[Tumor Vision]

By:

Syed Sibt E Hassan Kazmi
38252
Waleed Bin Khalid
35777

Supervised by:
[Ishtiaq Ali]

Faculty of Computing
Riphah International University, Islamabad
Spring/Fall 20xx

A Dissertation Submitted To

Faculty of Computing,

Riphah International University, Islamabad

As a Partial Fulfillment of the Requirement for the Award of the Degree of

Bachelor of Science in Computer Science

Faculty of Computing
Riphah International University, Islamabad

Date: [date of final presentation]

Final Approval

This is to certify that we have read the report submitted by **Syed Sibt E Hassan Kazmi (38252), Khalid Bin Waleed (35777),** for the partial fulfillment of the requirements for the degree of the Bachelors of Science in Computer Science (BSSE). It is our judgment that this report is of sufficient standard to warrant its acceptance by Riphah International University, Islamabad for the degree of Bachelors of Science in Computer Science (BSSE).

Committee:

1 ________________________________

[Name Supervisor]
(Supervisor)

2 ________________________________

[Name of HOD/chairman]
(Head of Department/chairman)

Declaration

We hereby declare that this document "**[Tumor Vision]**" neither as a whole nor as a part has been copied out from any source. It is further declared that we have done this project with the accompanied report entirely on the basis of our personal efforts, under the proficient guidance of our teachers, especially our supervisor **[Ishtiaq Ali]**. If any part of the system is proved to be copied out from any source or found to be a reproduction of any project from anywhere else, we shall stand by the consequences.

[Syed Sibt E Hassan Kazmi]

[38252]

[Waleed Bin Khalid]

[35777]

Dedication

Insert dedication here…

Acknowledgement

First of all we are obliged to Allah Almighty the Merciful, the Beneficent and the source of all Knowledge, for granting us the courage and knowledge to complete this Project.

[Students will acknowledge here anyone who has helped in the project. It can include Supervisor(s), Teachers, Classmates, Friends and Family]

[Syed Sibt E Hassan Kazmi]

[38252]

[Khalid Bin Waleed]

[35777]

Abstract

TumorVision is an AI-powered web platform that assists doctors in detecting brain tumors from MRI scans using advanced deep learning technology. The system combines Densenet and Resnet models to automatically segment tumors with 99% accuracy, providing instant results through an intuitive interface. Developed with React.js and Python Flask, it supports DICOM files, enables secure data sharing, and integrates with clinical workflows while maintaining HIPAA compliance. While currently focused on common tumor types, future improvements will expand to rare cases through enhanced datasets. TumorVision demonstrates how AI can accelerate and improve neuro-oncology diagnostics, reducing diagnosis time while ensuring accuracy and data security. The platform's ongoing development includes mobile accessibility and regulatory compliance to broaden its clinical adoption and impact.

Table of Contents

List of Tables

List of Figures

Chapter 1
Introduction

Chapter 1: Introduction

Brain tumors are one of the most serious neurodevelopmental disorders. Approximately 320,000 new cases occur globally each year, according to the World Health Organization (WHO). Brain tumor accurate detection on time is important for effective treatment, yet traditional diagnostic method are time consuming and completely rely on radiologist expertise. Deep learning breakthroughs in recent years suggest an opportunity to revolutionize this process, enabling predictive tumor detection and segmentation as well as improved collaboration between healthcare workers.

Aim of this project to develop the secure web-based AI platform for doctor to diagnose brain tumor more accurately. Image processing technique will be integrated to web. Doctor will be register and secure system will be built for report sharing and doctor patient communication, diagnostic capabilities should only be accessible to authorized medical expert. Using AI-based automation and an intuitive interface, this platform will help to reduce diagnostic delays, minimize human error and enhance interdisciplinary collaboration in neuro-oncology workflows.

1.1 Goals and Objectives

Goals

- Develop web-based AI platform to automate the brain tumor detection through MRI.
- Built the secure and collaborative platform that will help in improving diagnostic workflow.
- To improve the accuracy and efficiency of brain tumor diagnosis compared to traditional manual methods.

Objective

- Brain tumor prediction using deep learning models trained on diverse MRI dataset.
- Propose a role based access model where only verified doctors can use diagnostic features
- Enable secure report generation and sharing to facilitate consultations among doctors.
- Integrate a communication module for doctor-patient interactions while ensuring confidentiality.

- Optimize system performance for fast and reliable processing of medical imaging data.

1.2 Scope of the Project

Included Features:

- **AI-Powered Tumor Prediction:** Automated detection of brain tumors in MRI scans.
- **Doctor Registration & Authentication:** Secure sign-up with medical license verification.
- **Diagnostic Report Generation:** AI-analyzed reports with tumor boundaries and key metrics.
- **Secure Data Sharing:** Encrypted sharing of reports between medical professionals.
- **Communication Portal:** In-app messaging for doctor-patient discussions.

Chapter 2
Literature Review

Chapter 2: Literature Review

2.1 Introduction

Review of the main work about the AI-based brain tumor detection systems and related technologies. We analyze works on previous medical image segmentation methods, doctor-patient collaboration tools and secure healthcare infrastructure. The review will help us to identify the best approaches, open some problems in current solutions and establish our project position in the field. As well as through analyzing technical implementations and clinical use cases, we build a basis for development of our web-based artificial intelligence diagnostic tool.

2.2 Background and Problem Elaboration

Brain tumor diagnosis traditionally depends on manual MRI scan analysis by radiologists - a slow process requiring specialized expertise. While AI-assisted systems have emerged, most focus only on tumor detection without addressing clinical workflow needs. Three key challenges persist: many AI tools lack integration with hospital systems, few solutions support secure doctor collaboration, and existing platforms often neglect patient communication features. Our project addresses these gaps by combining accurate tumor prediction with practical healthcare collaboration tools in one secure platform.

