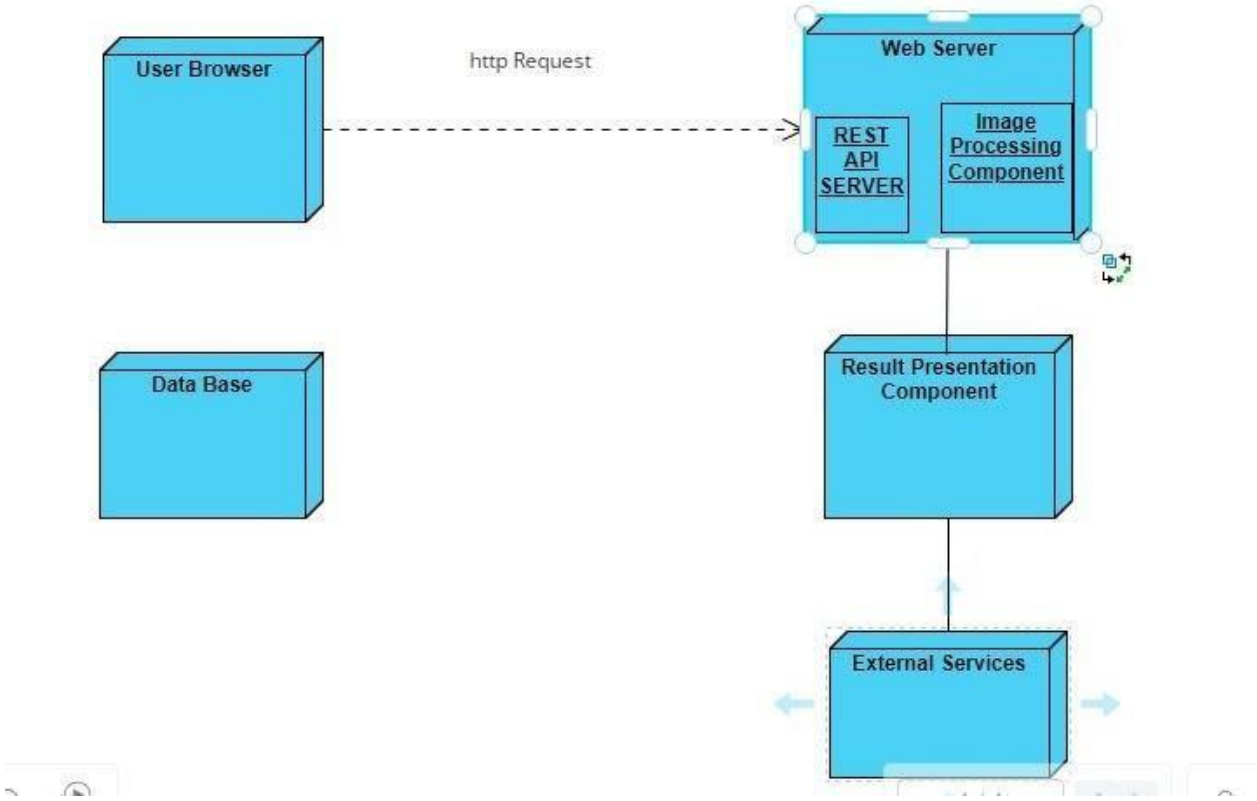
Contract-co6: Essential Instruction

Operation	Show Result
Responsibility	To provide result
Cross reference	use case: UC_ Show Result
Pre-condition	Have all required small results

