BIOINFORMATICS PROGRAM

Brain Tumor Detection

A graduation project dissertation by:

Ahmed Mostafa
Ahmed Bader
Ziad Ashraf
Yossef Khiery
Khaled Waleed
Kareem Mohamed
Raghda Rajab

Supervised by: **Dr. BASSEM MOHAMED**

June 9, 2024

Acknowledgment

First and foremost, we would like to express our heartfelt gratitude to our research supervisor, **Dr.BASSEM MOHAMED**, from the Faculty of Computer and Artificial Intelligence at Fayoum University. This project would not have been possible without their unwavering support and dedicated involvement in every step of the process. We sincerely thank **Dr.BASSEM MOHAMED** for imparting invaluable knowledge on research methodologies, technical writing, and presentation skills. Their guidance has been instrumental in shaping the success of this project.

Additionally, we would like to extend our gratitude to the Faculty of Computer and Artificial Intelligence at Fayoum University for permitting us to utilize the available resources that have greatly contributed to the progress and development of our project.

We acknowledge and appreciate the contributions of all those involved and remain indebted to their invaluable assistance and encouragement throughout this endeavor.

Abstract

Our project presents a website that assists in the early identification of brain cancer, reducing the need for immediate medical consultation and streamlining the diagnosis process. By leveraging advanced technology, users can conveniently upload brain images and receive rapid assessments, facilitating early intervention when necessary.

The website also allows healthcare professionals to register, simplifying the appointment booking process and ensuring personalized care. Through this project, we aim to improve outcomes for individuals affected by brain cancer by enabling early detection and facilitating seamless connections between patients and doctors.

TABLE OF CONTENTS

Chapter Six: System Design

Chapter Seven: Proposed Model

LIST OF FIGURES

LIST OF TABLES

CHAPTER ONE
INTRODUCTION

1.1 Introduction

In recent years, medical advancements have revolutionized the field of diagnostics, particularly in the detection and treatment of various diseases. One such area that has witnessed significant progress is the early detection and diagnosis of cancer. Timely identification of cancer plays a crucial role in improving patient outcomes and increasing the chances of successful treatment.

This graduation project aims to develop a pioneering website that harnesses the power of artificial intelligence (AI) and medical imaging to predict the presence of cancer in individuals and determine its stage. The website will provide users with a user-friendly platform to upload their brain images, which will then be analyzed by advanced machine learning algorithms to generate accurate and reliable predictions.

The primary objective of this project is to provide individuals with a quick and accessible tool for assessing their risk of having cancer and understanding the stage of the disease. By utilizing AI technology, we can expedite the diagnostic process, potentially leading to earlier intervention and improved treatment outcomes.

Additionally, this website will go beyond just providing predictions. It will also serve as a comprehensive platform for users to make appointments with doctors specializing in oncology. Users will have the freedom to choose from a list of qualified healthcare professionals available through the website. This feature aims to streamline the process of seeking medical advice and ensure that users receive prompt attention from professionals with expertise in cancer diagnosis and treatment.

The significance of this project lies in its potential to positively impact the lives of countless individuals. By combining the power of medical imaging, AI algorithms, and the convenience of an online platform, we can empower users to take control of their health and make informed decisions regarding their well-being. Early cancer detection can significantly enhance treatment outcomes and potentially save lives, and this website seeks to contribute to that crucial objective.

In the following sections of this graduation project, we will delve into the technical aspects of the website, explaining the image analysis algorithms and machine learning techniques employed. We will also discuss the development process, and challenges faced, and evaluate the performance of the system through extensive testing and validation.

Ultimately, we hope that this website will prove to be an invaluable tool in the fight against cancer, enabling early detection, and accurate prediction, and facilitating seamless connections between users and healthcare professionals.

1.2 Project Motivation

Several factors contribute to the motivation behind this graduation project. The primary motivations are:

1. **Early Detection and Improved Treatment Outcomes:** [1*]

   Early detection of cancer is crucial for successful treatment and improved patient outcomes. By developing a website that can accurately predict the presence of brain cancer and determine its stage, individuals can potentially receive timely medical intervention. This project aims to contribute to early cancer detection, leading to increased chances of successful treatment and better patient outcomes.

2. **Accessibility to Diagnostic Tools:**

Access to advanced diagnostic tools is essential for all individuals, regardless of their geographic location or socioeconomic status. By creating a website that leverages AI technology, individuals can access a user-friendly platform to upload their brain images and receive predictions about the presence of cancer. This project aims to enhance accessibility to diagnostic tools, providing a valuable resource for individuals who may not have easy access to specialized medical facilities or expertise.

3. **Empowerment and Informed Decision-Making:**

By providing individuals with a tool to assess their risk of having brain cancer and understand the stage of the disease, this project aims to empower users and enable them to make informed decisions about their health. Having access to accurate predictions can help individuals seek appropriate medical attention, plan their treatment journey, and actively participate in their healthcare decisions.

4. **Streamlined Appointment Booking Process:**

Booking appointments with healthcare professionals can be a challenging and time-consuming task, leading to delays in accessing medical expertise. By integrating an appointment booking feature within the website, users can conveniently choose from a list of qualified oncology specialists and schedule appointments directly through the platform. This streamlined process aims to improve efficiency, reduce waiting times, and ensure prompt access to medical consultation.

5. **Advancements in AI and Medical Imaging:**

AI and medical imaging have witnessed significant advancements in recent years. Applying these cutting-edge technologies to the early detection

and diagnosis of brain cancer, this project aims to leverage the progress made in these fields to benefit individuals. The motivation lies in harnessing the potential of AI algorithms and machine learning techniques to provide accurate predictions and improve healthcare outcomes.

1.3 Problem definition

The traditional diagnostic process may be very stressful for the patient as well as for the physician involved in the diagnostic process, as the patient has to, we assume that:

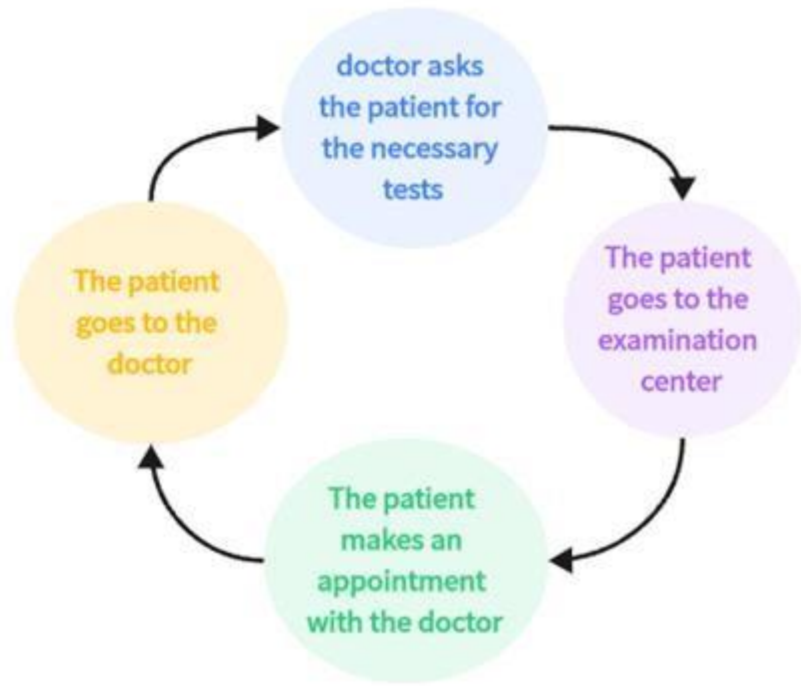


Figure 1.1

1. The patient is waiting for his appointment at the doctor for several minutes, up to an hour maybe.

2. The patient waits a few minutes to perform the examinations that the doctor asked him to do.

3. The patient waits a few days to receive his tests from the examination center .

Therefore, the average time a patient takes to undergo the examination, know his health condition, and whether or not he has brain cancer is only two to four days.

In this project, we seek to reduce the time taken in the initial examination for brain cancer and to book with the doctor whom the patient wishes to consult in his condition remotely thus we have saved the long time consumed in the process of examination, booking and waiting at the doctor and shortened it to only a few minutes.

1.4 System Objectives

The desired goal of our project is that we help the person who wants to know whether he has brain cancer or not more comfortably and respectfully of his time by the lowing:

1. helping him know the necessary tests in a faster way
2. Reducing the waiting time for the patient at the doctor to find out the diagnosis.
3. Helping the patient to find the doctor who wants to consult him .

4. Assisting the doctor in managing his time, so the patient gets a better-quality consultation.
5. Making the patient check his health condition with higher efficiency

1.5 Project Description

In light of the interest in the problem facing humanity, which is the dangerous disease of cancer, and in light of technological developments and scientific advances in the medical field and the field of computer science, we seek to combine them to present a project that can help people who want to ensure their safety from brain cancer by solving some problems that they face during The process of diagnosis and help them choose the doctor and the appointment of the consultation in an easier way.

1.6 System Features

The features offered by the project are:

1. Information about the necessary tests for brain cancer.
2. Ability to upload images of examinations.
3. Diagnosis of brain cancer
4. Choose a doctor for a consultation.
5. Choose the appropriate date for the consultation.

1.7 Stockholders

- The user wants to know whether he has brain cancer or not.
- Specialist doctors for brain cancer consultations.

CHAPTER TWO BIOLOGICAL BACKGROUND

2.1 Introduction

The brain is a complex organ that controls thought, memory, emotion, touch, motor skills, vision, breathing, temperature, hunger, and every process that regulates our body. Together, the brain and spinal cord that extends from it make up the central nervous system.

Weighing about 3 pounds in the average adult, the brain is about 60% fat. The remaining 40% is a combination of water, protein, carbohydrates, and salts. It contains blood vessels and nerves, including neurons and glial cells. [2*]

2.2 Anatomy of the brain:

The brain is made up of 3 main parts: the cerebrum, the cerebellum, and the brain stem. The meninges, which surround the brain, are also considered part of the brain.

Cerebrum:

The cerebrum (front of the brain) comprises gray matter (the cerebral cortex) and white matter at its center. The largest part of the brain, the cerebrum initiates and coordinates movement and regulates temperature. Other areas of the cerebrum enable speech, judgment, thinking and reasoning, problem-solving, emote, ions, and learning. Other functions relate to vision, hearing, touch, and other senses.

1. **Cerebral Cortex:** Cortex is Latin for "bark," and describes the outer gray matter covering the cerebrum. The cortex has a large surface area due to its folds and comprises about half of the brain's weight. The cerebral cortex is divided into two halves, o hemispheres. It is covered with ridges (gyri) and folds (sulci). The two halves join at a large, deep sulcus (the interhemispheric fissure, AKA the medial longitudinal fissure) that runs from the front of the head to the back. The right hemisphere controls the left side of the body, and

2. The left half controls the right side of the body. The two halves communicate with one another through a large, C-shaped structure of white matter and nerve pathways called the corpus callosum. The corpus callosum is in the center of the cerebrum.

3. **Frontal lobe:** The largest lobe of the brain, located in the front of the head, the frontal lobe is involved in personality characteristics, decision-making, and movement. Recognition of smell usually involves parts of the frontal lobe. The frontal lobe contains Broca's area, which is associated with speech ability.

4. **Parietal lobe:** The middle part of the brain, the parietal lobe helps a person identify objects and understand spatial relationships (where one's body is compared with objects around the person). The parietal lobe is also involved in interpreting pain and touch in the body. The parietal lobe houses Wernicke's area, which helps the brain understand spoken language.

5. **Occipital lobe:** The occipital lobe is the back part of the brain that is involved with vision.

6. **Temporal lobe:** The sides of the brain, and temporal lobes are involved in short-term memory, speech, musical rhythm, and some degree of smell recognition.

2.3 Brainstem

The brainstem (middle of the brain) connects the cerebrum with the spinal cord. The brainstem includes the midbrain, the pons, and the medulla.

1. **Midbrain:** (or mesencephalon) is a very complex structure with a range of different neuron clusters (nuclei and colliculi), neural pathways, and other structures. These features facilitate various functions, from hearing and movement to calculating responses and environmental changes. The midbrain also contains the substantia nigra, an area affected by Parkinson's disease that is rich in dopamine neurons and part of the basal ganglia, which enables movement and coordination.

2. **Pons:** The pons is the origin of four of the 12 cranial nerves, which enable a range of activities such as tear production, chewing, blinking, focusing vision, balance, hearing, and facial expression. Named for the Latin word for "bridge," the pons is the connection between the midbrain and the medulla.

3. **Medulla:** At the bottom of the brainstem, the medulla is where the brain meets the spinal cord. The medulla is essential to survival. Functions of the medulla regulate many bodily activities, including heart rhythm, breathing, blood flow, and oxygen and carbon dioxide levels. The medulla produces reflexive activities such as sneezing, vomiting, coughing, and swallowing.