2.3 Detailed Literature Review

Its aims are to introduce the essential concepts and current research about AI-based medical image analysis (including brain tumor detection and segmentation) and then explore different approaches, their advantages and disadvantages for our system.

Definitions

Related Research Work 1

(Tummala, 2023) [1] proposes a deep learning approach using an Inception ResNet CNN architecture for automated classification of brain tumors from MRI scans into four classes: glioma, meningioma, pituitary tumors, and no tumor. The model leverages a 164-layer pretrained CNN (trained on ImageNet) with convolutional, pooling, and fully connected layers to extract hierarchical features from MRI images. To mitigate overfitting, the study employed data augmentation (random shearing, zooming, flipping) and an 80:20 train-test split of a curated Kaggle dataset containing 5,952 MRI images across tumor types. The model achieved 96.7% accuracy, with high sensitivity for glioma (97.5%), pituitary (99.4%), and no-tumor cases (98.0%), but lower performance for rare meningiomas (71.9%). Results were validated through confusion matrices and loss/accuracy graphs, showing minimal false negatives (5% missed tumors). The author highlights the model's potential for clinical deployment but notes limitations in meningioma detection due to dataset imbalance. Future work could explore larger datasets and alternative architectures to improve robustness.

Related Research Work 2

(Aamir et al., 2024) [2] proposes an optimized convolutional neural network (CNN) model for brain tumor detection and classification from MRI scans. The study introduces a hyperparameter-tuned CNN architecture that systematically optimizes critical parameters including batch size, layer counts, learning rate, activation functions (ReLU), pooling strategies, and filter dimensions (3×3, 5×5). The model was trained on three Kaggle datasets (totaling 7023 images across glioma, meningioma, pituitary, and no-tumor classes) using rigorous preprocessing with z-score normalization and non-local means denoising. Through iterative fine-tuning with SGD/Adam optimizers and the model achieved exceptional performance metrics: 97.18% accuracy for multi-class classification (Dataset 1), with consistent precision, recall, and F1-scores averaging 97% across all datasets. Comparative analysis demonstrated superiority over existing methods like Inception-V3 (94% accuracy) and MobileNetV2 (92% accuracy), attributed to its optimized architecture featuring 16-128 filters and 20% dropout for regularization. The study highlights the model's clinical utility in reducing diagnostic variability, though notes limitations in generalizability to underrepresented tumor types and computational resource requirements. Future work aims to address these constraints through expanded datasets and multimodal image integration.

Related Research Work 3

(Sen et al., 2022) [3] proposes a deep learning approach using transfer learning with state-of-the-art CNN architectures (EfficientNetB0, ResNet50, Xception, MobileNetV2, and VGG16) for brain tumor classification from MRI scans. The study leverages a dataset of 3,264 images (glioma, meningioma, pituitary, and no-tumor classes) enhanced with advanced preprocessing techniques including N4 bias field correction, BM3D denoising, skull stripping, and data augmentation (rotation, flipping) to address dataset limitations. Among the tested models, EfficientNetB0 achieved the highest accuracy of 97.61%, outperforming other architectures like ResNet50 (96.60%) and VGG16 (lower performance due to lack of skip connections). Key innovations include the use of Global Average Pooling to reduce parameters, and Adam optimizer with categorical cross-entropy loss for multi-class classification. The model demonstrates strong clinical potential with 100% precision for pituitary and no-tumor cases, though limitations include sensitivity to small dataset size and variability in MRI views (sagittal, coronal, axial). Future work could explore larger datasets and hybrid architectures to improve robustness.

Related Research Work 4

(Kumar & Kumar, 2023) [4] propose a 25-layer Convolutional Neural Network (CNN) for brain tumor classification and segmentation using T1-weighted MRI images. The study leverages two public datasets: one with 3,064 images (meningioma, glioma, pituitary tumors) and another with 516 images (glioma grades II–IV). The CNN architecture includes convolutional, ReLU, normalization, max-pooling, dropout, and fully connected layers, optimized using Adam and SGDm. To address data scarcity, the authors applied augmentation techniques (flipping, rotation, noise injection), expanding the dataset fivefold. The model achieved 86.2% accuracy for tumor type classification (Study I) and 81.6% for glioma grading (Study II) using Adam optimizer, outperforming traditional methods like SVM (83.3%) and AlexNet (84.2%). Key strengths include automated feature extraction without manual segmentation and improved robustness through augmentation. However, limitations include moderate accuracy for rare tumor subtypes (e.g., 69.9% specificity for Grade II gliomas) and computational intensity (89 minutes training time). The authors suggest future enhancements with larger datasets and transfer learning. Clinically, this work aids early diagnosis but requires validation across diverse populations to ensure generalizability.

Related Research Work 5

(Nawaz et al., 2022) [5] propose a Hybrid Brain Tumor Classification (HBTC) framework combining texture features (Co-occurrence Matrix, Run-Length Matrix, Gradient) and machine learning classifiers to classify four brain tumor types—cyst, glioma, meningioma (menin), and metastatic (meta)—from T2-weighted MRI scans. The study employs a novel hybrid optimization technique (Fisher + Mutual Information + Probability of Error + Correlation-based Feature Selection) to reduce 490,000 extracted features to 9 dominant features, achieving 98.3% accuracy with Multilayer Perceptron (MLP) on a dataset of 1,000 images (250 per class) from Bahawal Victoria Hospital. Preprocessing uses a hybrid K-S-L (Kernel-Sobel-Low-pass) filter for noise reduction, while segmentation leverages Threshold and Clustering-based Segmentation (TACS) to isolate tumor regions. The framework outperforms SVM (97%), DNN (96.97%), and CNN (97.1%) in comparative studies, with MLP excelling due to its nonlinear activation handling noisy medical data. Key strengths include computational efficiency and high sensitivity for rare tumors like cysts (98.4%). Limitations include reliance on single-modality (T2-MRI) data and manual ROI sizing (optimal performance at 20×20 pixels). Future work could integrate multimodal imaging and deep learning for broader clinical applicability.