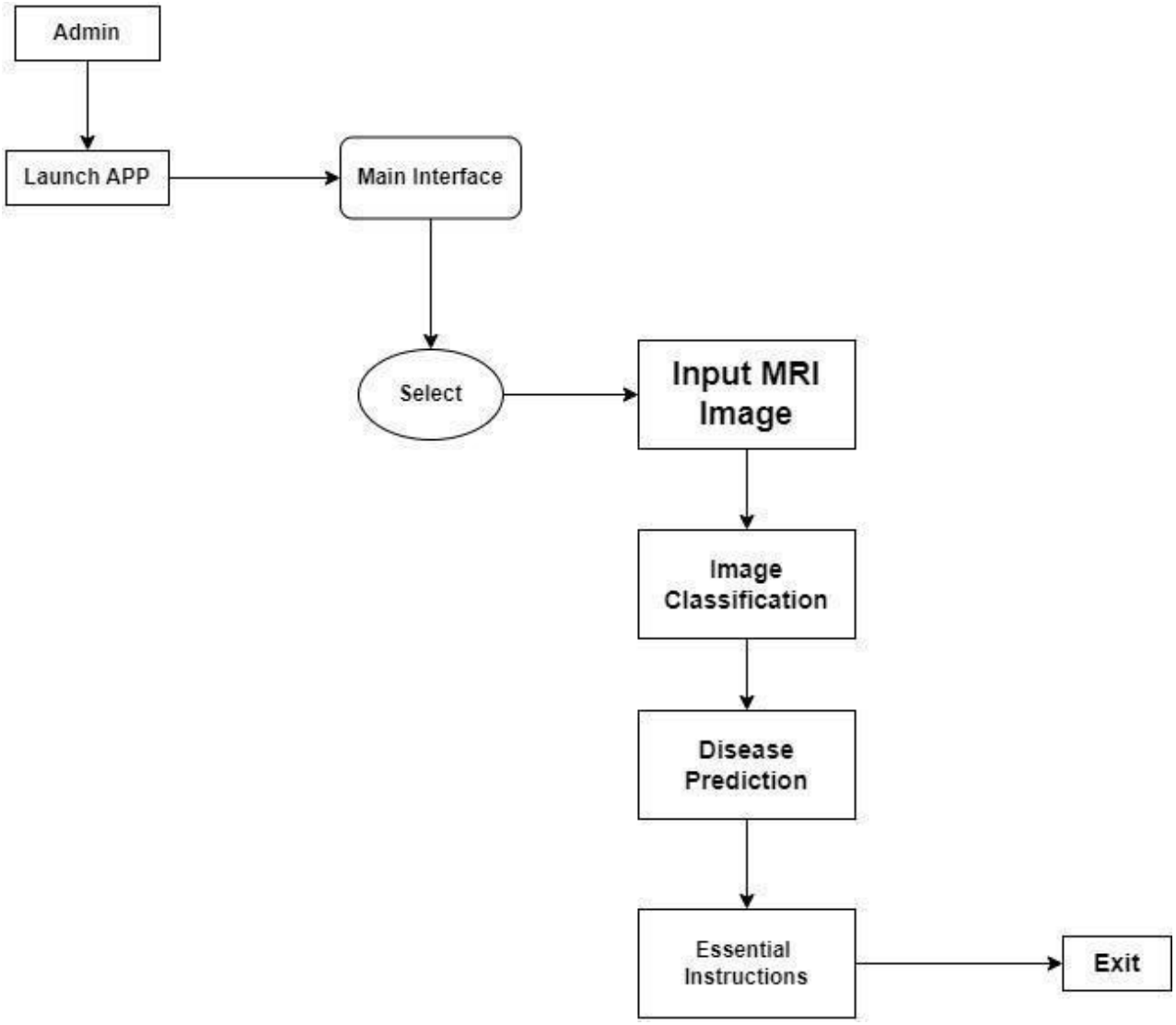
4.6. Activity Diagram



4.7. Deployment Diagram



4.8. Component Diagram



Chapter 5

Implementation

Chapter 5: Implementation

5.1. Components, Libraries, Web Services and stubs

NumPy : For linear operations and mathematical operations and handling array

Pandas: Handle data frames.

Seaborn: For visualization

Pickle: It's used for save the file and load the data file when it used.

Matplotlib: This is used to visualize image, graphs and losses matrices.

OS: Handling the operating system.

Stubs:

Image Segmentation Stub

- **Description:** A placeholder stub for image segmentation during early development stages.
- **Purpose:** Enables frontend and backend teams to work concurrently by providing simulated segmentation results.

Machine Learning Model Stub

- **Description:** A simplified model stub used in development and testing before the integration of the actual ResNet model and neural network.
- **Purpose:** Facilitates the iterative development process and allows for early testing of system functionalities.

Machine Learning Models:

ResNet Model

- **Description:** Integrated into the backend for hierarchical feature extraction.

- **Purpose:** Identifies intricate patterns within MRI images, contributing to the accuracy of tumor detection.

Neural Network

- **Description:** A custom neural network architecture is employed to further enhance feature extraction and refine the classification process.
- **Purpose:** Augments the ResNet model by capturing additional nuances within segmented MRI images.

5.2. Deployment Environment

The deployment environment for the Brain Tumor Detection Application is strategically designed to leverage the robust features of Amazon Web Services (AWS). The application's backend, powered by the Flask framework, will be deployed on Amazon EC2 instances using a web server to ensure optimal performance and resource utilization. The SQL database, a critical component for storing user data and system logs, will be hosted on an AWS RDS instance, benefiting from the managed database service's high availability and simplified administration. The fine-tuned ResNet model and neural network, integral to the machine learning aspects of the application, will be served using TensorFlow Serving. Static assets, including HTML, CSS, and JavaScript.

5.3. Tools and Techniques

TOOL	REASON
PyCharm Community	For python code and training data
Google Collaboratory	For model training
Anaconda prompt	For hosting API
Visio	For UML diagram
LibreOffice Draw	For UML diagram
Tensor flow, Keras	For feature extraction/ Build Neural Network
Flask	For API development
OpenCV	For Edge Detection
Scikit-learn	For training algorithm (Machine Learning)

5.4. Best Practices / Coding Standards

5.1.1 Backend Coding Standards

5.1.1.1 **Python**

- **Description:** Adhere to the Python Enhancement Proposal 8 for Python code styling and conventions.
- **Rationale:** Consistent code style enhances readability and maintainability, fostering collaboration among developers.

5.1.1.2 Flask Coding Standards

- **Description:** Follow Flask coding standards and best practices outlined in the official documentation.
- **Rationale:** Flask-specific conventions ensure consistency and compatibility with the Flask framework.

5.1.2 Machine Learning Model Coding Standards

5.1.2.1 TensorFlow Best Practices

- **Description:** Adhere to TensorFlow best practices for model development, including proper use of layers, variable scopes, and naming conventions.
- **Rationale:** Consistent practices ensure efficient and maintainable machine learning model code.

5.1.3 Frontend Coding Standards

5.1.3.1 JavaScript Standards

- **Description:** Use JavaScript 6 and above for JavaScript development, following modern JavaScript coding standards.
- **Rationale:** ES6+ features enhance code readability and introduce modern language constructs.

5.1.3.2 React.js Best Practices

- **Description:** Adhere to React.js best practices, including

component-based development, proper state management, and effective use of hooks.

- **Rationale:** React.js best practices ensure efficient and scalable frontend development.

5.1.4 Version Control

5.1.4.1 Git Workflow

- **Description:** Follow a Git workflow, such as Gitflow, to manage branches, releases, and feature development.
- **Rationale:** A structured Git workflow enhances collaboration, versioning, and release management.

5.1.5 Testing Standards

5.1.5.1 Unit Testing

- **Description:** Implement unit tests for backend logic, machine learning models, and frontend components.
- **Rationale:** Unit testing ensures the reliability and correctness of individual components.