2.4 Cerebellum

The cerebellum ("little brain") is a fist-sized portion of the brain located at the back of the head, below the temporal and occipital lobes, and above the brainstem. Like the cerebral cortex, it has two hemispheres. The outer portion contains neurons, and the inner area communicates with the cerebral cortex. Its function is to coordinate voluntary muscle movements and to maintain posture, balance, and equilibrium. New studies are exploring the cerebellum's roles in thought, emotions, and social behavior, as well as its possible involvement in addiction, autism, and schizophrenia.

2.5 Brain Coverings: (Meninges)

Three layers of protective covering called meninges surround the brain and the spinal cord.

1. **The outermost layer:(the dura mater)** is thick and tough, It includes two layers: The periosteal layer of the dura mater lines the inner dome of the skull (cranium) and the meningeal layer is below that. Spaces between the layers allow for the passage of veins and arteries that supply blood flow to the brain.
2. **The arachnoid mater:** is a thin, weblike layer of connective tissue that does not contain nerves or blood vessels. Below the arachnoid mater is the cerebrospinal fluid, or CSF. This fluid cushions the entire central nervous system (brain and spinal cord) and continually circulates these structures to remove impurities.
3. **<u>The pia mater:</u>** is a thin membrane that hugs the surface of the brain and follows its contours. The pia mater is rich with veins and arteries.

Cancer:

Is a disease in which some of the body's cells grow uncontrollably and spread to other parts of the body.

Cancer can start almost anywhere in the human body, which is made up of trillions of cells. Normally, human cells grow and multiply (through a process called cell division) to form new cells as the body needs them. When cells grow old or become damaged, they die, and new cells take their place.

Sometimes this orderly process breaks down, and abnormal or damaged cells grow and multiply when they shouldn't. These cells may form tumors, which are lumps of tissue. Tumors can be cancerous or not cancerous (benign).

Cancerous tumors spread into, or invade, nearby tissues and can travel to distant places in the body to form new tumors (a process called metastasis). Cancerous tumors may also be called malignant tumors. Many cancers form solid tumors, but cancers of the blood, such as leukemias, generally do not.

Benign tumors do not spread into, or invade, nearby tissues. When removed, benign tumors usually don't grow back, whereas cancerous tumors sometimes do. Benign tumors can sometimes be quite large, however. Some can cause serious symptoms or be life-threatening, such as benign tumors in the brain.

2.6 Types of Cancer:

1. **Sarcoma:** Sarcomas are cancers that form in bone and soft tissues, including muscle, fat, blood vessels, lymph vessels, and fibrous tissue (such as tendons and ligaments)

2. **Leukemia:** Cancers that begin in the blood-forming tissue of the bone marrow are called leukemias. These cancers do not form solid tumors. Instead, large numbers of abnormal white blood cells (leukemia cells and leukemic blast cells) build up in the blood and bone marrow, crowding out normal

3. blood cells. The low level of normal blood cells can make it harder for the body to get oxygen to its tissues, control bleeding, or fight infections.

4. **Lymphoma:** Lymphoma is a cancer that begins in lymphocytes (T cells or B cells). These are disease-fighting white blood cells that are part of the immune system. In lymphoma, abnormal lymphocytes build up in lymph nodes and lymph vessels, as well as in other organs of the body.

5. **Multiple Myeloma:** Multiple myeloma is cancer that begins in plasma cells, another type of immune cell. The abnormal plasma cells, called myeloma cells, build up in the bone marrow and form tumors in bones all through the body. Multiple myeloma is also called plasma cell myeloma and Kahler disease.

6. **Melanoma:** Melanoma is a cancer that begins in cells that become melanocytes, which are specialized cells that make melanin (the pigment that

gives skin its color). Most melanomas form on the skin, but melanomas can also form in other pigmented tissues, such as the eye.

Brain tumor

A brain tumor is a collection, or mass, of abnormal cells in your brain. Your skull, which encloses your brain, is very rigid. Any growth inside such a restricted space can cause problems.

Brain tumors can be cancerous (malignant) or noncancerous (benign). When benign or malignant tumors grow, they can cause the pressure inside your skull to increase. This can cause brain damage, and it can be life-threatening. Brain tumors can be cancerous (malignant) or noncancerous (benign).

Brain tumors are categorized as:

1. **A primary brain tumor** originates in your brain. Many primary brain tumors are benign.
2. **A secondary brain tumor** also known as a metastatic brain tumor, occurs when cancer cells spread Trusted Source to your brain from another organ, such as your lung or breast.

Noncancerous and Cancerous:

1. **Noncancerous (benign):** can cause many serious issues, they are not cancerous, meaning that they grow slowly and don't typically spread to other tissues. They also usually have more clearly defined borders, making them easier to remove surgically, and they don't usually come back after removal. these are low grades (grade 1 or 2)

2. **Cancerous (malignant):** grow rapidly and can spread to other parts of your brain or central nervous system, which can cause life-threatening complications. These are high grades (grade 3 or 4).

Types of brain tumors:

1. **Primary brain tumors:** Primary brain tumors originate in your brain.

1. Brain cells
2. the membranes that surround your brain, which is called meninges
3. nerve cells
4. glands, such as the pituitary of the pineal

5. Primary tumors can be benign or cancerous. In adults, the most common
6. types of brain tumors are gliomas and meningiomas.

2. **Gliomas**:

Gliomas are tumors that develop from glial cells. These cells normally:

1. support the structure of your central nervous system
2. provide nutrition to your central nervous system
3. clean cellular waste
4. break down dead neurons

Gliomas can develop from different types of glial cells:

1. astrocytic tumors, such as astrocytomas, which originate in the cerebrum
2. oligodendroglia tumors, which are often found in the frontal temporal lobes

3. glioblastomas, which originate in the supportive brain tissue and are the most aggressive type

3. **Other primary brain tumors:**

   1. pituitary tumors, which are usually benign
   2. pineal gland tumors, which can be benign or malignant
   3. ependymomas, which are usually benign
   4. craniopharyngiomas, which occur mostly in children and are benign but can have clinical symptoms like changes in vision and premature puberty
   5. primary central nervous system (CNS) lymphomas, which are malignant
   6. meningiomas, which originate in the meninges

4. **Secondary brain tumors:**

   Secondary brain tumors make up the majority of brain cancers; they start in one part of the body and spread, or metastasize, to the brain:

   1. lung cancer
   2. breast cancer
   3. kidney cancer
   4. skin cancer

2.7 The symptoms of a brain tumor:

Symptoms of brain tumors depend on the location and size of the tumor. Some tumors cause direct damage by invading brain tissue and some tumors cause pressure on the surrounding brain.

Common symptoms include:

1. headaches

2. seizures (fits)
3. persistently feeling sick (nausea), being sick (vomiting), and drowsiness
4. mental or behavioral changes, such as memory problems or changes in personality
5. progressive weakness or paralysis on one side of the body
6. vision or speech problems

CHAPTER THREE
RELATED WORK

3.1 Introduction

The brain is he significant part of our central nervous system that controls all our functionalities through a huge number of connected neurons. Any malfunction or abnormality in the brain cells affects the organs connected to the corresponding part of the brain, consequently damaging the functionalities of that organ. Cancer originating in the brain and other nervous systems is considered to be the 10th leading cause of death. The 5-year survival rate of patients having a cancerous brain is only 36%. As brain tumor is caused by the unnatural and uncontrolled growth of brain cells, its severe consequences can be life-threatening. Around 400,000 people are affected by a brain tumor and 120,000 people have died in the past years all over the world, as reported by the y World Health Organization (WHO). Early and proper detection can play an indispensable role in increasing the survival rate by accelerating the treatment process. [2*]

Manual detection of brain tumors can be tedious, time-consuming, and erroneous due to the variations in types and sizes. Proper and precise detection needs expertise and it's even harder for complicated cases. Hence, besides human inspection, we can't avoid the necessity of an automated process for the precise detection and classification of brain tumors. Deep learning and convolutional neural network can significantly accelerate the whole diagnosis process making the classification task automated and conscientious.

3.2 Data set

This dataset contains a total of 5732 MRI images of three types of brain tumors: meningioma, glioma, pituitary, and normal brain tissue. These images are the combination of T1 (type of MRI where fat tissue is highlighted and seems brighter), T2 (type of MRI where fat tissue and water are highlighted and seem brighter), and Flair types (same as T2 with free-flowing water and fat seem dark), which is available in Ref. [3*]

3.3 Summarization of Recent Works on Different Brain Tumors

3.3.1 Badža et al [4*]

Badža's research aimed to classify three tumor types from an imbalanced database using a convolutional neural network (CNN). Their goal was to develop a model that could be used in clinical diagnostics and on mobile platforms, which required a simpler network that could be trained and implemented with fewer resources.

To preprocess the magnetic resonance images from the database, the images were normalized and resized to 256 x 256 pixels. To augment the dataset, each image was transformed by rotating it by 90 degrees and flipping it vertically, resulting in three times the original dataset size. The resulting dataset consisted of 9192 images.

The CNN architecture used by Badža consisted of 22 layers, including 4 convolutional layers, each followed by a ReLU activation function, dropout layer, and max pooling layer. They used the Adam optimizer with a mini-batch size of 16 and an initial learning rate of 0.0004.

The proposed model achieved an accuracy of 89.45%. The precision was 90%, the recall was 88%, and the F1 score was 0.885.

3.3.2 Ayadi et al [5*]

In Ayadi's research, they proposed a CNN-based model for multi-class brain tumor classification, which required minimal pre-processing. The model was evaluated on three brain tumor datasets, and various performance metrics were studied to evaluate its accuracy and robustness.

The proposed model consisted of 10 blocks of convolutional layers, ReLU activation functions, and batch normalization layers, each followed by a max pooling layer. The model used the Adagrad optimizer with a learning rate of 0.003, a batch size of 16, and 20 epochs.

The input to the model was an image with a size of 256 x 256 pixels. The model used 3 x 3 filters in the convolutional layers and 2 x 2 filters in the pooling layers. Non-linearity layers were added to improve the model's fitting ability, and batch normalization was used after each convolutional layer to optimize the results and speed up network convergence. The model also included fully connected layers with 64 neurons and a SoftMax classifier as the output layer.

The proposed model achieved an accuracy of 95.71% and a validation accuracy of 85%. The precision, recall, and F1 score were 0.957, 0.957, and 0.955, respectively. The high accuracy of the proposed model demonstrated its effectiveness for brain tumor classification with minimal pre-processing compared to other techniques that require tumor segmentation before classification.

Overall, Ayadi's work highlights the potential of CNN-based models for accurate and efficient brain tumor classification, which can have significant implications for diagnosis and treatment planning.

3.3.3 Sultan et al [6*]

Sultan et al proposed a CNN-based model for brain tumor classification. The CNN architecture consisted of 16 layers, each followed by a Rectified Linear Unit (ReLU) activation function and a max pooling layer. The model used L2 regularization and stochastic gradient descent optimizer.

Before feeding the images into the CNN, a pre-processing step was performed. The original images were downsized from 512 x 512 x 1 pixels to 128 x 128 x 1 pixels to reduce dimensionality and computation time. The data was shuffled and split into training, validation, and test sets with their target labels. The images of Study I were then augmented using geometric augmentation and grayscale distortion (salt noise) to increase the dataset size. The final dataset consisted of 15,320 images.

The CNN model achieved an accuracy of 96%, with a precision of 0.90, recall of 0.91, and F1 score of 0.90. The high accuracy of the proposed model demonstrated its effectiveness for brain tumor classification. The use of data augmentation techniques helped to increase the robustness of the model and prevent overfitting.

Overall, Sultan et al.'s work highlight the potential of CNN-based models for accurate and efficient brain tumor classification. The use of data pre-processing and augmentation techniques can further improve the model's performance and robustness.

3.3.4 Kumar et al [7*]

Proposed a ResNet-50 model with global average pooling for brain tumor classification. The ResNet-50 model was initialized with weights from the ImageNet dataset and fine-tuned on the brain tumor dataset. The model used five types of convolution blocks with varying convolution layers and filter sizes, and global average pooling at the output layer. The stochastic gradient descent with momentum optimizer was used for faster convergence during training.

The proposed model achieved an accuracy of 95%, with a precision of 0.95, a recall of 0.95, and an F1 score of 0.95. The use of ResNet-50 with global average pooling helped to handle the vanishing gradient problem and reduce the computational burden for large data applications. The fine-tuning of pre-trained weights from the ImageNet dataset also helped to improve the performance of the model.

Overall, Kumar et al.'s work highlight the potential of transfer learning and ResNet50 with global average pooling for accurate and efficient brain tumor classification. The use of pre-trained weights and global average pooling can help to reduce computational time and prevent overfitting.