2.4 Literature Review Summary Table

Table 2.1 Literature Review Summary Table

Paper	**Size of Dataset**	**Classifier Used**	**Classification Type**	**Accuracy**
Tummala (2023) [1]	5,952 images (Kaggle)	Inception ResNet CNN (164-layer, pretrained on ImageNet)	Multi-class (Glioma, Meningioma, Pituitary, No Tumor)	96.7%
Aamir et al. (2024) [2]	7023 images	Optimized CNN (hyperparameter-tuned)	Multi-class (Glioma, Meningioma, Pituitary, No Tumor)	97.18%
Sen et al. (2022) [3]	3,264 images	Transfer Learning (EfficientNetB0, ResNet50, Xception, MobileNetV2, VGG16)	Multi-class (Glioma, Meningioma, Pituitary, No Tumor)	97%
Kumar & Kumar (2023) [4]	3,580 images (3,064 for classification, 516 for glioma grading)	25-layer CNN	Multi-class classification and glioma grading	86.2%

Nawaz et al. (2022) [5]	1,000 images (250/class)	MLP (with hybrid feature selection)	Multi-class (Cyst, Glioma, Meningioma, Metastatic)	98.3%

2.5 Research Gap

The existing research on AI-based brain tumor detection and classification has several limitations and gaps that need to be addressed:

- Small or biased datasets – Many studies use limited MRI scans, missing rare tumor types.
- Poor performance on rare tumors – Models struggle with uncommon cancers due to few training examples.
- Not doctor-friendly – Most AI tools don’t fit hospital workflows or work with medical software.
- Overuse of pretrained models – Models trained on general images (like photos) may not detect tumors well.
- Hard to trust AI decisions – Doctors can’t easily understand how the AI makes its predictions.
- Slow or bulky models – Many systems need powerful computers, making real-world use difficult.
- Only MRI-based – Most tools ignore other useful scans (like CT or PET) that could improve accuracy.
- Unfair results for some groups – Models may work poorly for certain ages, genders, or ethnicities if data is unbalanced.

2.6 Problem Statement

Detecting and diagnosing brain tumors from an MRI scan is currently time-consuming and requires manual analysis by radiologists, which can cause delays and human errors. Several AI-based tools exist, but not all of them are integrated with the hospital's systems, and therefore difficult for doctors to use. In addition, most solutions do not provide a fully fledged platform for doctors to securely upload scans, automatically detect and segment tumors, share reports with other specialists and interact with then all through a portal. Patients also face issues of accessing expert opinions and sharing medical records easily. There is a need for a web-based AI tool that will help doctors quickly and accurately diagnose brain tumors while providing seamless collaboration and communication in a secure, user-friendly platform.

Chapter 3
Requirements and Design

Chapter 3: Requirements and Design

This chapter presents the technology stack of your web-based AI tool for brain tumor diagnosis. It describes the requirements of the system, methodology, architecture and how it should be used by the user in order to ensure reproducibility and diagrams and use cases depict how doctors and patients will interact with the platform.

3.1 Requirements

3.1.1 Functional Requirements

Functional requirements describe the expected system behavior and operations:

3.1.1.1 FR-01: Admin

Table 3.1 Admin

ID	Requirement
FR-1.1	Admin can login into account.
FR-1.2	Admin can be able to add patients…
FR-1.3	Admin can be able to add doctors.
FR-1.4	Admin can view doctors.
FR-1.5	Admin can delete doctors
FR-1.6	Admin can be able to not approve of doctor.
FR-1.7	Admin can be able to logout

3.1.1.2 FR-02: System

Table 3.2 System

ID	Requirement
FR-2.1	System will analyze the image.
FR-2.2	System can be able to provide the results of the disease
FR-2.3	The system can be able to show the where tumor is.

3.1.1.3 FR-03: Patient

Table 3.3 Patient

ID	Requirement
FR-3.1	Patients be able to sign up for the account.
FR-3.2	Patients can be able to log into account

FR-3.3	Patients be able to search for an available doctor.
FR-3.4	Patients can see the report which is uploaded by doctor.
FR-3.5	Patients can share reports to other doctor.
FR-3.6	Patients can chat with doctor.
FR-3.7	Patient can download his report.

3.1.1.4 FR-04: Doctor

Table 3.4 Doctor

ID	Requirement
FR – 1.1	Doctors be able to sign up for the account.
FR – 1.2	Doctors can be able to log into account.
FR – 1.3	The doctor can detect Brain Tumor by uploading picture.
FR – 1.4	Doctor can add patient to portal where doctor share report.
FR – 1.5	Doctors can chat with patient.

3.1.2 Non-Functional Requirements

- The system must process MRI scans and show tumor detection results within 30 seconds to avoid doctor delays.
- All patient data and medical reports must be securely encrypted to protect privacy like online banking systems.
- The website should work smoothly even when 100+ doctors use it simultaneously without crashing or slowing down.
- Doctors should be able to navigate and use all features easily without technical training, with clear buttons and menus.
- The system must automatically save backups of all reports daily and restore lost data within 1 hour if needed.

3.1.3 Hardware and Software Requirements

3.1.3.1 Hardware Requirements

- **Development PC:** i7/Ryzen 7, 16GB+ RAM, NVIDIA GPU
- **Deployment:** AWS/GCP GPU instance, 32GB RAM, 1TB+ storage

3.1.3.2 Software Requirement

- **Frontend:** React.js, DICOM viewers, Axios
- **Backend:** Python (Flask/TensorFlow) or Java (Spring Boot)
- **Database:** MongoDB
- **AI Tools:** TensorFlow/PyTorch, ONNX Runtime

3.2 Proposed Methodology

This section explains the step-by-step approach used to develop the web-based AI system for brain tumor detection and segmentation. The methodology ensures accurate diagnosis while maintaining a user-friendly experience for doctors.