5.1.5.2 Integration Testing

- **Description:** Conduct integration tests to verify the seamless interaction between different system components.
- **Rationale:** Integration testing identifies and resolves issues in the overall system integration.

5.1.6 Security Best Practices

5.1.6.1 OWASP Guidelines

- **Description:** Follow Open Web Application Security Project (OWASP) guidelines for secure coding practices, including input validation, authentication, and protection against common vulnerabilities.
- **Rationale:** Adhering to security best practices reduces the risk of security vulnerabilities.

5.1.7 Documentation

5.1.7.1 Inline Comments and Docstrings

- **Description:** Include clear and concise inline comments and docstrings in the code for enhanced documentation.
- **Rationale:** Well-documented code facilitates understanding, maintenance, and collaboration among developers.

5.1.7.2 Readme.md

5.5. Version Control

Version control is a critical aspect of Brain Tumor Detection development lifecycle, providing a systematic approach to managing code changes, collaborating among developers, and ensuring the stability of the codebase. Brain Tumor Detection uses Git as its version control system, employing best practices to streamline development workflows and facilitate collaboration.

Key Version Control Practices:

1. Git Repository:

- The Brain Tumor Detection code base is hosted in a central Git repository, providing a single source of truth for the project.

2. Branching Strategy:

- A branching strategy is in place to manage different aspects of development.
- **main Branch:** Represents the production-ready code.
- **development Branch:** Integrates feature branches for ongoing development.
- **Feature Branches:** Created for each new feature or enhancement, ensuring isolation and easy integration.

3. Commit Messages:

- Descriptive and meaningful commit messages are used to explain the purpose of each commit clearly.
- Follows the conventional commit format for consistency.

4. Code Review:

- Code reviews are an integral part of the version control process.
- Peers review each other's code to catch bugs, ensure adherence to coding standards, and share knowledge.

Chapter 6

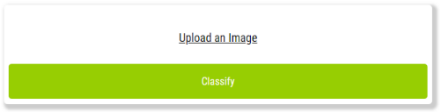
Testing and Evaluation

Chapter 6: Testing and Evaluation

The "Testing and Evaluation" chapter documents the processes used to verify the system's functionality and performance. This section describes the testing strategies applied, such as unit, integration, and system testing, to ensure all components work seamlessly together. Evaluation metrics and criteria are defined to measure the system's effectiveness, reliability, and user satisfaction. The chapter also includes an analysis of the test results, highlighting any detected issues and the steps taken to address them, ultimately demonstrating the project's adherence to its objectives and quality standards.

6.1. Use Case Testing

- **Input image**

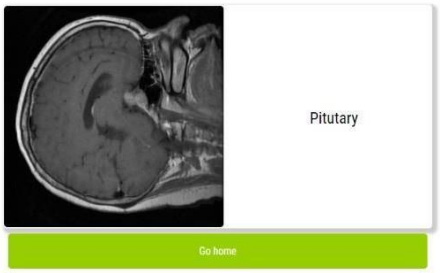


- **MRI Image Classification**

  Classify image(): It will classify the image from 3 categories.

Classification of Meningioma

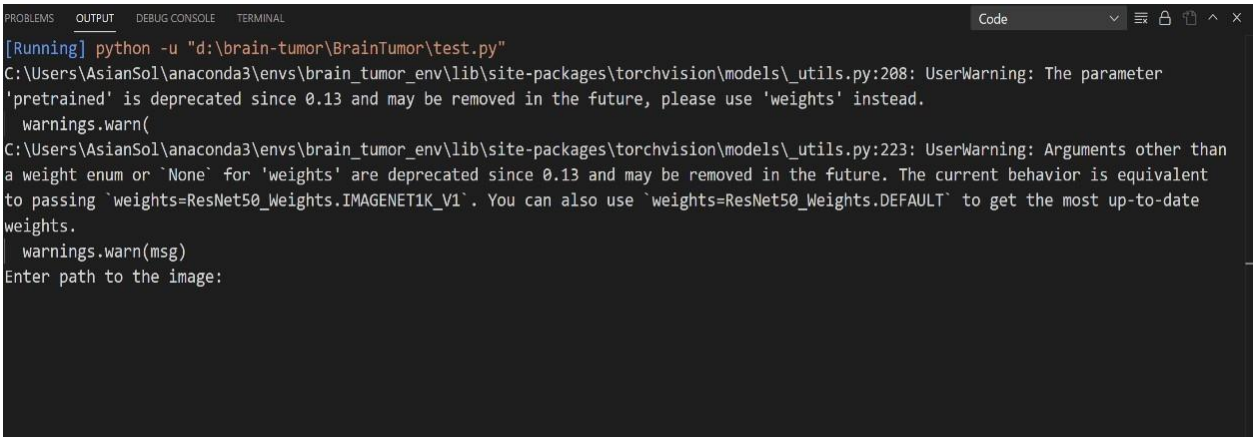
Classification of Glioma



Classification of Pitutary

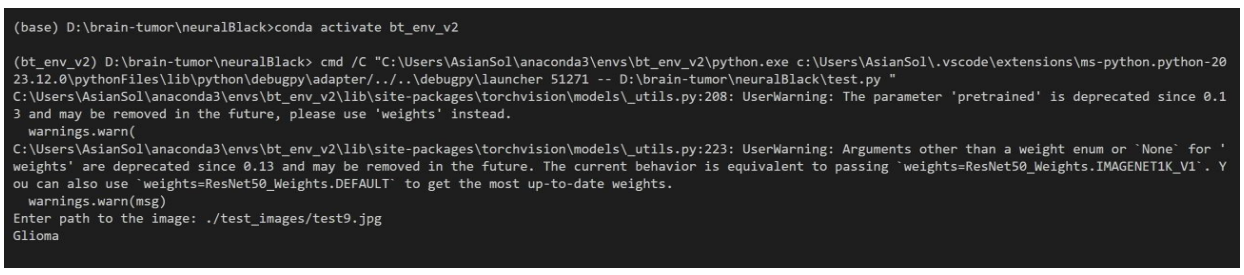
- **MRI Disease Prediction**

  Disease prediction(). System will the disease on a particular image.