3.3.5 Abiwinanda et al [8*]

used a CNN architecture with 2 convolutional layers, ReLU activation function, max pooling layer, and Adam optimizer for brain tumor classification. Five different architectures were tested, each with a different depth of convolution layers and fully connected layers. The input images were downsized to 64x64 pixels to reduce the computational cost, and all convolution layers used 32 filters of size 3x3.

The output layer had 3 neurons for classifying glioma, meningioma, and pituitary tumors, with SoftMax activation function.

The proposed model achieved an accuracy of 88%, with a precision of 0.89, a recall of 0.885, and a Cana of 0.88. The use of simpler CNN architectures with fewer trainable parameters helped to reduce computational costs and improve efficiency. The Adam optimizer was chosen for its ability to handle sparse gradients on noisy problems The cross-entropy method was used for loss calculation.

Overall, Abiwinanda et al.'s work highlight the potential of simpler CNN architectures and Adam optimizers for accurate and efficient brain tumor classification. The use of hyperparameter optimization can further improve the performance of the model.

3.3.6 Deepak et al [9*]

used a CNN architecture with 5 convolutional layers and 2 fully connected layers for brain MRI classification, followed by an SVM for classification. The CNN model has an input layer of size and uses multiple filters in each convolution layer to capture multiple activations for the same input image. The model accounts for computational complexity and memory requirements, limiting it to having smaller convolution filters and two FC layers.

The CNN uses a SoftMax activation function after the final FC layer, and the SVM is a maximum margin classifier with a one-versus-all approach for the 3-class classification task. The loss function used for the SVM classification algorithm is Hinge Loss, and the model incorporates L2 regularization to enable modeling with a soft margin. The SVM model uses bfgs as a solver for optimization.

The proposed model achieved an accuracy of 96.3%, with a precision of 0.964, a recall of 0.963, and d F1 score of 0.963. The use of both CNN and SVM helped to extract features and classify trained MRI images accurately. The incorporation of error-correcting output codes (ECOC) further improved the classification accuracy.

Overall, Deepak et al.'s work highlight the potential of combining CNN and SVM for accurate brain MRI classification. The consideration of computational complexity and memory requirements in the CNN design can improve efficiency. The use of hyperparameter optimization and regularization can further improve the performance of the model.

3.4 Summarizes the Presented Exhaustive

Table: summarizes the presented exhaustive survey of state-of-the-art studies related to Brain tumors based on deep Learning

Author	year	Data	number classes	Method used	Precision	Recall	F1 score	Accuracy
Sultan et al	2019	Brain tumor	4	Deep neural network	.90	.91	90	96
Abiwinanda et al	2019	Brain tumor	4	CNN	.889	.88	.88	88
Badža et al	2020	Brain tumor	4	CNN	.90	.951	.95	95
Deepak et al	2020	Brain tumor	4	CNN and SVM	0.964	0.963	0.963	96.3
Kumar et al	2021	Brain tumor	4	Resnet-50	0.95	0.951	0.95	95
Ayadi et al tumor	2021	Brain	4	CNN	. 957	. 957	. 957	95.7

Table 3.1 summarizes state-of-the-art studies related to Brain tumors based on deep Learning.

CHAPTER FOUR

PROJECT MANAGEMENT

4.1 Introduction

Project management is the application of processes, methods, skills, knowledge, and experience to achieve specific project objectives according to the project acceptance criteria within agreed parameters.

In this chapter, we will discuss the plan that we developed to reach the specific objectives of our project and how to complete a project in a certain timeframe by scheduling the project into phases that are completed using a specific project management methodology that we will discuss in the t chapters.

4.2 Project Plan

Gathering Requirements

Identifying our project's exact requirements from start to finish.

System Analysis

-functional and non-functional requirements.

- DFD diagram.

-Use case diagram.

-UI/UX.

System Design

-Sequence diagram.

-Class diagram.

-Activity diagram.

Database Design

ERD Model.

Implementation -

Database.

-Back-end.

-Front-end. **Testing**

4.2.1 Gantt Chart

		Task Mode	Task Name	Duration	Start	Finish	Predecessors	Resource Names	Add New Column
1			**Gathering Requirments**	**1 mon**	**Thu 10/11/22**	**Wed 07/12/22**			
2			Gather Inforamtion About the Project	20 days	Thu 10/11/22	Wed 07/12/22		All Members	
3			Gather Data Needed For Esablishing The Project	10 days	Thu 08/12/22	Wed 21/12/22	2	All Members	
4			**Planning**	**10 days**	**Wed 30/11/22**	**Tue 13/12/22**		ahmed mostafa	
5			Creat Plan	5 days	Wed 07/12/22	Tue 13/12/22	6	ahmed mostafa	
6			Divide Project into Tasks	3 days	Fri 02/12/22	Tue 06/12/22	7	ahmed mostafa	
7			Make Project Schedual	2 days	Wed 30/11/22	Thu 01/12/22			
8			**System Analysis**	**10 days**	**Sat 10/12/22**	**Thu 22/12/22**			
9			Determining Functional & Non Functiona Requirments	5 days	Sat 10/12/22	Thu 15/12/22		ahmed bader	
10			Data Flow Diagram DFD	5 days	Fri 16/12/22	Thu 22/12/22	9	ziad	
11			Use Case Diagram						
12			**Design**	**30 days**	**Fri 23/12/22**	**Fri 03/02/23**		ahmed mostafa	
13			Sequance Diagram	10 days					
14			Class Diagram	10 days					
15			Activity Diagram	10 days					
16			**Database Design**	**15 days**	**Sat 04/02/23**	**Thu 23/02/23**		ziad	
17			Entity Relatioship Diagram ERD	6 days	Sat 04/02/23	Fri 10/02/23			
18			UI/UX	9 days	Mon 13/02/23	Thu 23/02/23	17		
19			**Implementation**	**2 mons**	**Tue 17/01/23**	**Mon 13/03/23**		ahmed bader	
20			website	1 mon	**Tue 17/01/23**	Mon 13/02/23		ziad	
21			Backend	1 mon	**Tue 17/01/23**	Mon 13/02/23		raghda rajab	
22			**API**	**40 days**	**Wed 15/02/23**	**Tue 11/04/23**		khaled	
23			Services of API	40 days	**Wed 15/02/23**	**Tue 11/04/23**			
24			Fast API	40 days	**Wed 15/02/23**	**Tue 11/04/23**		khaled	
25			**ML model**	**2 mons**	**Sat 11/02/23**	**Thu 06/04/23**		yossef khiery	
26			Annotation	10 days	Sat 11/02/23	Thu 23/02/23		yoosef	
27			Moldel Building	1 mon	Fri 24/02/23	Thu 23/03/23	26	karim	
28			Model Training	20 days	Fri 24/03/23	Thu 20/04/23	27		
29			Testing	2 wks	Fri 07/04/23	Thu 20/04/23		karim	

Figure 4.2 Gantt chart

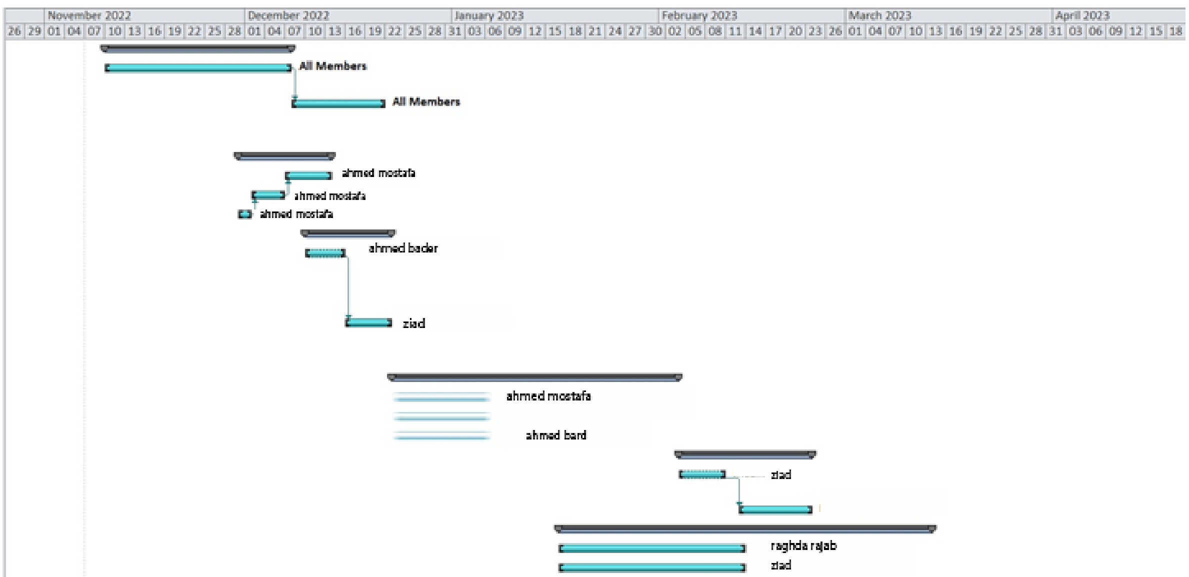


Figure 4.3 Gantt chart

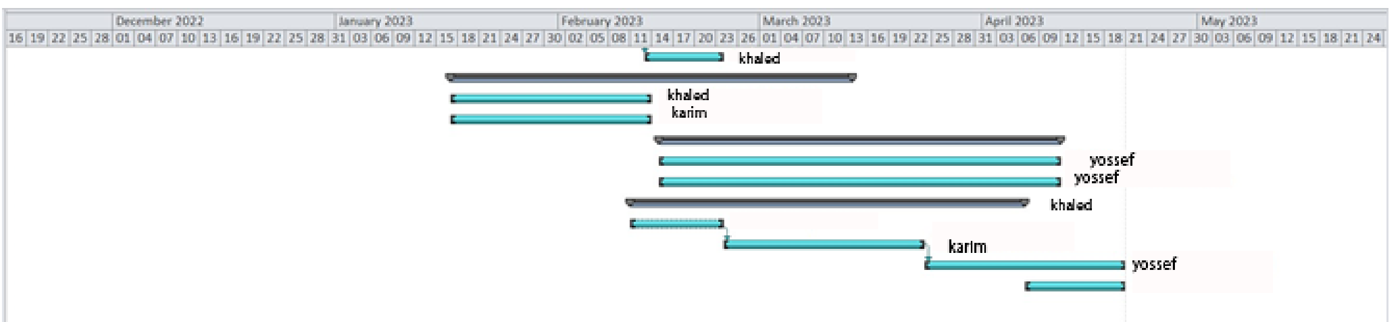


Figure 4.4 Gantt chart

4.3 Feasibility Study

A feasibility study is an assessment of the practicality of a proposed plan or project. A feasibility study analyzes the viability of a project to determine whether the project or venture is likely to succeed. The study is also designed to identify potential issues and problems that could arise while pursuing the project.

4.3.1 Description of products and services

A project that aims to educate people about the symptoms of brain tumors and help them in the initial diagnosis of brain tumors by using ML skills and facilitating the process of booking a doctor's consultation, and thus aims to reduce wasted time for people who want to ensure their safety and organize doctors' time to make patients get a better service.

4.3.2 Service Marketplace

This project will be directed toward patients and doctors to facilitate their task of diagnosis and consultation

4.3.3 Organization and Staffing

Include an integrated team, each of whom has an effective role. Positions are distributed according to each person's ability to perform the assigned tasks.

Position 1: Team Leader – He manages, distributes tasks, handles team members, solves internal problems, and leads the team well.

Position 2: UI/UX Designer -a member who creates the user interface designs of both Website and Mobile applications.

Position 3: A Frontend developer - a member who creates the front end of the website

Position 4: A back-end developer - They will design and create a database, create, and implement API services.

Position 5: Reporter - a member who documents all the steps that happen in the project.

Figure 4.5 Project team

CHAPTER FIVE

SYSTEM ANALYSIS

5.1 Overview

Through this chapter we are going through the project requirements that the system must satisfy are of two types, which are functional and non-functional requirements. Functional requirements are the requirements that define a function of the software that runs on the system. Non-functional requirements are the requirements that specify criteria that can be used to judge the operation of the system, rather than specific behaviors. In the following subsections, both functional and non-functional requirements of the proposed system are listed.

In addition, it includes Diagramming software and hardware requirements that allows users to create detailed diagrams which involve Data flow Diagrams that map out the flow of information for any process or system without a clear time flow of the operations and UML Diagrams based on the UML (Unified Modeling Language) to visually represent a system along with its main actors, roles, actions, artifacts or classes, to better understand, alter, maintain, or document information about the system by accentuating time flow of the system operations.

5.2 Process Modeling

5.2.1 Context Diagram:

The Context Diagram shows the system under consideration as a single high-level process and then shows the relationship that the system has with other external entities.