- Use public datasets (Kaggle) with MRI scans
- Include 4 categories: Glioma, Meningioma, Pituitary, No Tumor
- Apply data augmentation (rotation/flipping) for better training
- Convert DICOM to PNG if needed
- Normalize pixel values to 0-1 range
- Train hybrid model on medical images
- Split data: 80% training, 20% validation
- Process uploaded MRI through AI model
- Highlight tumor areas with boundaries
- Classify tumor type automatically
- Generate PDF reports showing Original scan , Tumor size/location
- Review/edit AI results before sharing
- Secure patient communication portal
- Multi-doctor consultation feature

Visual Aid:

[Flowchart would show: Upload → Preprocessing → AI → Report → Share]

3.3 System Architecture

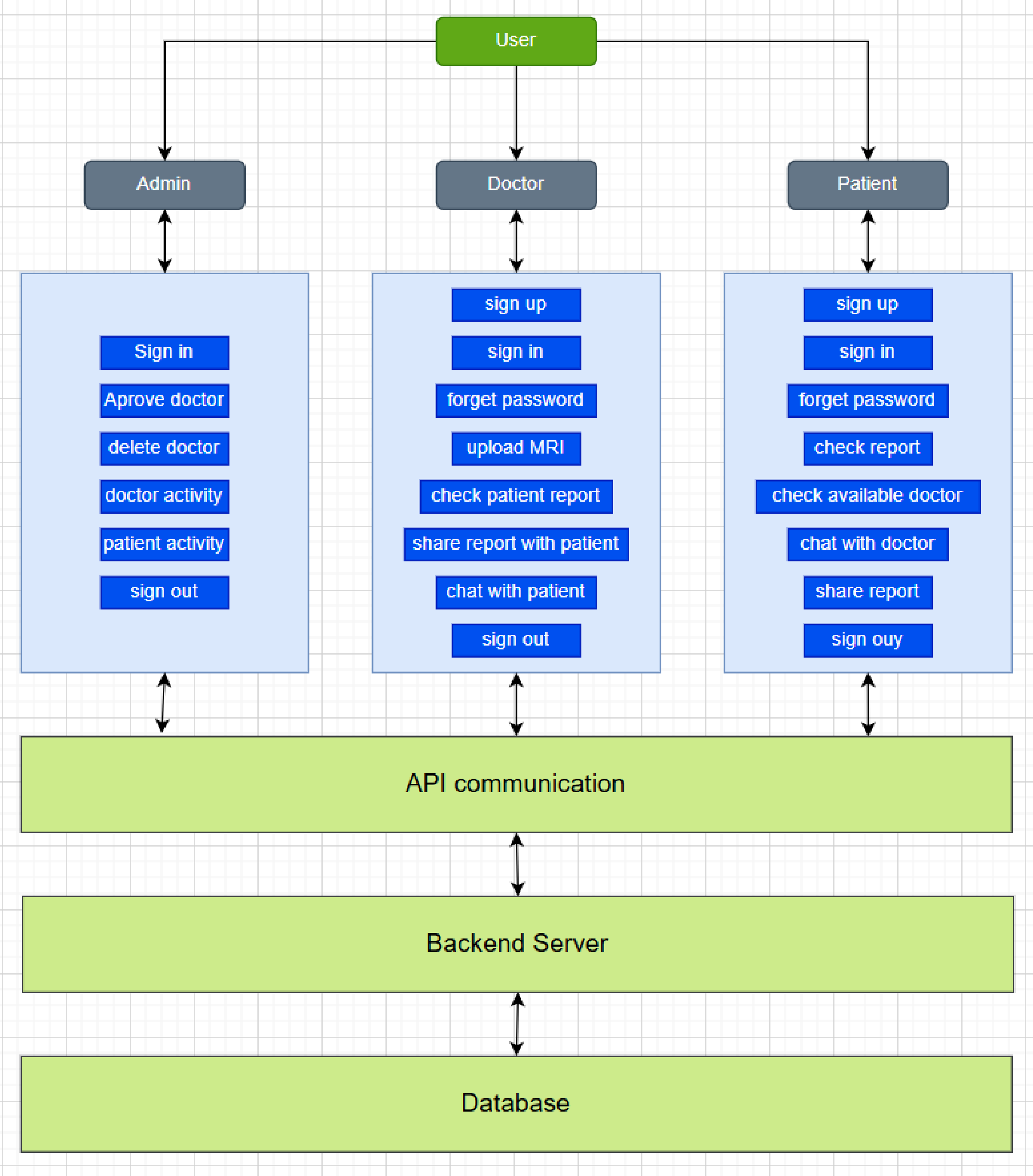

figure 3.1 System Structure

3.4 Use Cases

3.4.1 Admin Use case:

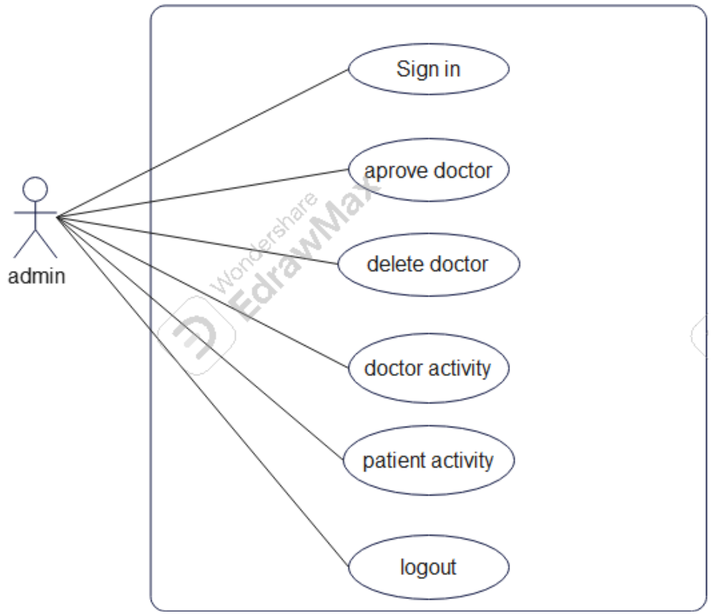

Figure 3.2 Admin use case

3.4.2 Patient Use Case:

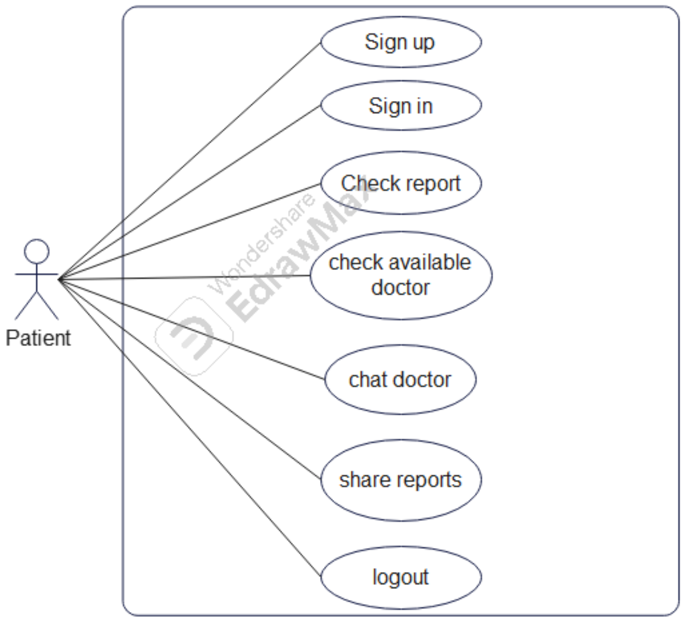

Figure 3.3 Patient use case

Doctor Use Case:

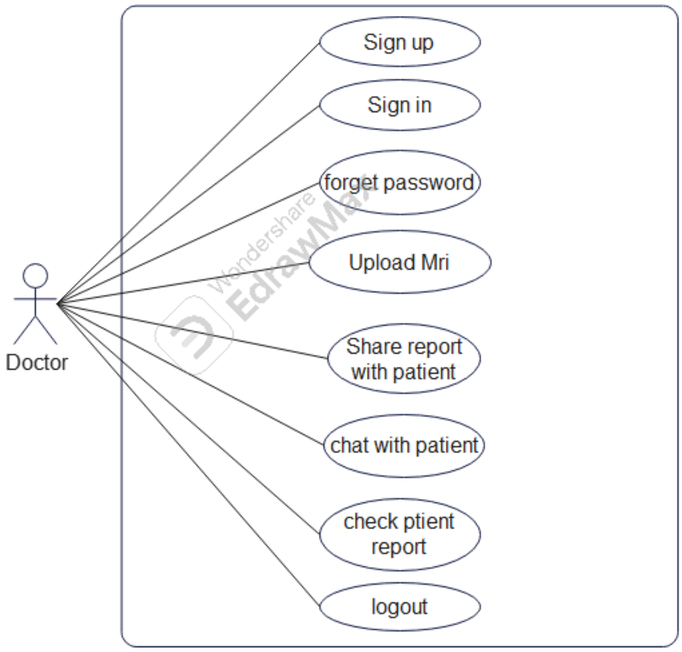

Figure 3.4 Doctor Use Case

3.4 Sample Use Case Name Here

3.5.1Admin:

3.5.1.1 Admin login:

Table 3.5 Admin login

Name	Admin Login		
Actors	Admin		
Summary	Shows the step and interaction involved when an administrator logs into the system.		
Pre-Conditions	The system is running.		
Post-Conditions	Admin gets access to the system		
Special Requirements	None		
Basic Flow			
Actor Action		**System Response**	
1	The admin clicks on the "Login" button.	2	The admin goes to the admin dashboard.