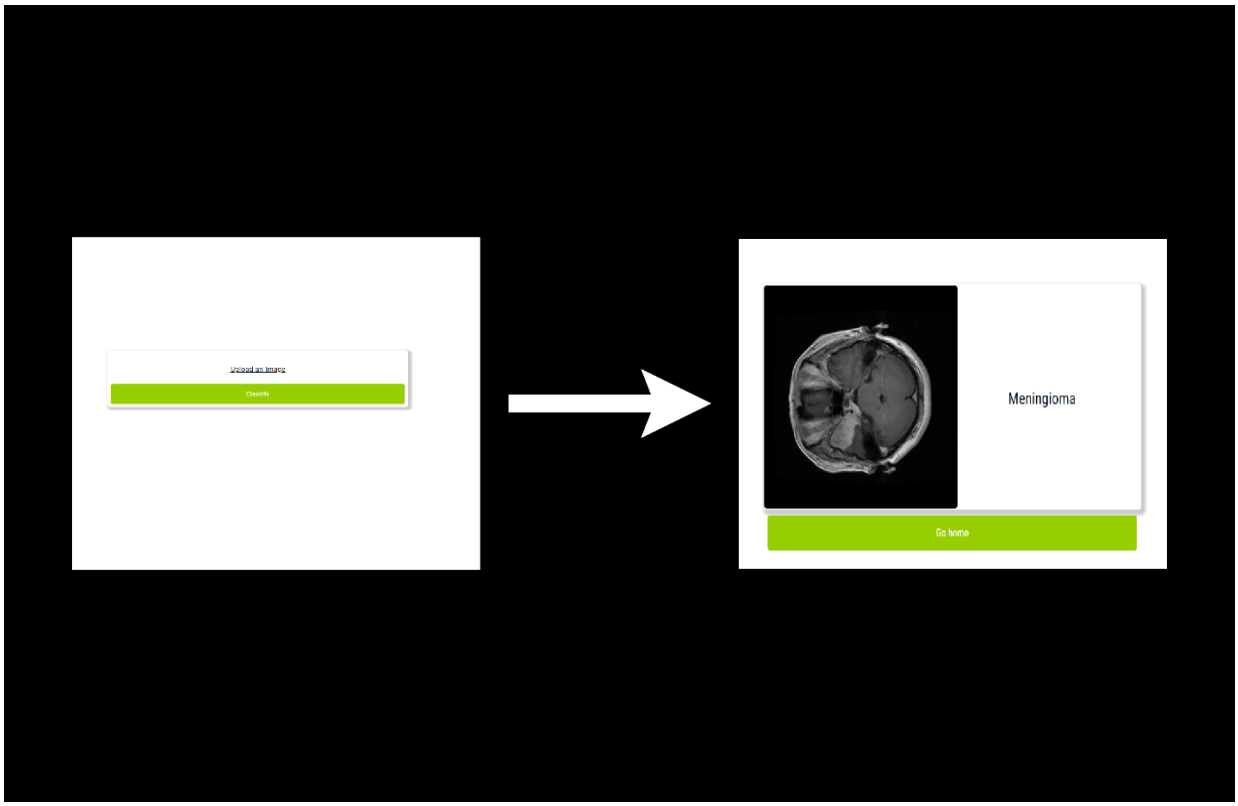


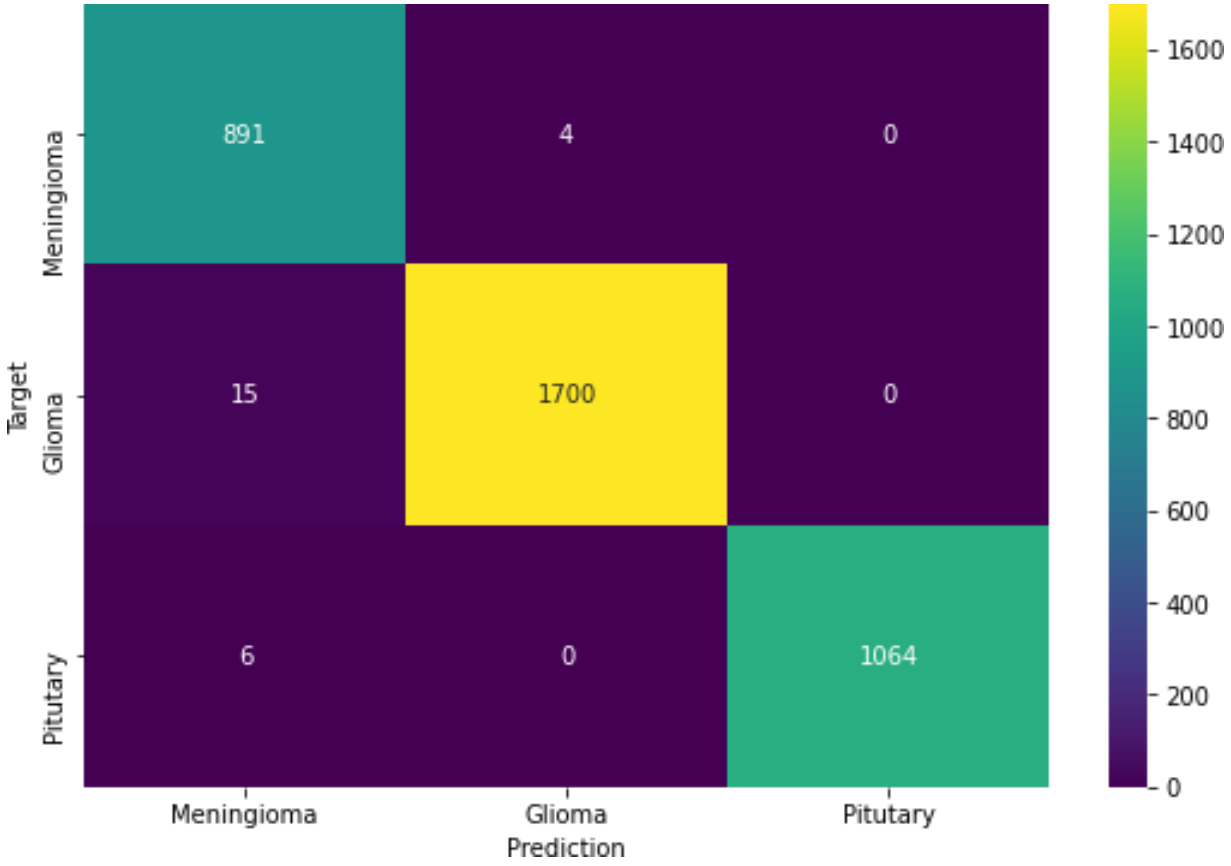
Enter the Image Path

- **Disease Prediction**



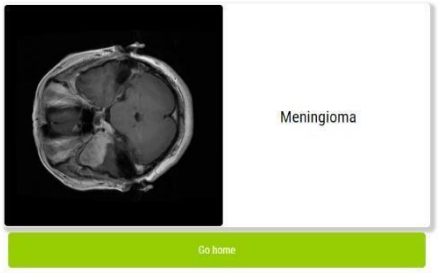