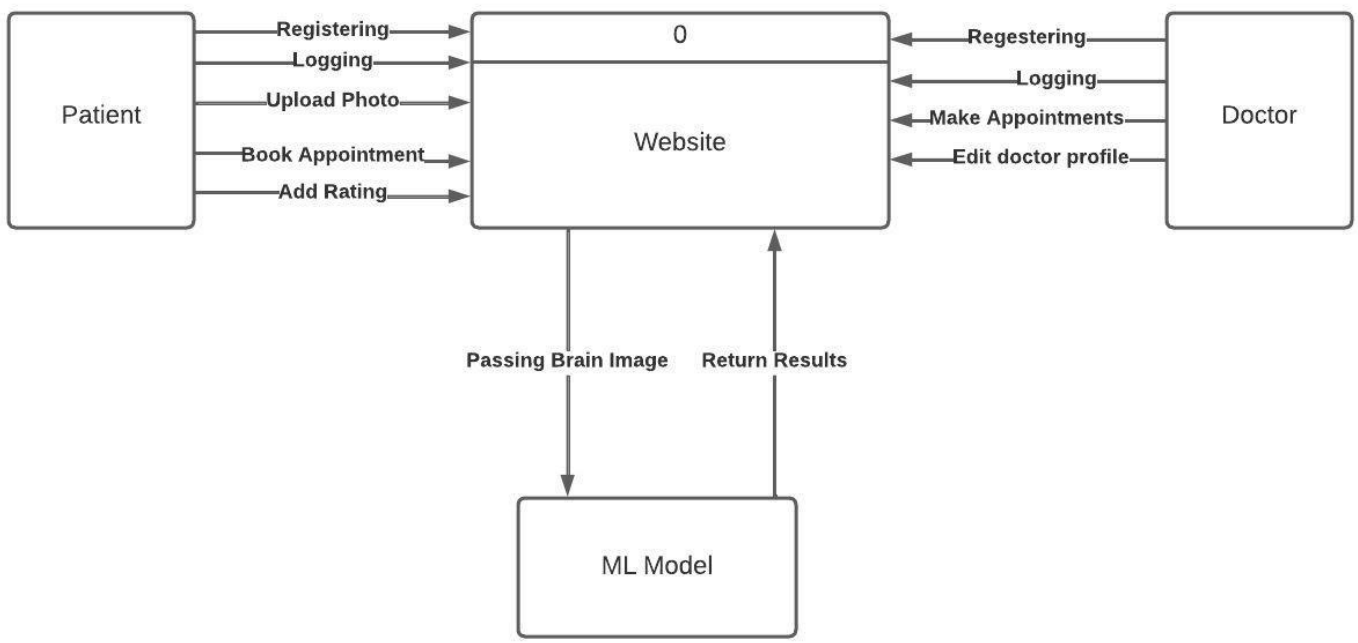


Figure 5.1 Context diagram

5.3 Requirements

5.3.1 -Functional Requirements:

Functional requirement defines a system or its component. It describes the functions a software must perform. A function is nothing but inputs, behavior, and outputs. It can be a calculation, data manipulation, business process, user interaction, or any other specific functionality which defines what function a system is likely to perform.

Patient: -

- Login

Patients can log in to the system.

- Signup

The patient can register their data to the system

- Logout

The patient can log out from his account to log in with another account.

- Upload MRI images

The patient can upload MRI to the website to diagnose his condition.

- Detect diseases

The website provides the detection of brain diseases and Information to heal them.

- Search for a doctor

The patient can search for the nearest doctor to him and book an appointment with him.

The patient can make an appointment with the available doctors.

- Book appointments

Doctor: -

- Login

A doctor can log in to the website.

- Signup

A doctor can register their data on the website.

- Logout

A doctor can log out from his account to log in with another account.

- Profile of the doctor

The website creates profiles for doctors available in the application.

- Organizing appointments with the doctor

The website knows the available appointments with the doctor to show them to the patients.

- Updating patient profile

During the examination of the patient and knowledge of his medical history, the doctor puts this data on the website to facilitate the re-examination again.

- Provide articles and advice to take care of our health.

The website provides articles about brain tumor information and advice for the best care of our health.

5.3.2 Non-Functional Requirements:

A non-functional requirement is essential to ensure the usability and effectiveness of the entire software system. Failing to meet non-functional requirements can result in systems that fail to satisfy user needs.

- Security

The system must support store and personal data security by providing login and registration functions and then validating these data.

- Usability

The system must provide a coherent user interface. The system must be obvious, allowing the user to navigate the system easily.

- Reliability

The system must perform and maintain its functions in routine circumstances.

- Scalability

The system must be scalable and able to handle any huge amount of data without any failure.

- Maintainability

Support system and solve any problems showing a duty to ensure better quality.

- High performance

The application must support high performance. It will work easily and at normal speed without any hanging. We will reduce the system's possibility of failure as possible as we can.

5.3.3 Use Case Diagram

- Use cases represent system functionality from the user's perspective • Use Case diagrams to describe who will use the system and in what ways the user expects to interact with the system.

Use Case diagrams represent the interactions between use cases and

actors.

- Use Case diagram represents the interactions between systems, external systems, and users.


Brain Tumer Detection
Registering
include
Check email& password
include
Logging
Take Appointment
include
Check valid appointments
Cancel appointment
Rate doctor
Search for a doctor
Upload brain image
Edit Personal profile
Set his appointments
View patient profile
Add note to his patient
Detect brain tumor
Determines level of tumer
patient
Doctor
ML Model

Figure 5.2 Patient, doctor, and ML model use case diagram

5.3.4 Use Case Scenario

A use case Scenario represents the sequence of events along with other information that relates to this use case. A typical use case specification template includes the following information:

- -Description
- -Pre- and Post- interaction condition
- -Basic interaction path

Use case name:	Registration
Actors:	**User**
Uses:	**Database (saving)**
Extends:	**Nil**
Preconditions:	**Nil**
Main Scenario:	**1- Enter Name.** **2- Enter the Password.** **3- Enter E-mail.** **4- Enter the Address.**
Postconditions:	**Remember this password to use when entering your account again.**
Error condition:	**Enter your info again**

Table 5.1 Registration use case scenario

Use case name:	Log in
Actors:	**User**
Uses:	**Database (checking)**
Extends:	**Nil**
Preconditions:	**Make sure you have an e-mail**
Main Scenario:	1- **Enter E-mail.** 2- **Enter the Password.**
Postconditions:	**Benefit and use the application.**
Error condition:	**Re-entering the e-mail or password.**

Table 5.2 Login use case scenario

Use case name:	Upload image
Actors:	**Patient**
Uses:	**Database (saving)**
Extends:	**Nil**
Preconditions:	**Log in**
Main Scenario:	1- **Press the upload image button.** 2- **Choose the intended image.** 3- **Confirm the chosen image.**
Postconditions:	**Wait until the result appears.**
Error condition:	**Repeat the uploading process or choose the right image.**

Table 5.3 Upload image use case scenario

Use case name:	Get the result
Actors:	**Patient**
Uses:	**Nil**
Extends:	**Nil**
Preconditions:	**Upload image**
Main Scenario:	**Wait until the model give you the result**
Postconditions:	**If he has cancer start to view available doctor**
Error condition:	**Repeat the uploading process or choose the right image.**

Table 5.4 Get Result Use case scenario

Use case name:	View available doctor
Actors:	**Patient**
Uses:	**Database (checking)**
Extends:	**Nil**
Preconditions:	**Get the result**
Main Scenario:	**Start to see available doctors.**
Postconditions:	**Book**
Error condition:	**Repeat the uploading process or choose the right image.**

Table 5.5 View available doctors use case scenario

Use case name:	Book Appointment

Actors:	**Patient**
Uses:	**Database (checking)**
Extends:	**Nil**
Preconditions:	**Get the result**
Main Scenario:	**Start to get to the user-available doctor.**
Postconditions:	**Nil**
Error condition:	**Reload the page**

Table 5.6 Book Appointment Use case scenario

Use case name:	manage appointment
Actors:	**Doctor**
Uses:	**Database (saving)**
Extends:	**Nil**
Preconditions:	**Log in**
Main Scenario:	**Modify his appointment**
Postconditions:	**Nil**
Error condition:	**Reload the page**

Table 5.7 Manage Appointments use case scenario

Use case name:	**Log out**

Actors:	**User**
Uses:	**Database (checking)**
Extends:	**Nil**
Preconditions:	**Make sure you have an e-mail**
Main Scenario:	**Log out from the application**
Postconditions:	**Nil**
Error condition:	**Reload the page**

Table 5.8 Logout use case scenario

5.3.5 Activity Diagram

An Activity Diagram is used to describe the sequential flow of activities of a use case (flow of functionality) in a system.