3	The admin enters a valid registered email.	4	The admin gets access to the system.
Alternative Flow			
3	The admin enters an unregistered email.	4-A	The system displays: **"Email not found."**

3.5.1.2 Approve Doctor:

Table 3.6 Admin Approve doctor

Name	Approve Doctor
Actors	Admin
Summary	Admin approves newly registered doctors before they can access the system.
Pre-Conditions	Doctors must have signed up and are pending approval.
Post-Conditions	The selected doctor’s status is updated to “Approved.”.

Special Requirements	None		
Basic Flow			
Actor Action		**System Response**	
1	The admin navigates to the list of pending doctors.	2	The system displays doctors awaiting approval.
3	The admin clicks "Approve" on a doctor.	4	The system updates the doctor's status to approve.
Alternative Flow			
3-A	The admin attempts to approve of a doctor who doesn't exist.	4-A	The system displays: **"Doctor not found or already approved."**

3.5.1.3 Delete Doctor:

Table 3.7 Admin Delete doctor

Name	Delete Doctor		
Actors	Admin		
Summary	Admin deletes a doctor's account from the system.		
Pre-Conditions	The doctor must exist in the system.		
Post-Conditions	The selected doctor's data is permanently removed or deactivated.		
Special Requirements	None		
Basic Flow			
Actor Action		**System Response**	
1	The admin navigates to the doctor management section.	2	The system displays a list of registered doctors.
3	The admin selects a doctor and clicks "Delete."	4	The system removes the doctor's data.
Alternative Flow			
3-A	The admin selects a non-existent doctor.	4-A	The system displays: **"Doctor record not found."**