6.2. Data Flow Testing



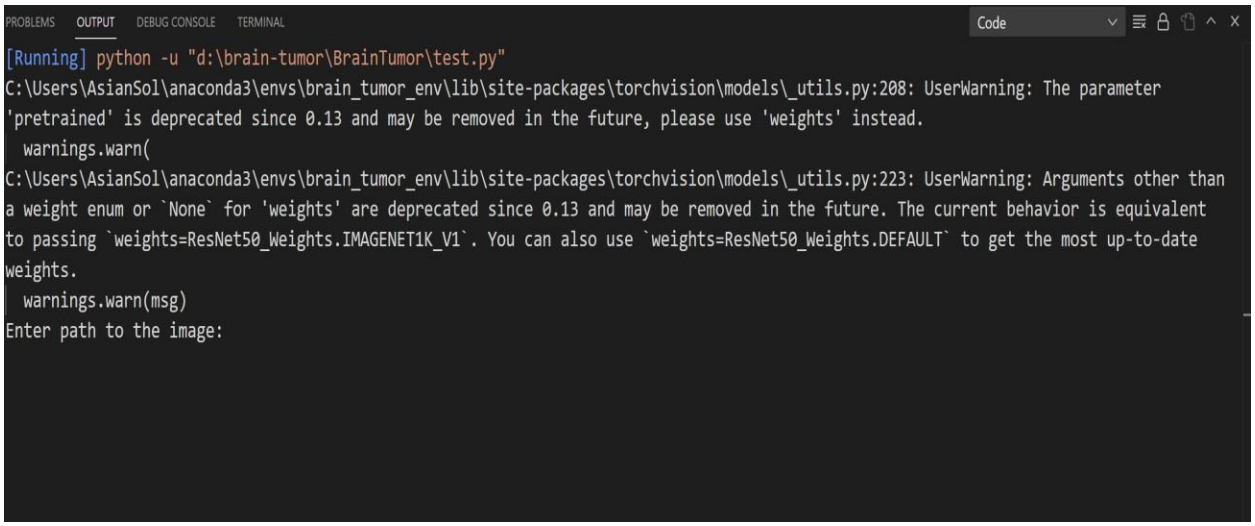


6.3. Unit Testing

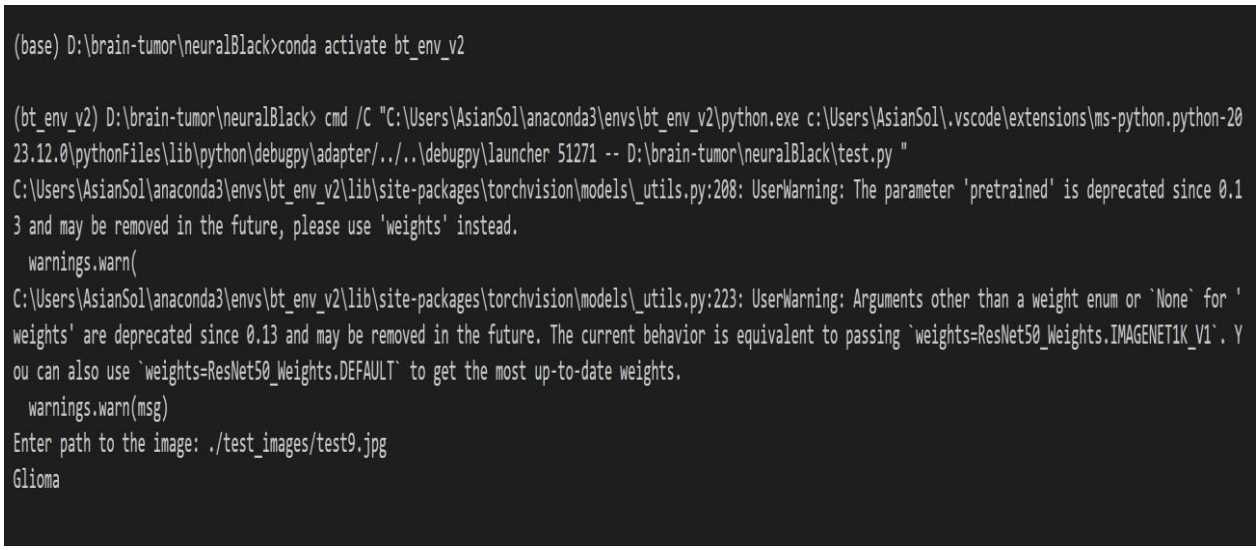
Screen # 03 (Classify Image)