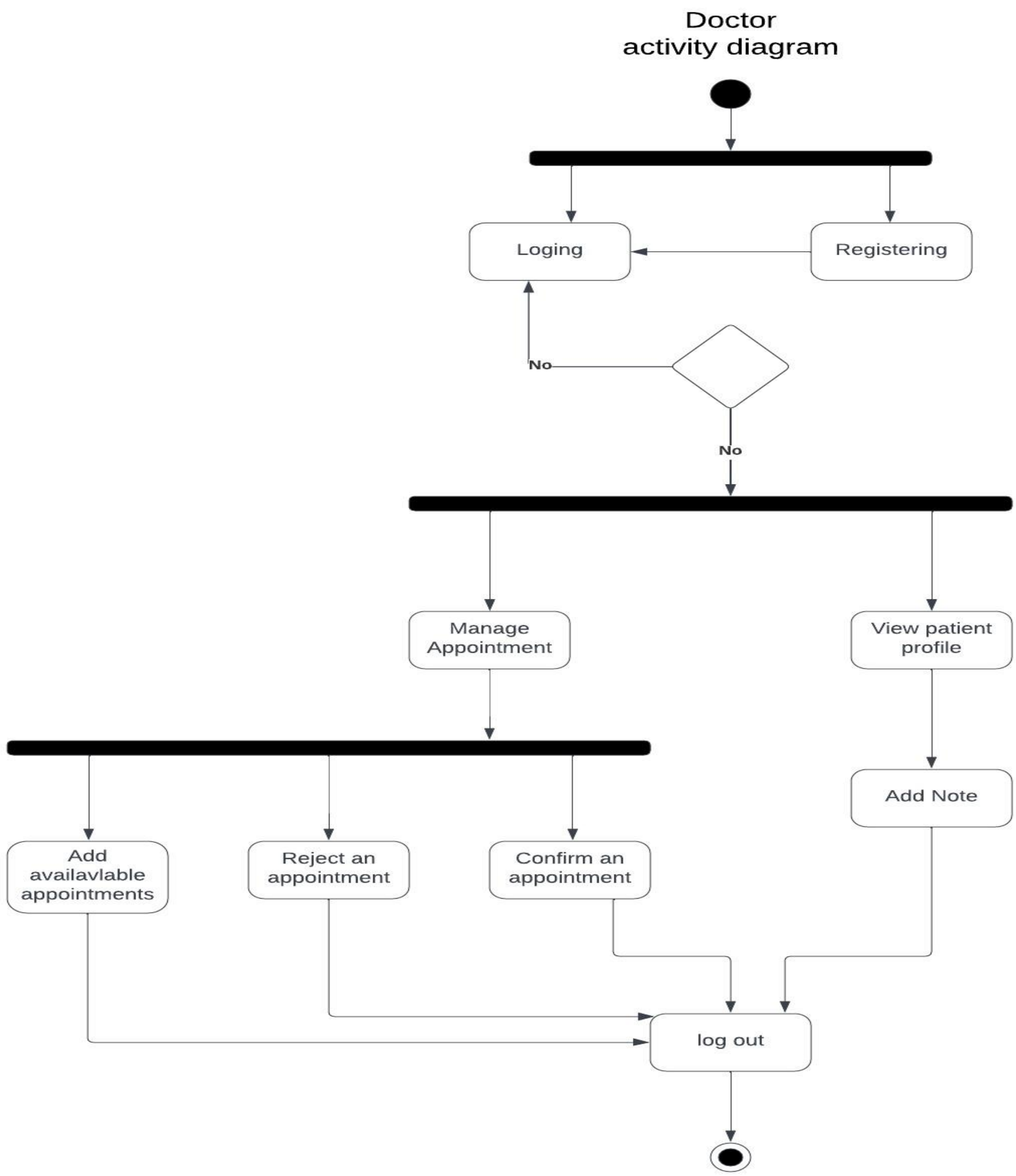


Figure 5.3 Doctor activity diagram

Patient
activity diagram

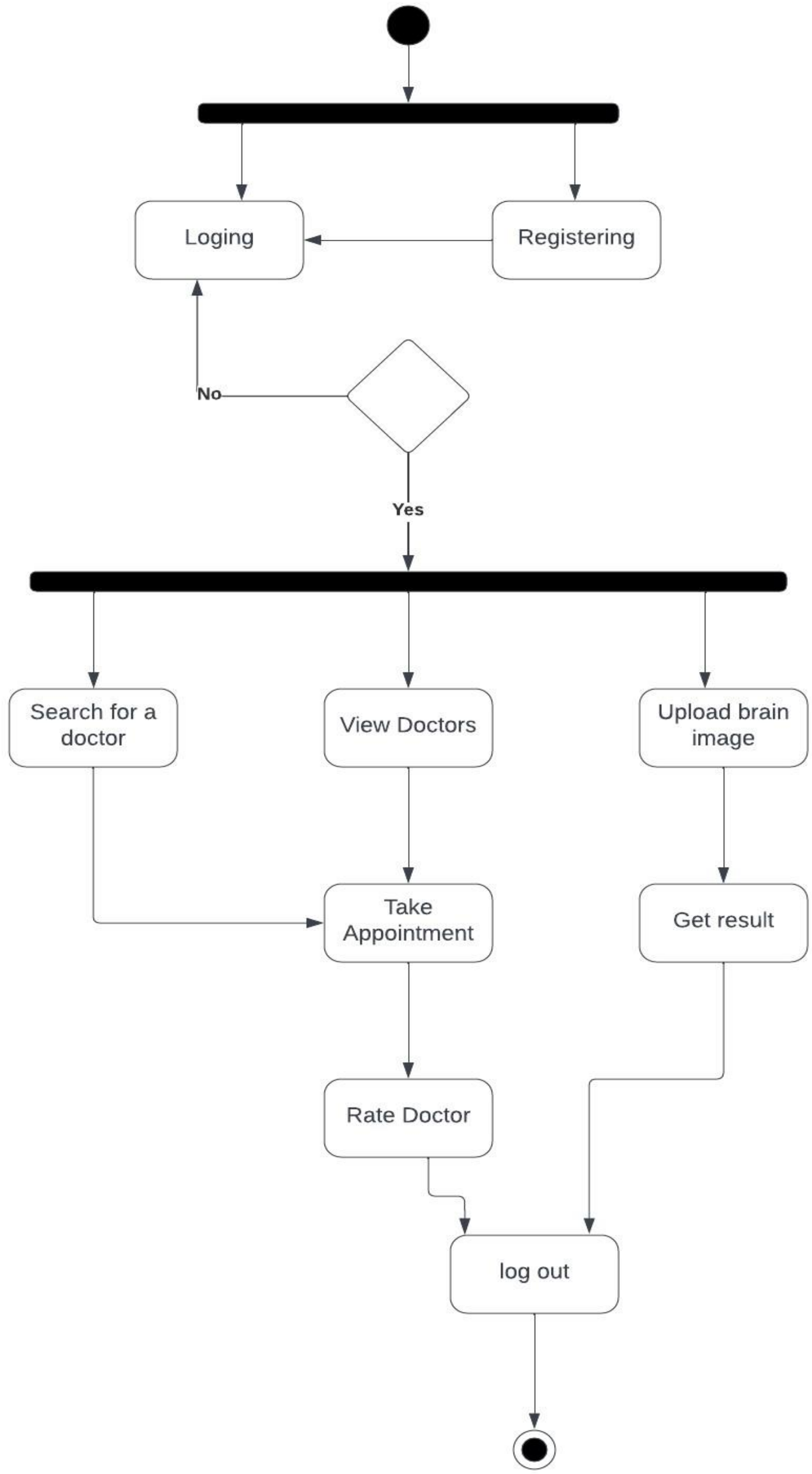


Figure 5.4 Patient activity diagram

CHAPTER SIX
SYSTEM DESIGN

6.1 Overview:

This chapter describes how the system works by discussing Sequence diagrams, class diagrams, and ERD. A sequence diagram shows, as parallel vertical lines (lifelines), different processes or objects that live simultaneously, and as horizontal arrows,

the messages exchanged between them, in the order in which they occur, Class Diagram is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects, Context Diagram which presents the sub-systems of our system and its data flow processing, and system architecture is the conceptual model that defines the structure, behavior, and more views of a system.

6.2 System Sequence Diagram

A system sequence diagram (SSD) is a sequence diagram that shows, for a particular scenario of a use case, the events that external actors generate their order, and possible inter-system events.

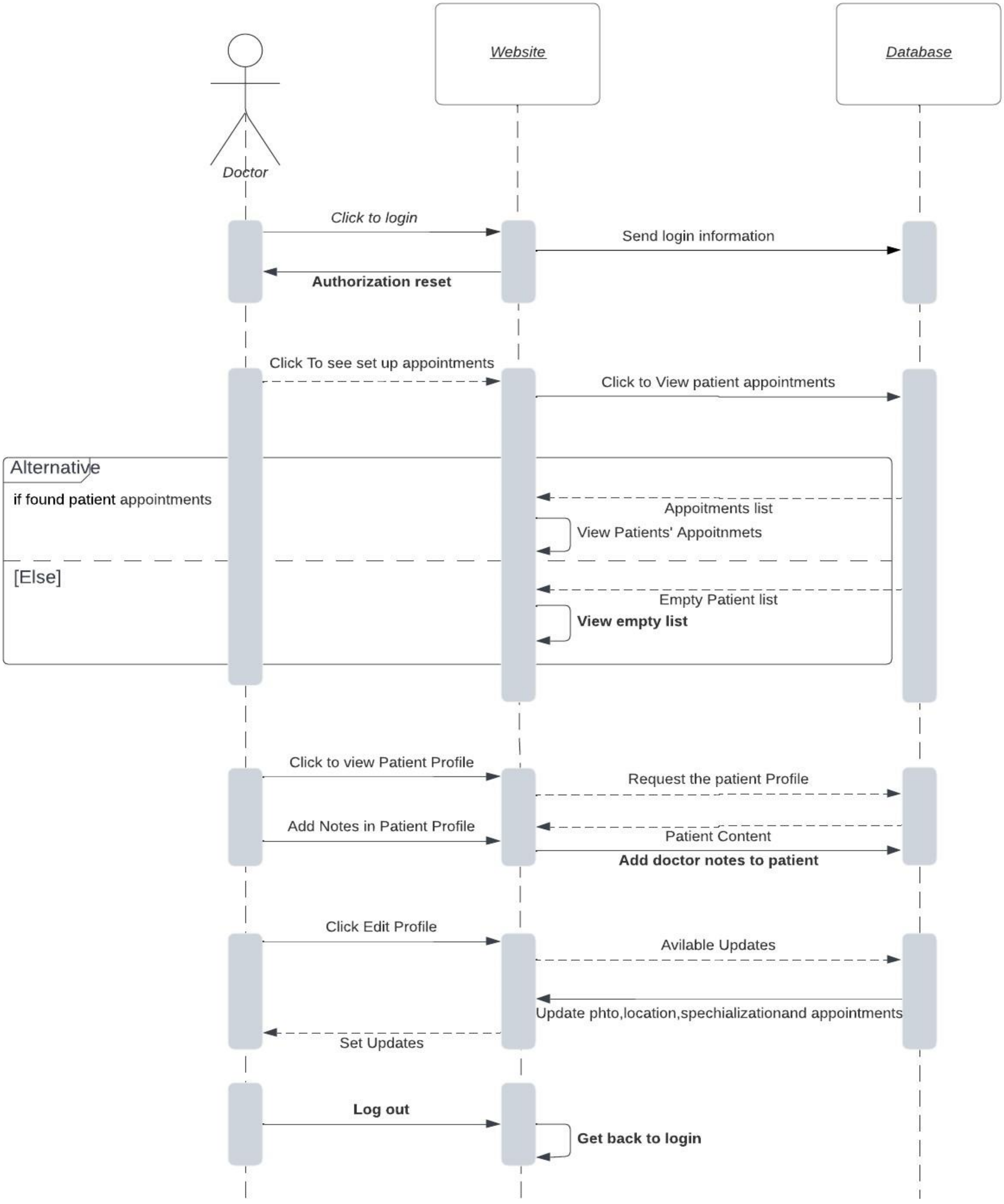


Figure 6.1 Doctor sequence diagram

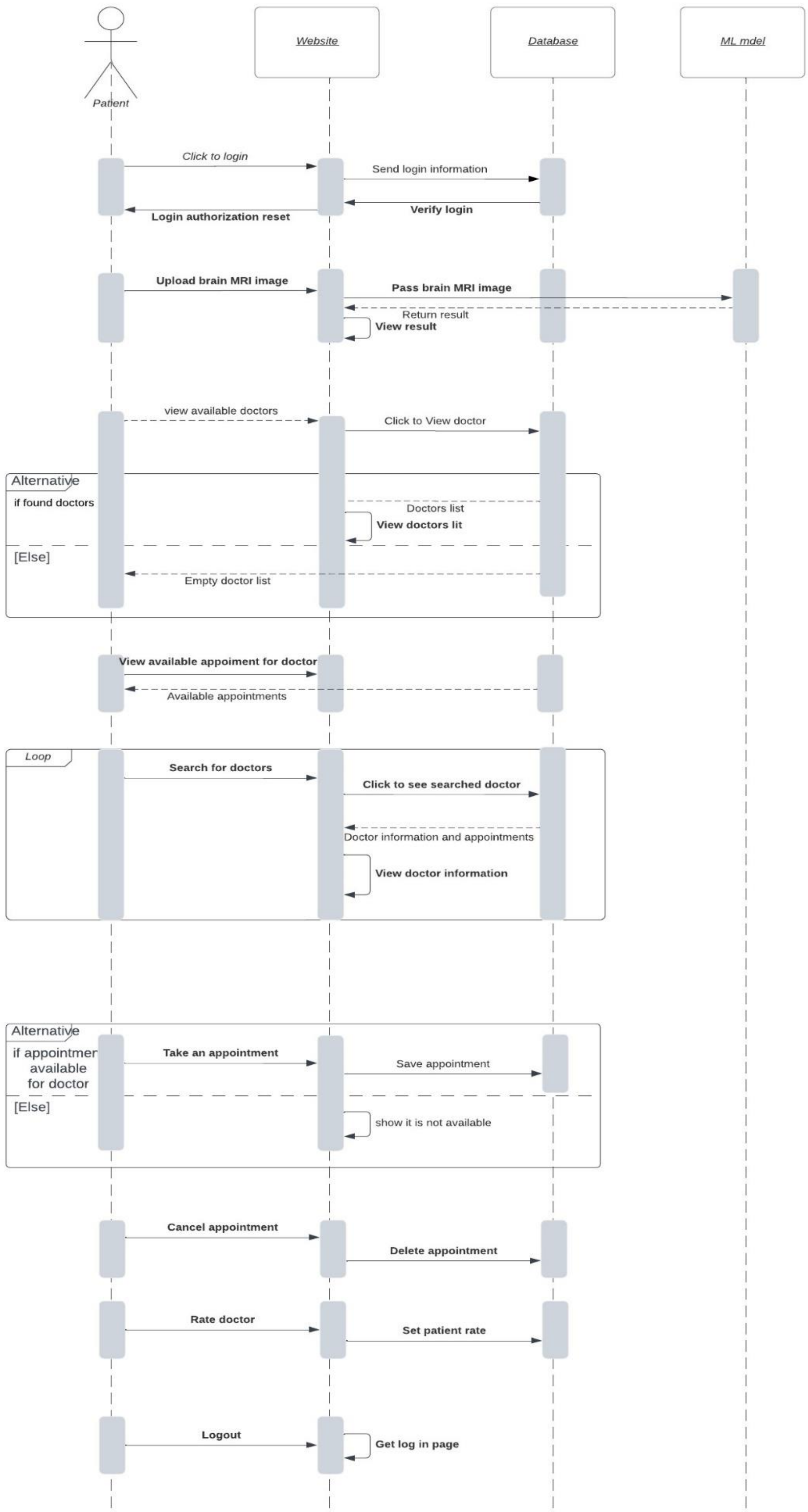


Figure 6.2 Patient sequence diagram

6.3 Sequence Diagram:

A sequence diagram shows object interactions arranged in a time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario.