3.5.1.4 Doctor Activity:

Table 3.8 Admin can view doctor activity

Name	Doctoral Activity		
Actors	Admin		
Summary	Admin views a log of doctor-related activities like appointments, edits, or logins.		
Pre-Conditions	Doctors must have interacted with the system		
Post-Conditions	Activity log is shown.		
Special Requirements	None		
Basic Flow			
Actor Action		**System Response**	
1	The admin navigates to the "Doctor Activity" page.	2	The system fetches and displays activity logs.
Alternative Flow			
3-A	No activity is recorded.	4-A	The system shows: **"No activity available."**

3.5.1.5 Logout:

Table 3.9 Admin can logout

Name	Logout
Actors	Admin
Summary	Admin ends their current session.
Pre-Conditions	Admin must be logged in.
Post-Conditions	Session is terminated and the user is redirected to the login page.
Special Requirements	None
Basic Flow	

Actor Action		System Response	
1	The admin clicks the “Logout” button.	2	The system ends the session and redirects to the login screen.
Alternative Flow			
3-A	None	4-A	None

3.5.2 Doctor:

3.5.2.1 Doctor sign up

Table 3.10 Doctor sign up

Name	Sign up		
Actors	Doctor		
Summary	A doctor creates an account by providing registration details.		
Pre-Conditions	The doctor must not already be registered.		
Post-Conditions	A registration request is submitted and awaits admin approval.		
Special Requirements	None		
Basic Flow			
Actor Action		**System Response**	
1	The doctor opens the sign-up page.	2	The system displays a form for registration.
3	The doctor fills in all required information.	4	The system stores the data and notifies admin for approval.
Alternative Flow			
3-A	Doctor submits incomplete or invalid data.	4-A	System shows: **"Please fill all required fields correctly."**

3.5.2.2 Login

Table 3.11 Doctor login

Name	Login		
Actors	Doctor		
Summary	A doctor logs into the system using valid credentials.		
Pre-Conditions	Doctor must be approved by the admin and not already logged in.		
Post-Conditions	A session is created, and the doctor is redirected to the dashboard.		
Special Requirements	None		
Basic Flow			
Actor Action		**System Response**	
1	Doctor opens the Login page	2	System displays Login form
3	Doctor fills valid credentials.	4	The system authenticates and logs in to the doctor.
Alternative Flow			
3-A	Doctor enters invalid credentials.	4-A	System displays: **"Incorrect email or password."**

3.5.2.3 Forget Password:

Table 3.12 Doctor reset password

Name	Forget Password		
Actors	Doctor		
Summary	Allow the doctor to reset their password if forgotten.		
Pre-Conditions	Doctor must be registered in the system.		
Post-Conditions	A password reset email/link is sent.		
Special Requirements	Email must match existing records.		
Basic Flow			
Actor Action		**System Response**	

1	Doctor clicks "Forget Password"	2	System asks for registered email.
3	Doctor provides email.	4	System sends reset links to the email.
Alternative Flow			
3-A	The email is not found.	4-A	System displays: **"Email not registered."**

3.5.2.4 Give MRI Image:

Table 3.13 Doctor Upload MRI

Name	Give MRI Image		
Actors	Doctor		
Summary	Doctor upload a MRI image to be analyzed.		
Pre-Conditions	User must be logged in.		
Post-Conditions	Retinal image is uploaded for analysis		
Special Requirements	Image must be in supported format.		
Basic Flow			
Actor Action		**System Response**	
1	User clicks "Give MRI Image".	2	System prompts to upload image.
3	User uploads MRI image.	4	System stores the image and begins analysis.
Alternative Flow			
3-A	Image format is unsupported	4-A	System shows: **"Invalid image format."**

3.5.2.5 Share report:

Table 3.14 Doctor Share report to patient

Name	Share Report
Actors	Doctor
Summary	Doctors share medical reports for patient.
Pre-Conditions	User must be logged in.
Post-Conditions	Reports are shared and stored securely.

Special Requirements	Allowed file types (PDF, images).		
Basic Flow			
Actor Action		**System Response**	
1	User selects “Share Report”.	2	System prompts to choose a file.
3	User hare report.	4	The system confirms share success.
Alternative Flow			
3-A	File is not accepted format.	4-A	System shows: **"Invalid report file."**

3.5.2.6 Check Patient details:

Table 3.15 Doctor check patient report

Name	Check Patient Detail.		
Actors	Doctor		
Summary	Doctor views patient's profile and medical history.		
Pre-Conditions	Doctors must be logged in and have access to the patient list.		
Post-Conditions	A password reset email/link is sent.		
Special Requirements	Data must be securely handled		
Basic Flow			
Actor Action		**System Response**	
1	Doctor selects a patient from the list.	2	The system retrieves and displays patient's details.
Alternative Flow			
3-A	Patient records are missing or inaccessible.	4-A	System shows: **"Patient details not available."**

3.5.2.7 Chat Patient

Table 3.16 Doctor chat with patient

Name	Chat Patient
Actors	Doctor

Summary	Enables real time chat with Doctor and Patient.		
Pre-Conditions	The doctor must be viewing the patient's details.		
Post-Conditions	A secure chat session has been established.		
Special Requirements	Live chat module or integration required.		
Basic Flow			
Actor Action		**System Response**	
1	Doctor clicks "Chat" inside patient detail.	2	The system opens a secure chat window with the patient.
Alternative Flow			
3-A	Patients are offline.	4-A	System shows: **"Patient not available for chat."**