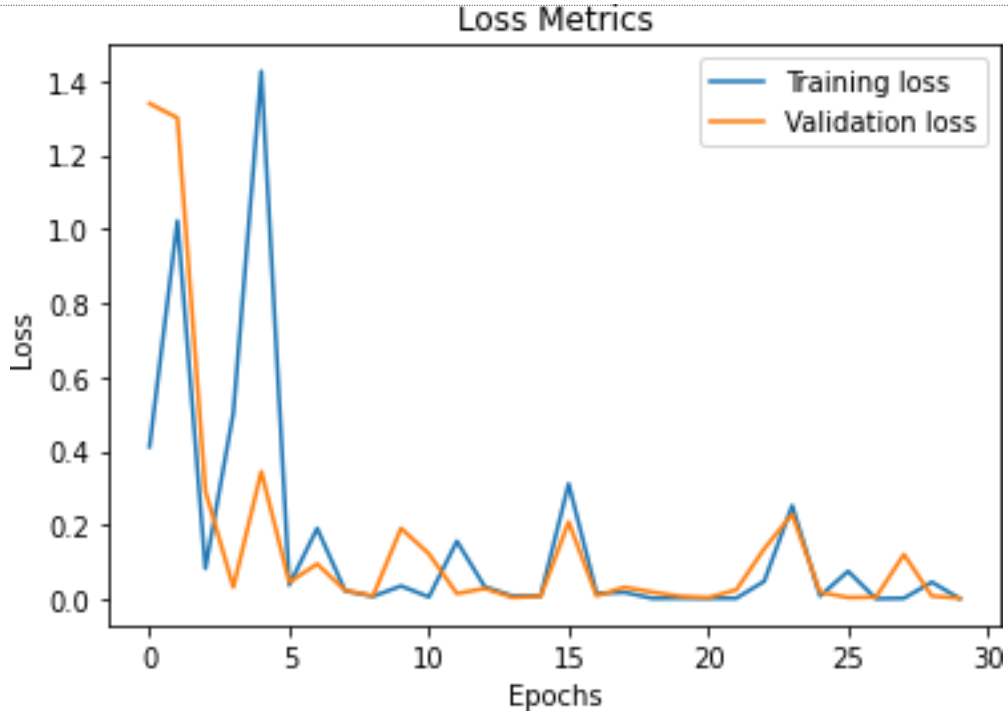
Screen # 03 (Run in Terminal)



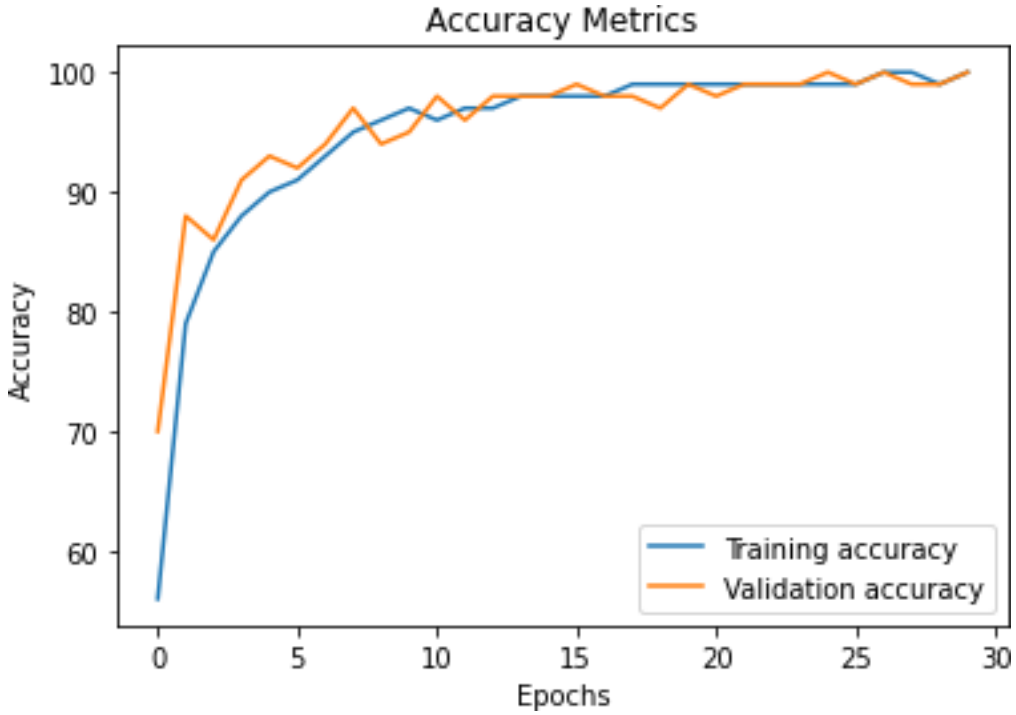
Screen # 04 (Disease Prediction)