6.3.1 Patient

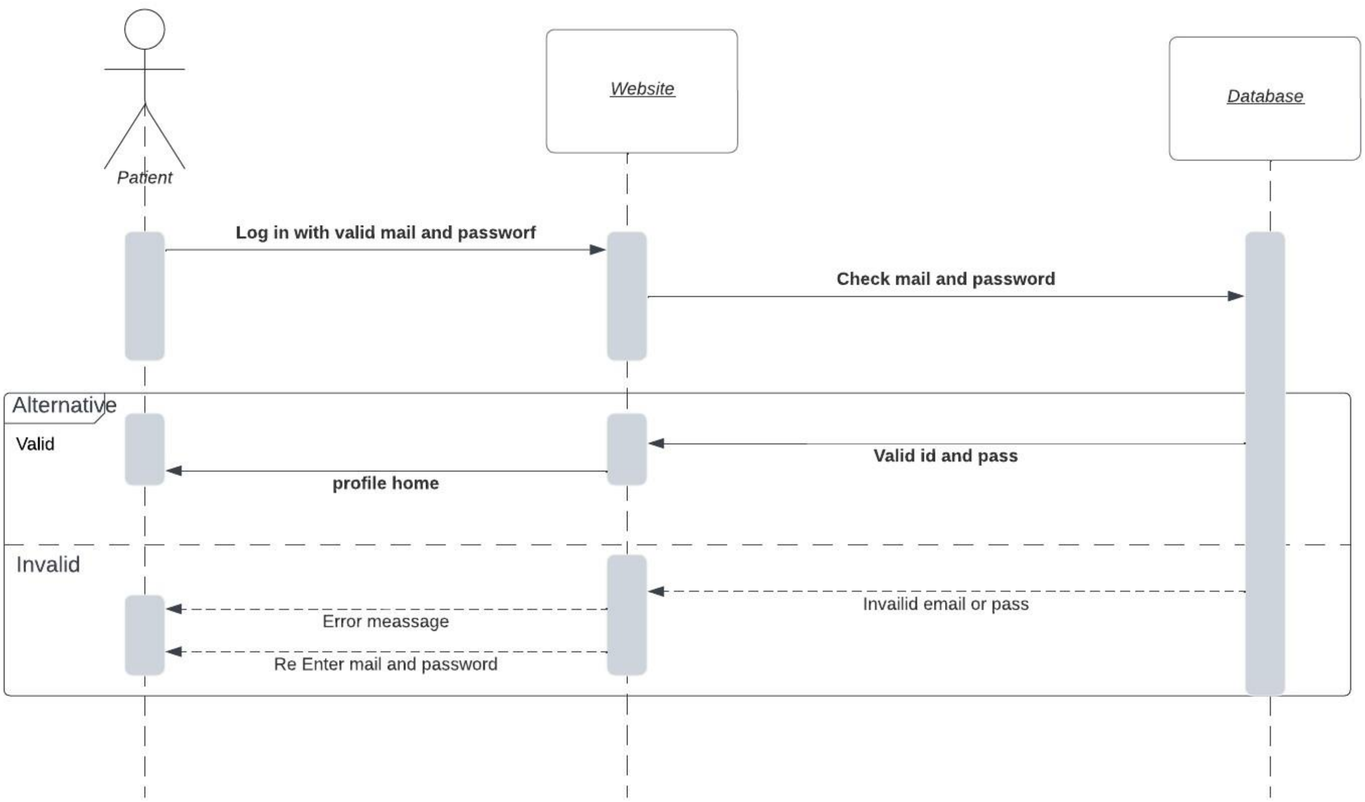


Figure 6.3 Patient login sequence diagram

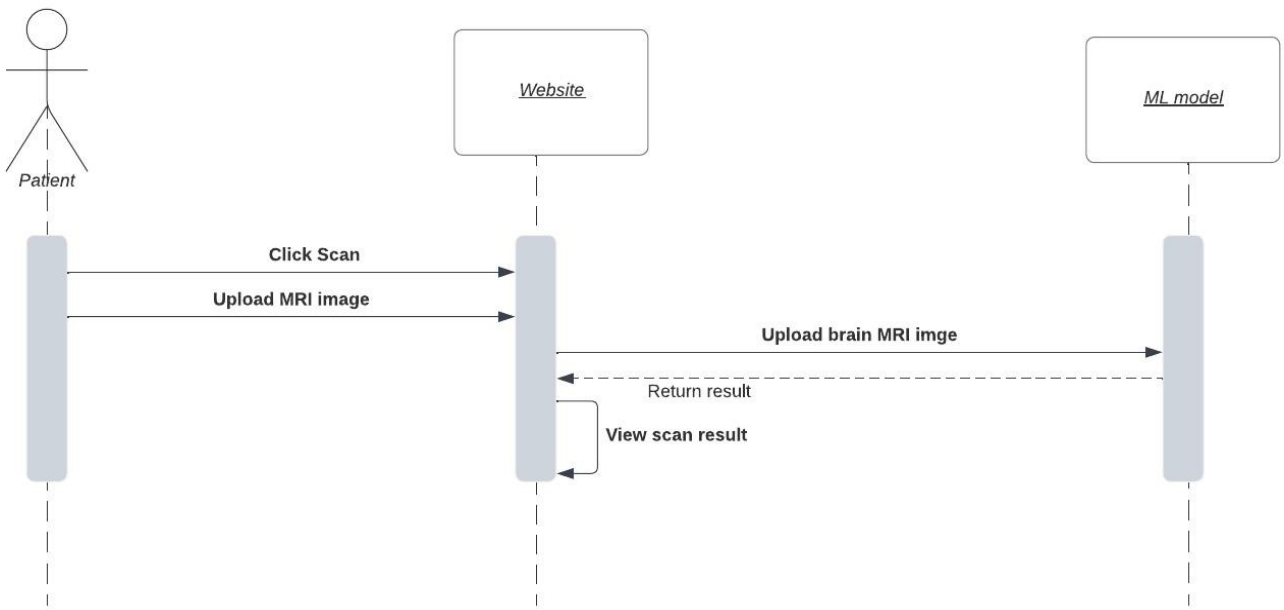


Figure 6.4 Patient scans MRI sequence diagram

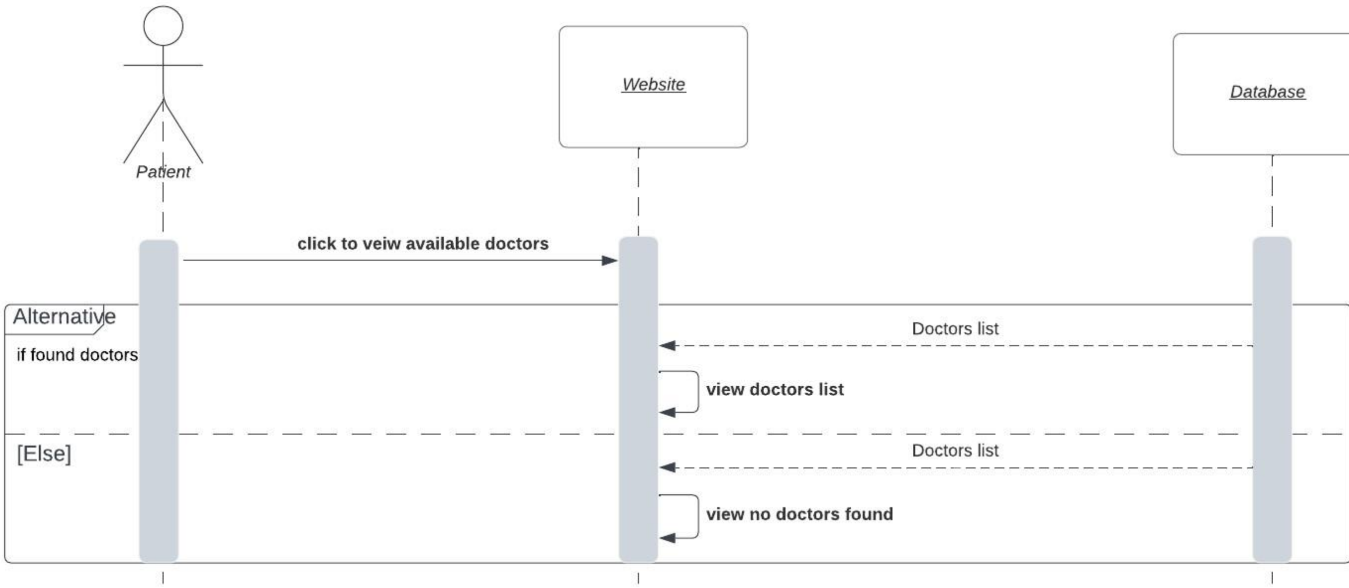


Figure 6.5 Patient view available doctors sequence diagram

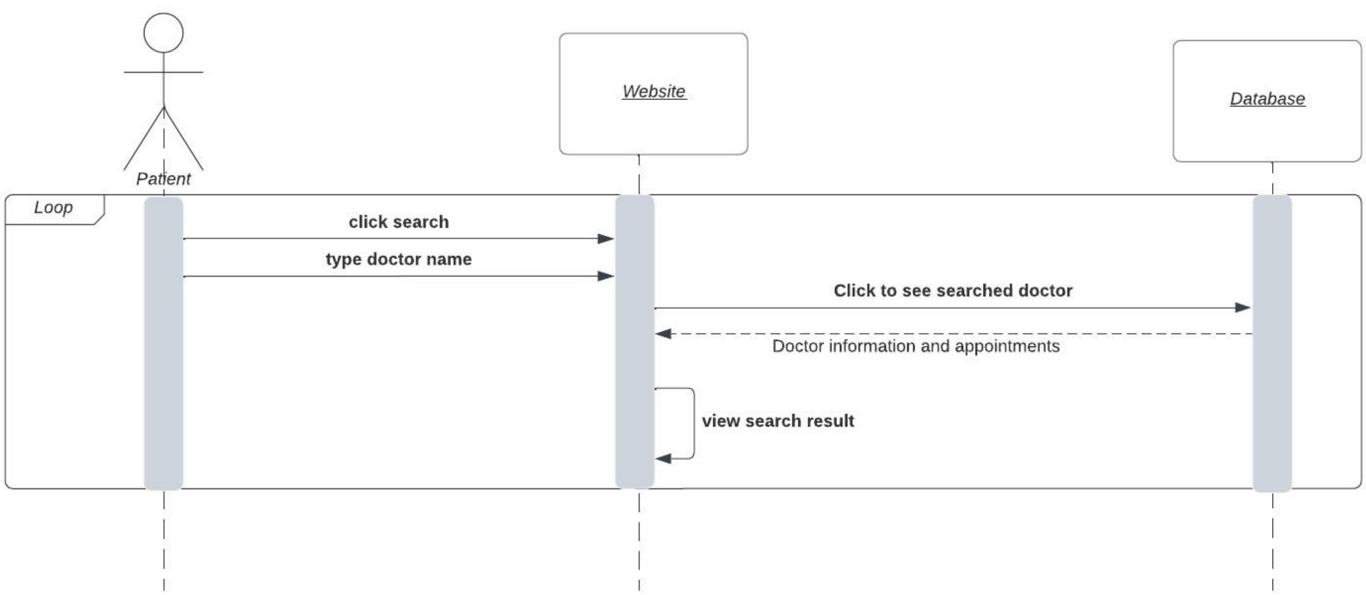


Figure 6.6 Patient search for a doctor sequence diagram