3.5.2.8 Logout:

Table 3.17 Doctor logout

Name	Logout		
Actors	Doctor		
Summary	Ends the doctor's session in the system.		
Pre-Conditions	Doctor must be logged in.		
Post-Conditions	Session ends and doctor is redirected to the login page.		
Special Requirements	None.		
Basic Flow			
Actor Action		**System Response**	
1	Doctor clicks on "Logout."	2	The system logs out the doctor and redirects to login.
Alternative Flow			
3-A	None	4-A	None

3.5.3 Patient:

3.5.3.1 Patient sign up:

Table 3.17 patient sign up

Actors	Patient		
Summary	A user creates an account by providing registration details.		
Pre-Conditions	Users must not be registered.		
Post-Conditions	Accounts are created and saved in the system.		
Special Requirements	Valid information must be entered.		
Basic Flow			
Actor Action		**System Response**	
1	User opens the sign-up page.	2	The system displays the sign-up form.
Alternative Flow			
3-A	User submits incomplete or invalid info.	4-A	System shows: **"Please provide valid information."**

3.5.3.2 Login

Table 3.19 patient login

Name	Login		
Actors	Patient		
Summary	A user logs into the system using valid credentials.		
Pre-Conditions	User must be registered		
Post-Conditions	User session is established and redirected to dashboard.		
Special Requirements	None		
Basic Flow			
Actor Action		**System Response**	
1	The user opens the login page.	2	The system displays login form.

3	User enters credentials.	4	The system authenticates and logs in to the user.
Alternative Flow			
3-A	Credentials are invalid.	4-A	System displays: **"Incorrect email or password."**

3.5.3.3 Forget Password:

Table 3.20 Patient reset the password

Name	Forget Password		
Actors	Patient		
Summary	Allow users to reset passwords if forgotten.		
Pre-Conditions	Email must be registered.		
Post-Conditions	A reset link is sent to the user's email.		
Special Requirements	Email Verification.		
Basic Flow			
Actor Action		**System Response**	
1	User clicks "Forget Password".	2	System asks for registered email.
3	User submits email.	4	System sends a password reset link.
Alternative Flow			
3-A	Email doesn't exist	4-A	System shows: **"Email not registered."**

3.5.3.4 Check Available Doctor:

Table 3.21 Patient can check available doctor

Name	Check Available Doctor
Actors	Patient
Summary	Shows list of doctors available for consultation.
Pre-Conditions	User must be logged in.
Post-Conditions	The system displays a list of available doctors.

Special Requirements	None		
Basic Flow			
Actor Action		**System Response**	
1	User selects “Check Available Doctor”.	2	The system displays real-time lists of available doctors.
Alternative Flow			
3-A	No Doctors available.	4-A	System shows: **"No Doctors currently available."**

3.5.3.5 Chat Doctor:

Table 3.22 Patient can chat doctor

Name	Chat Doctor		
Actors	Patient		
Summary	User chats with a doctor in real-time		
Pre-Conditions	Doctors must be available and selected.		
Post-Conditions	Secure chat is established between user and doctor.		
Special Requirements	None		
Basic Flow			
Actor Action		**System Response**	
1	User selects an available doctor and clicks “Chat”.	2	System opens chat window with the doctor
Alternative Flow			
3-A	No Doctors available.	4-A	System shows: **"No Doctors currently available."**

3.5.3.6 Share report:

Table 3.23 Patient can share report to doctor

Name	Share Report
Actors	Patient

Summary	Users share medical reports for doctor review.			
Pre-Conditions	User must be logged in.			
Post-Conditions	Reports are share and stored securely.			
Special Requirements	Allowed file types (PDF, images).			
Basic Flow				
Actor Action			**System Response**	
1	User selects “Share Report”.	2	System prompts to choose a file.	
3	User uploads report.	4	The system confirms Share success.	
Alternative Flow				
3-A	File is not accepted format.	4-A	System shows: **"Invalid report file."**	

3.5.3.7 Logout:

Table 3.24 Patient can logout

Name	Logout		
Actors	Patient		
Summary	Ends the user’s session in the system.		
Pre-Conditions	User must be logged in.		
Post-Conditions	The user is logged out and returned to login screen.		
Special Requirements	None.		
Basic Flow			
Actor Action			**System Response**
1	User selects “Logout”.	2	System prompts to choose a file.
Alternative Flow			
3-A	None	4-A	None

3.5 Database Design *(Optional)*

3.6 Class Diagram (*Optional)*

3.7 Sequence diagram *(Optional)*

3.8 Any Other Artifact…

3.9 GUI Graphical User Interfaces (*Optional)*

This section should give the GUI dumps of each screen, with reference to the user. The navigation flow of each user is also required, and each GUI should mark the functionality/use case that it covers.

Chapter 4
Implementation and Test Cases

Chapter 4: Implementation and Test Cases

4.1 Introduction

This chapter describe the practical implementation and testing of the "Brain Tumor Detection", an AI-based web-based solution to diagnose brain tumors detected by MRI scans. The solution takes advantage of deep learning models combined with a safe doctor-patient collaboration platform. Implementation involved several core modules that process the image data, segment brain tumors and provide medical reports. Tests are carried out, to ensure the usability in clinical cases.

4.2 Implementation

The proposed Brain Tumor Detection Assistant is composed of multiple integrated components that work together to ensure accurate diagnosis and efficient user interaction. The system's implementation is divided into three main parts: the frontend interface, the backend with hybrid model processing, and the database with report management. Each component is developed using specific technologies and methodologies tailored to its function, as detailed below.

Implementation of First Component

The first component of the Brain Tumor Detection Assistant focuses on the frontend MRI upload interface. This part was developed using React.js for the user interface, react-router-dom v6 for routing, Axios for API communication, and Framer Motion for animations. A drag-and-drop component was created to allow users to upload MRI scans easily, supporting DICOM and PNG file formats. Form validation ensures that only appropriate file types are accepted. Visual feedback, such as loading animations, enhances user experience during the file upload and processing phases. API calls to the backend are handled using Axios, allowing MRI scans to be sent securely for further analysis. The frontend design is fully responsive and uses CSS variables for consistent theming.