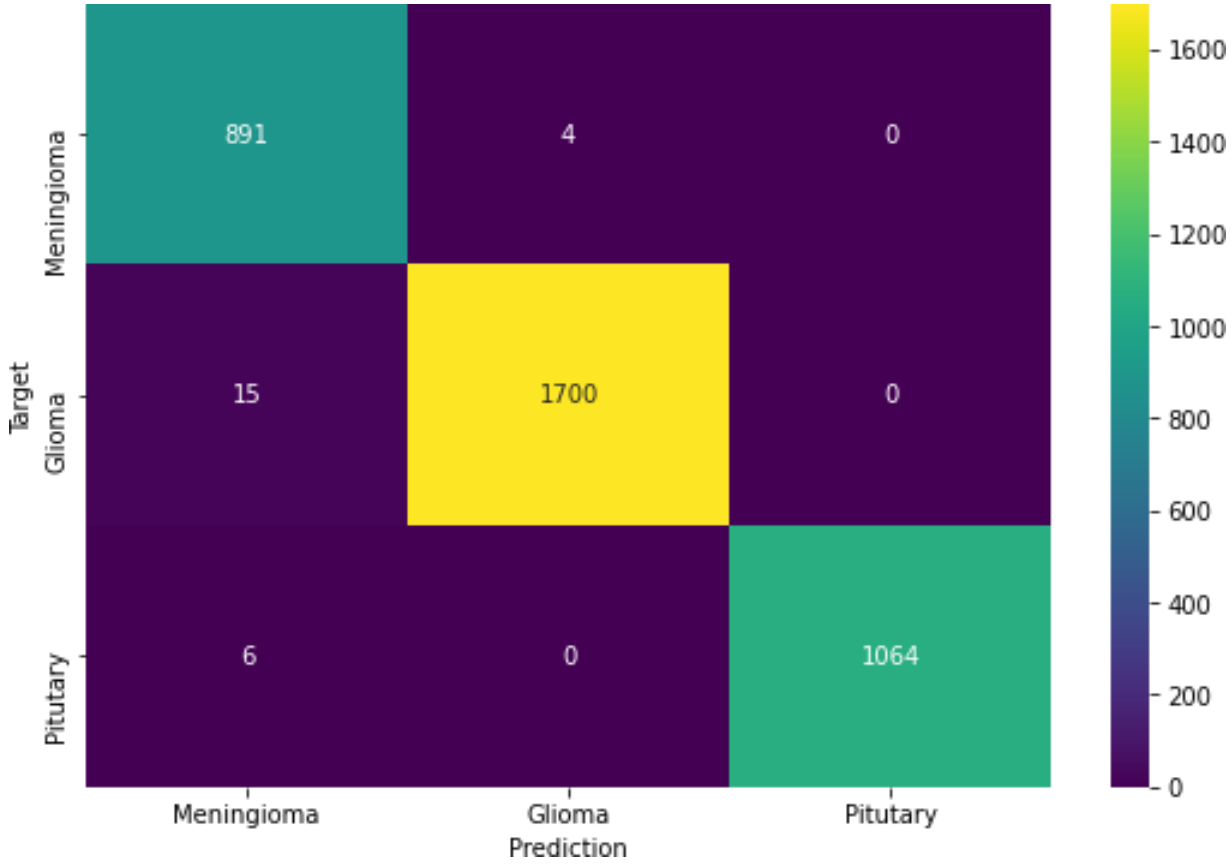
Screen # 05 (Loss Matrics)



Screen # 06 (Accuracy Metrics)

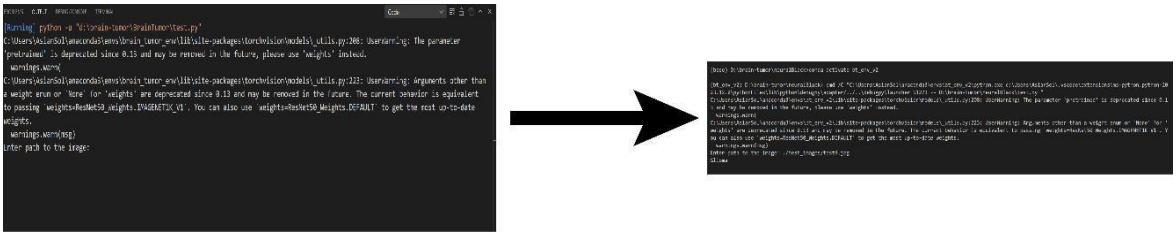


Screen # 07 (Confusion Matrics)