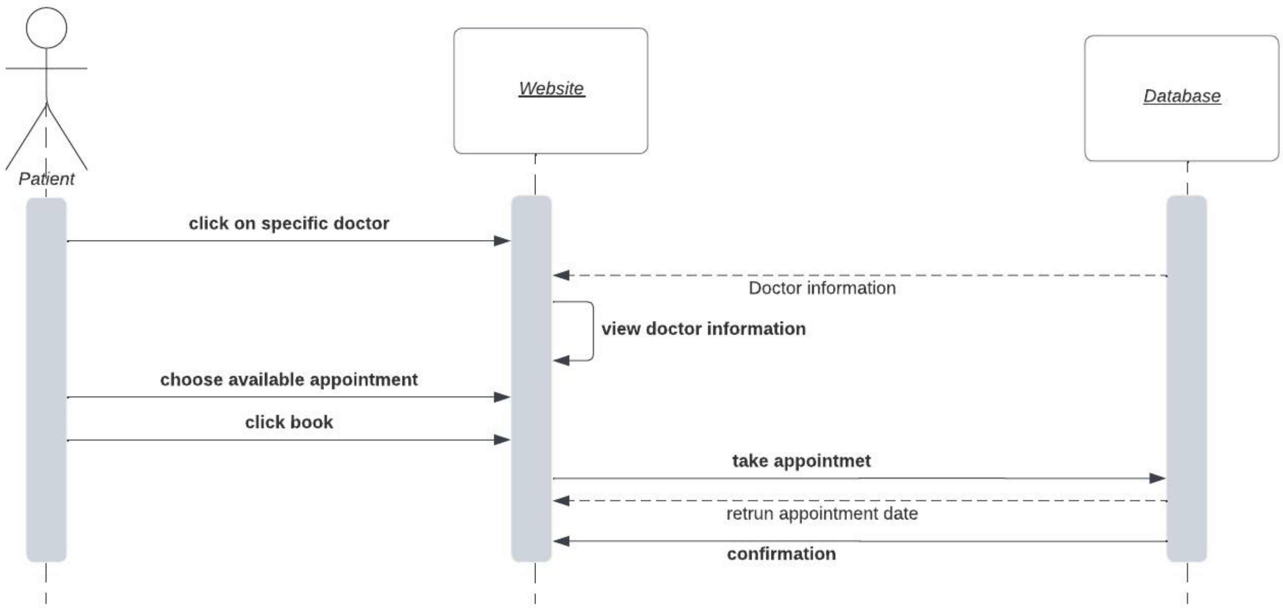


Figure 6.7 Patient takes an appointment sequence diagram

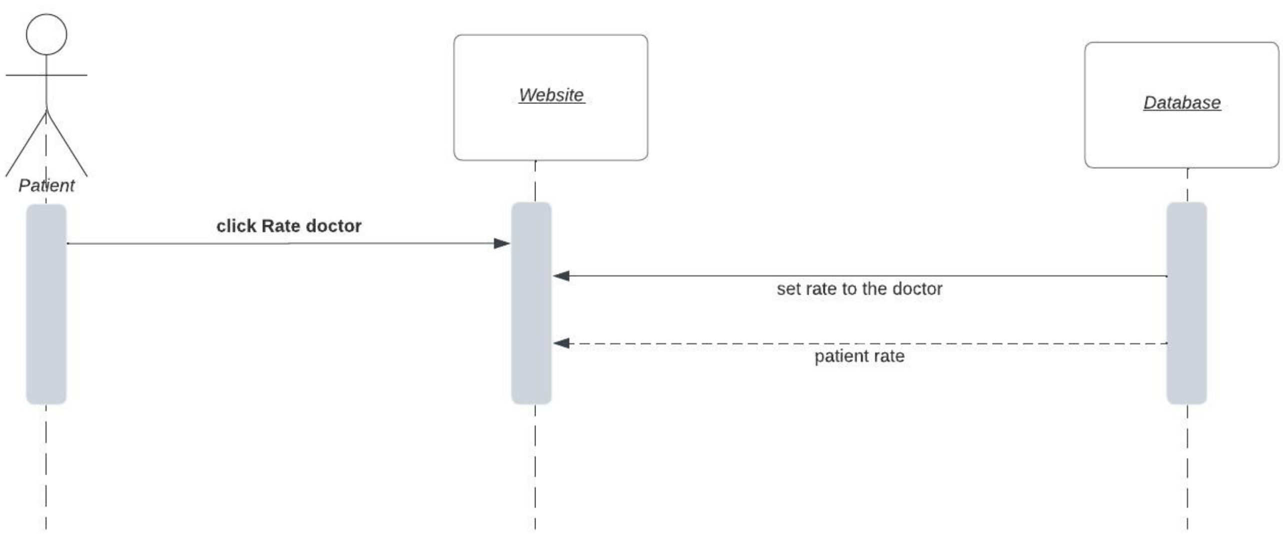


Figure 6.8 Patient rates his doctor sequence diagram

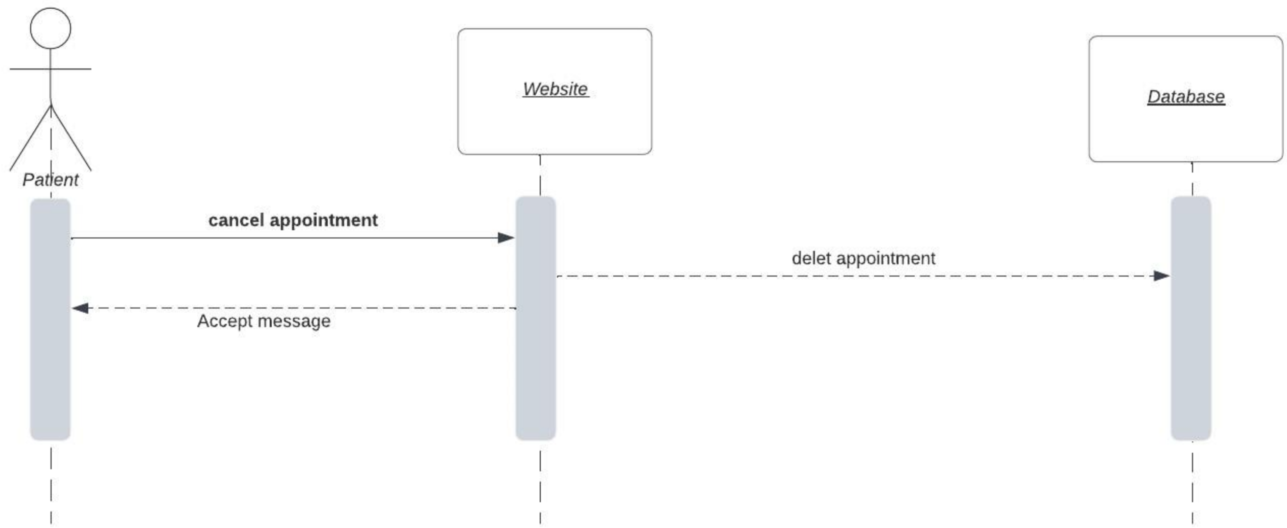


Figure 6.9 Patient cancels his appointment sequence diagram

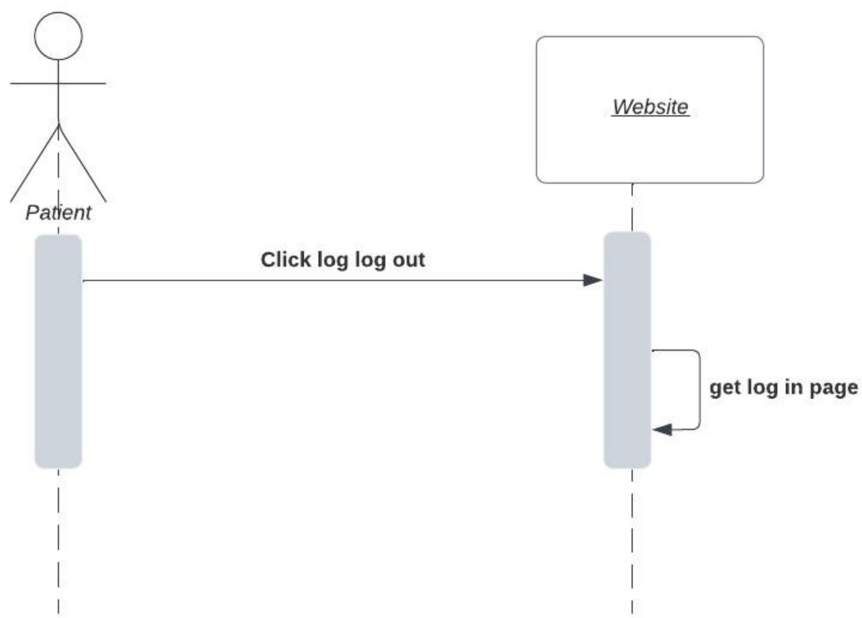
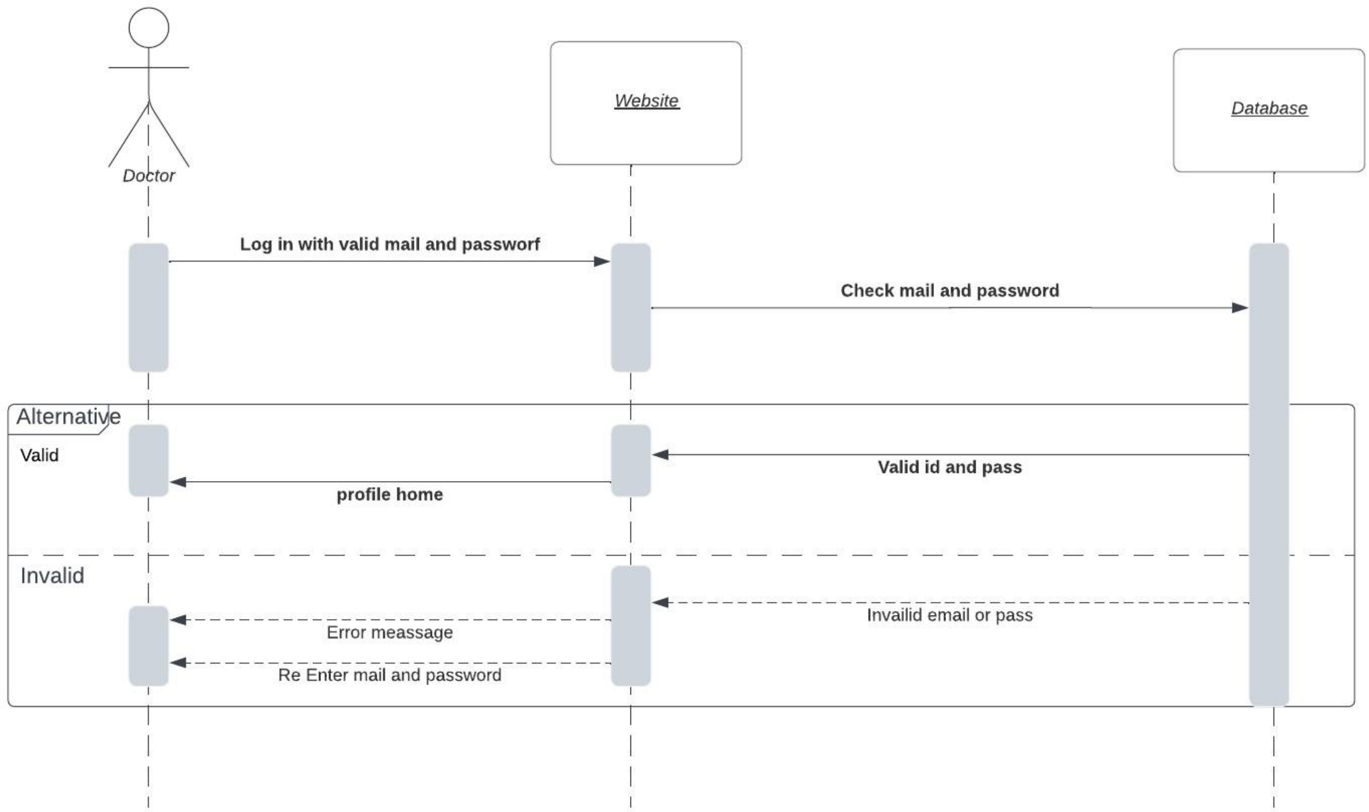