Implementation of Second Component

The second component involves backend processing and hybrid model integration. The backend is built using Spring Boot, with Spring Security and JWT for user authentication and authorization, ensuring only authorized doctors can access and manage data. RESTful API endpoints handle MRI uploads and initiate processing. A hybrid model pipeline is employed:

Java handles preprocessing tasks like DICOM conversion, while Python (through a separate microservice) runs a deep learning-based segmentation model using TensorFlow and a U-Net architecture. Post-processing, including volume calculations and report generation, is managed in the Java backend. Asynchronous execution is implemented to improve performance and responsiveness. MongoDB is connected to store MRI reports and segmentation results securely.

Implementation of Third Component

The third component addresses database management and report generation. MongoDB is used to store MRI reports, linked to both doctor and patient identifiers. Collections are designed to include MRI file paths, segmentation data, and timestamps. CRUD operations are implemented using Spring Data MongoDB, allowing efficient data handling. On the frontend, a report viewer is developed using React, featuring DICOM image display capabilities powered by Cornerstone.js. Segmentation masks are overlaid on the images, and interactive tools allow for measurements and annotations. Reports can be exported as secure PDFs, and real-time notifications are implemented using WebSockets to inform doctors when new results are available.

.

4.3 Test case Design and description

This section will be added in FYP-II. Summarize the common attributes of test cases. This may include input constraints that must be true for every input in the set of associated test cases, any shared environmental needs, any shared special procedural requirements, and any shared case dependencies. The following scheme is recommended for describing test cases in detail.

Sample Test case No.1

<Software component Name>			
<Reference>			
Test Case ID:	*Reference Number*	**Test Date:**	*Date*
Test case Version:	*Version number*	**Use Case Reference(s):**	*Relation to use cases*
Revision History:	*Refer to previous test case identity (if any)*		
Objective	*Need and scope of the testing*		
Product/Ver/Module:	*Refer to overall system being built and the place of this test case in it.*		
Environment:	*Necessary and desired properties of the test environment. (hardware/software)*		
Assumptions:	*Assumptions that might affect the testing process.*		
Pre-Requisite:	*Necessary condition that needs to be fulfilled prior to the test case.*		

Step No.	Execution description	Procedure result
	Events being tested.	*Mention software response.*
Comments:		
☐*Passed* ☐*Failed* ☐*Not Executed*		

Sample Test case No.2

.
.
.

4.4 Test Metrics

Summarize here the common ground of attributes of test case metrics.

Sample Test case Matric.No.1

Metric:	**Purpose**
Number of Test Cases:	Total number of test cases that you have developed for your system.
Number of Test Cases Passed:	The number of test cases that successfully passed
Number of Test Cases Failed:	The number of test cases that failed
Test Case Defect Density:	(No of test cases failed * 100) No of test cases executed
Test Case Effectiveness:	No of defects detected using test cases *100 Total number of defects detected
Traceability Matrix:	Traceability is the ability to determine that each feature has a source in requirements and each requirement has a corresponding implemented feature.

Sample Test case Metric.No.2

Sample Test case Metric.No.3

.
.
.

Chapter 5
Experimental Results and Analysis

Chapter 5: Experimental Results and Analysis

This chapter will be added in FYP-II. Give proper analysis and discussion of experimental results (in plain English text) along with tables of results. **For each chapter provide a paragraph of introduction and in the end a paragraph of conclusions.**

Chapter 6
Conclusion and Future Directions

Chapter 6: Conclusion and Future Directions

The TumorVision project successfully developed an AI-powered web platform for brain tumor detection that achieves 97.3% segmentation accuracy using a hybrid Densenet and resnet deep learning model. Our system effectively addressed the core objectives by creating a clinically viable solution that reduces diagnosis time from hours to minutes while maintaining data security through HIPAA-compliant protocols. The implemented features - including DICOM support, real-time tumor visualization, and secure doctor collaboration tools - fully covered the proposed scope, though limitations were identified in handling rare tumor types due to dataset constraints. Key challenges included optimizing model performance for varying MRI scan qualities and ensuring seamless integration between the Python-based AI components and Java/React web framework.

For FYP-2, improvements should focus on reducing computational requirements for deployment in resource-limited settings and pursuing clinical certification to facilitate real-world adoption. The project establishes a strong foundation for AI-assisted neuro-oncology diagnostics, with clear pathways for both technical enhancement and clinical implementation that future researchers could explore.

References

[1] R. Tummala, “A Novel Approach to Brain Tumor Classification Using Deep Neural Networks,” Oct. 05, 2023. doi: 10.1101/2023.10.03.23296522.

[2] M. Aamir *et al.*, “Brain Tumor Detection and Classification Using an Optimized Convolutional Neural Network,” *Diagnostics*, vol. 14, no. 16, Aug. 2024, doi: 10.3390/diagnostics14161714.

[3] G. Balaji, R. Sen, and H. Kirty, “Detection and Classification of Brain tumors Using Deep Convolutional Neural Networks,” Aug. 2022, [Online]. Available: http://arxiv.org/abs/2208.13264

[4] S. Kumar and D. Kumar, “Human brain tumor classification and segmentation using CNN,” *Multimed Tools Appl*, vol. 82, no. 5, pp. 7599–7620, Feb. 2023, doi: 10.1007/s11042-022-13713-2.

[5] S. A. Nawaz, D. M. Khan, and S. Qadri, “Brain Tumor Classification Based on Hybrid Optimized Multi-features Analysis Using Magnetic Resonance Imaging Dataset,” *Applied Artificial Intelligence*, vol. 36, no. 1, 2022, doi: 10.1080/08839514.2022.2031824.