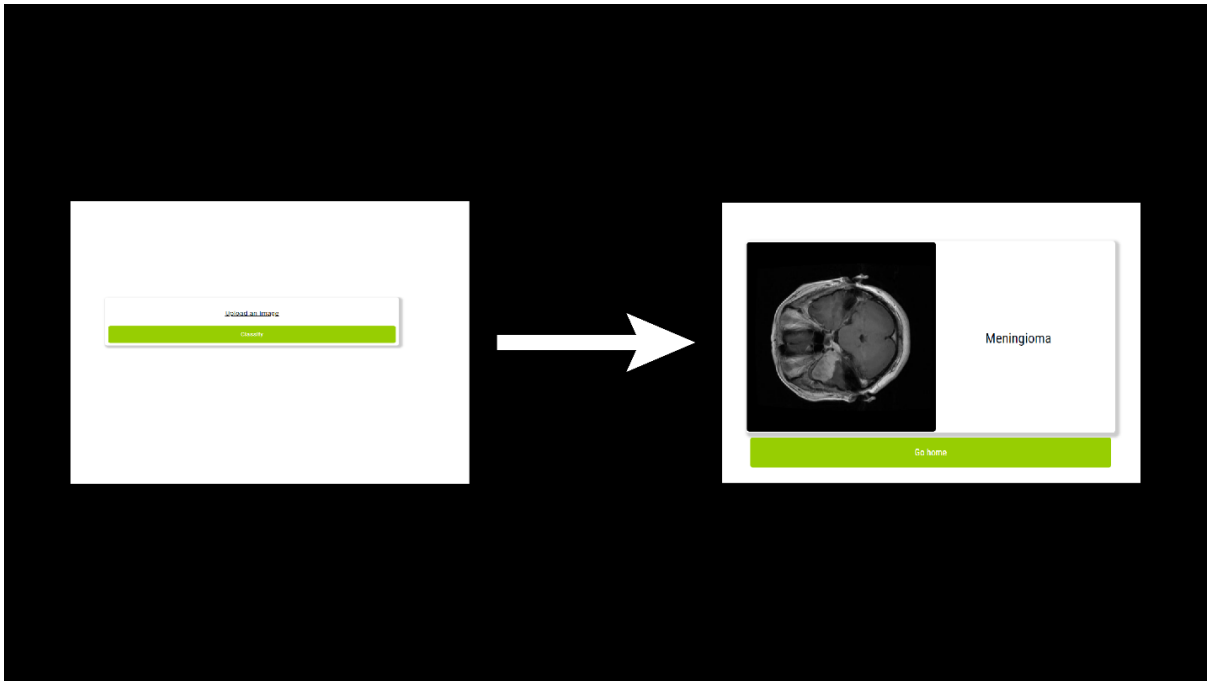


6.4. Integration testing

Machine Testing:



User System Testing:



Chapter 7

Summary, Conclusion and Future Enhancements

Chapter 7: Summary, Conclusion & Future Enhancements

7.1. Project Summary

Medical image classification has received considerable attention in recent years, and convolutional neural networks (CNN) are the most popular neural network models for image classification problems. CNNs aim to adaptively determine features through backpropagation by applying many building blocks such as convolutional layers, pooling layers, and fully connected layers. In this project, we focus on developing a CNN model to classify brain tumors in T1-weighted contrast-enhanced MRI images. The proposed system consists of two important steps. First, the images are preprocessed using different image processing techniques, and then a CNN is used to classify the preprocessed images. The experiment was conducted on a dataset of 3865 images. Using our CNN model, we achieved a high-test accuracy of 94.39%, an average precision of 93.33%, and an average recall of 93%. The proposed system shows satisfactory accuracy on the dataset and outperforms many existing major methods. These algorithms are trained using Python Machine Learning (CNN) on the backend.

7.2. Achievements and Improvements

We created a user-friendly interface where users can input MRI images and display tumor detection results after classification. You can achieve high accuracy when using the resnet-50 model to detect brain tumors. Explore integration opportunities with healthcare systems as well as electronic healthcare systems and image archiving for seamless data exchange. Explore opportunities to use advanced fine-tuning hyperparameters to improve detection model performance.

7.3. Lessons Learnt

Understand the field of medical imaging and gain detailed knowledge about brain tumor detection. The program teaches how to work in and with teams. We also gained experience using Python, which has been very useful in my career. We deployed the machine learning model in a web environment using the Flask framework. We process the images and enhance the data to improve the model's generalization ability.

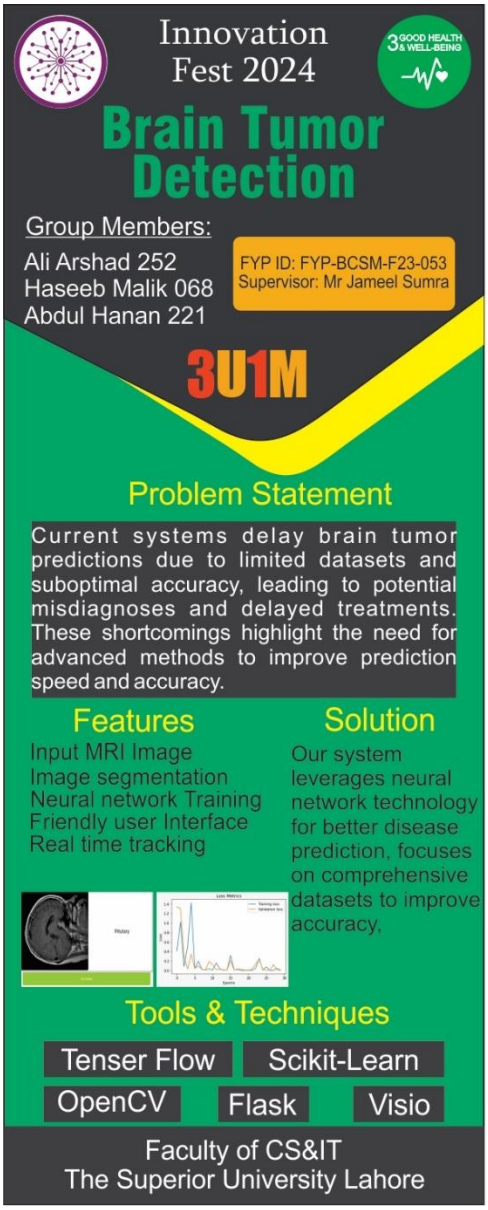
7.4. Future Enhancements/Recommendations

In Future we have explore advanced deep learning architecture and techniques to improve the accuracy and robustness of system. This is also help in detection on large dataset and also increase the performance of the system. Also integrate with health care system in hospitals which help to check the patient's brain tumor reports. We have also train over model on CT scans and other Images so it detects the Brain Tumor and its Types. Also give Information to patients and health care people and educate to how they can use AI tools and its benefits in its field.

Appendices

Appendix A: Information / Promotional Material

C.1. Standee



Reference and Bibliography

Reference and Bibliography

1. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC10453020/
2. https://www.cancer.net/cancer-types/brain-tumor/diagnosis
3. https://www.hindawi.com/journals/jhe/2022/2693621/
4. https://www.mdpi.com/2075-1729/13/7/1449