Figure 6.10: Patient logout sequence diagram

6.3.2 Doctor:

Figure 6.11 Doctor login sequence diagram



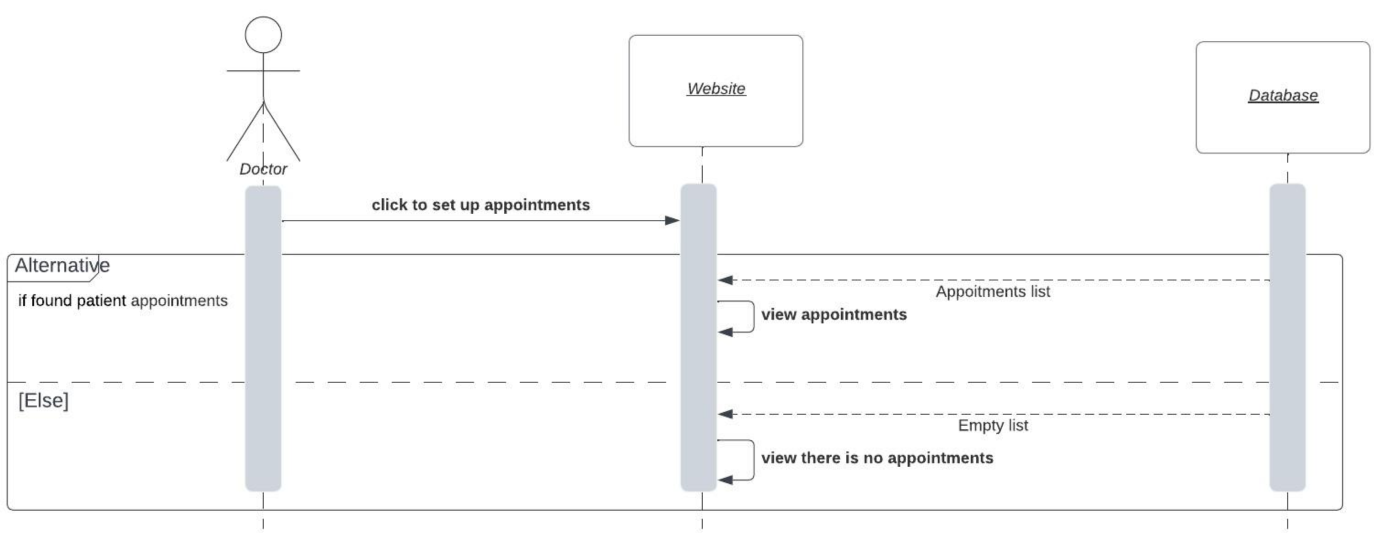


Figure 6.12 Doctor view appointment sequence diagram

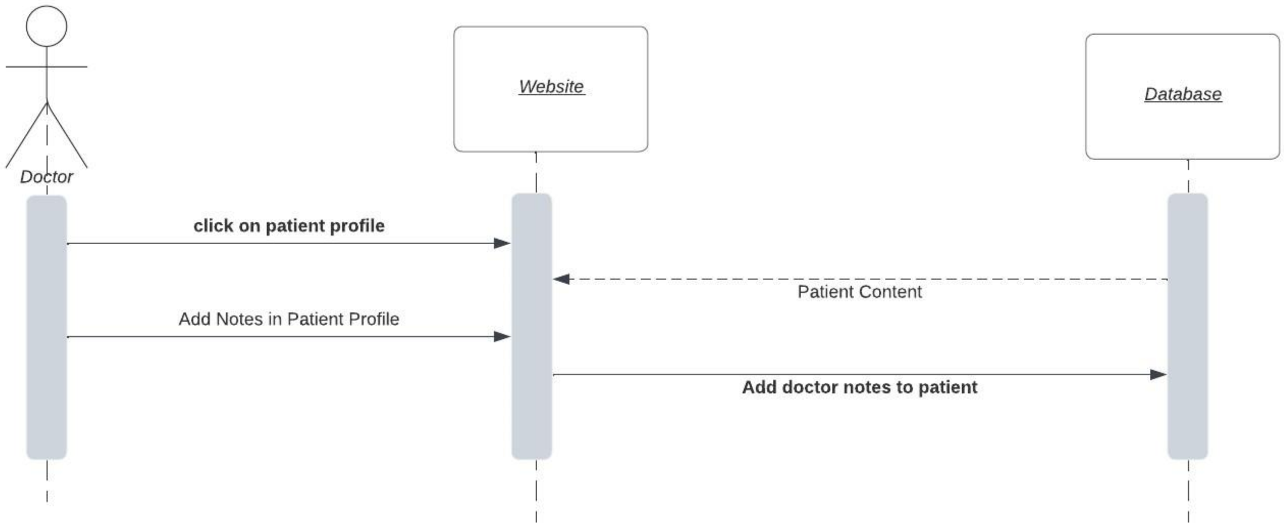


Figure 6.13 Doctor view patient profile sequence diagram

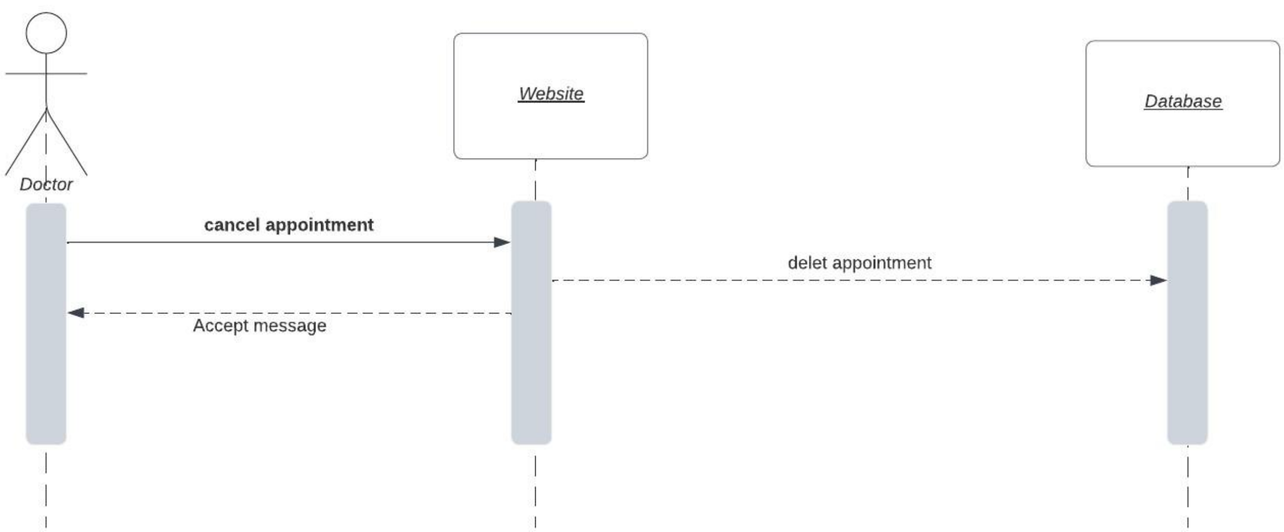


Figure 6.14 Doctor cancels appointment sequence diagram

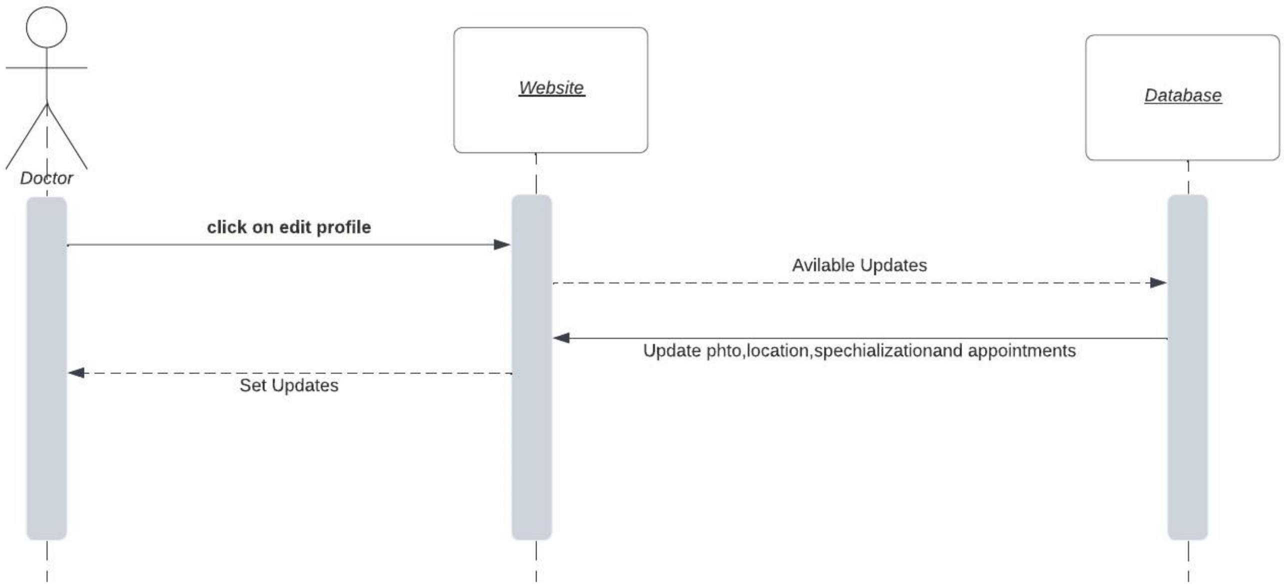


Figure 6.15 Doctor edits his profile sequence diagram

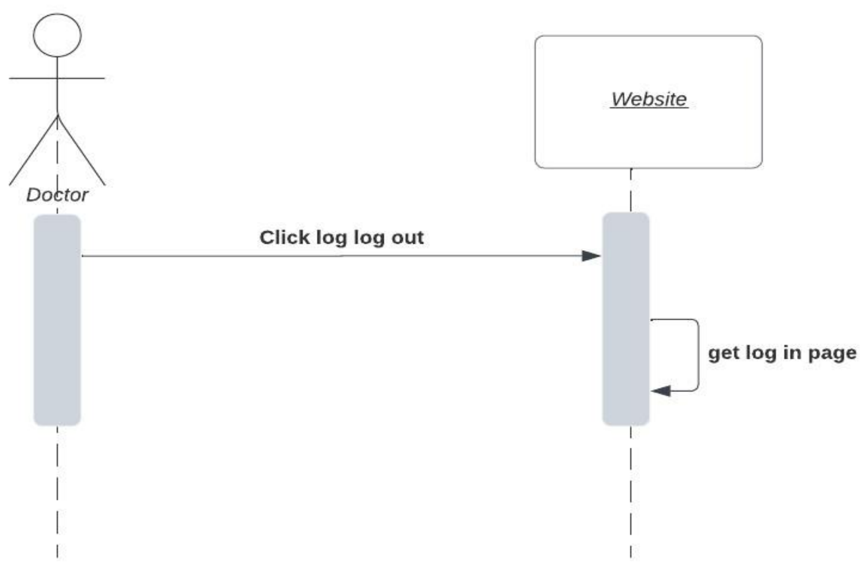


Figure 6.16 Doctor logout sequence diagram

6.4 Class Diagram

A class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, attributes, operations (or methods), and relationships among objects.

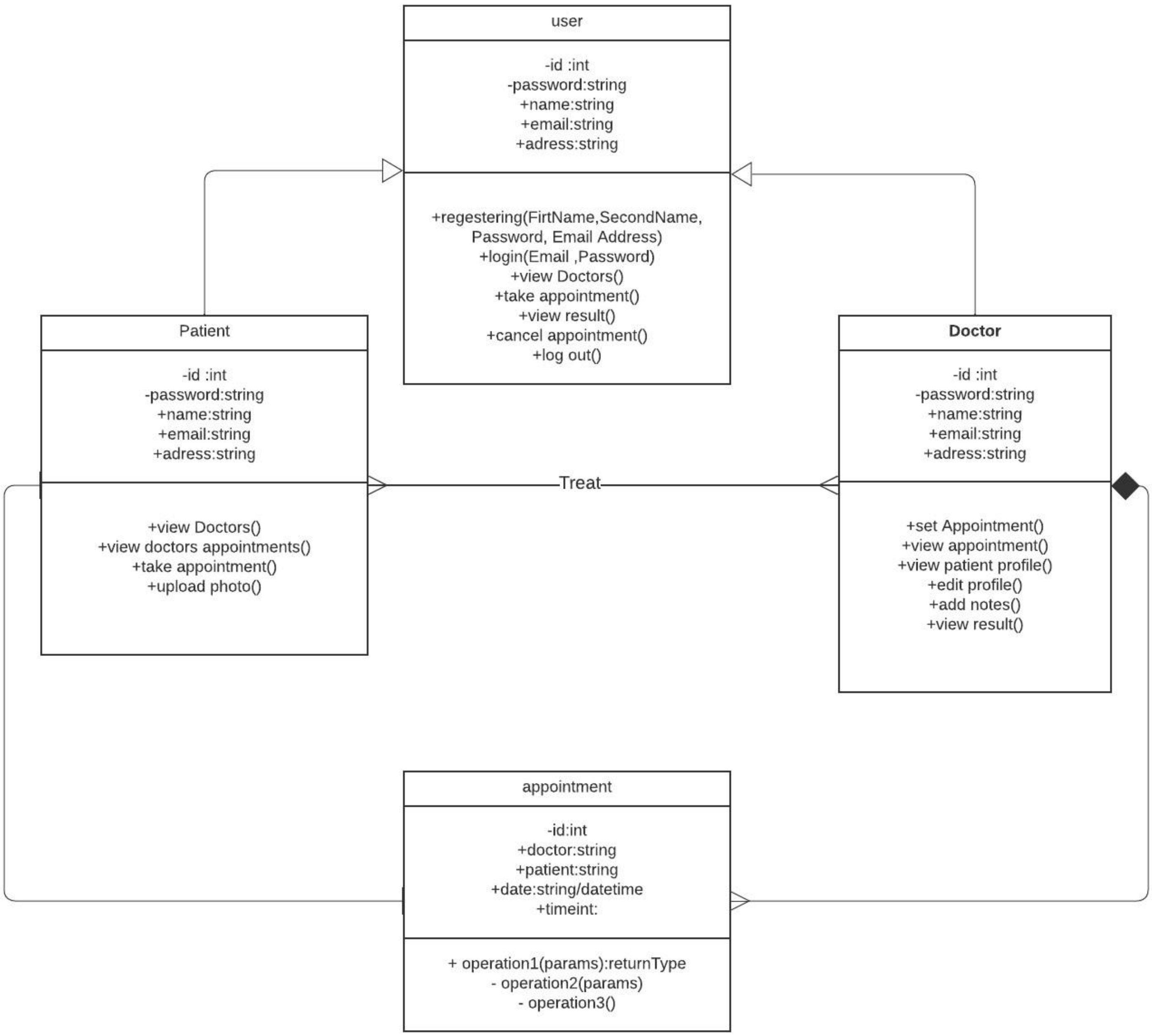


Figure 6.17 Class diagram

6.5 Data Modeling

6.5.1 Entity Relationship Diagram (ERD):

An entity-relationship model (ERM) is an abstract and conceptual representation of data. Entity-relationship modeling is a database modeling method, used to produce a type of conceptual schema or semantic data model of a system, often a relational database, and its requirements in a top-down fashion.

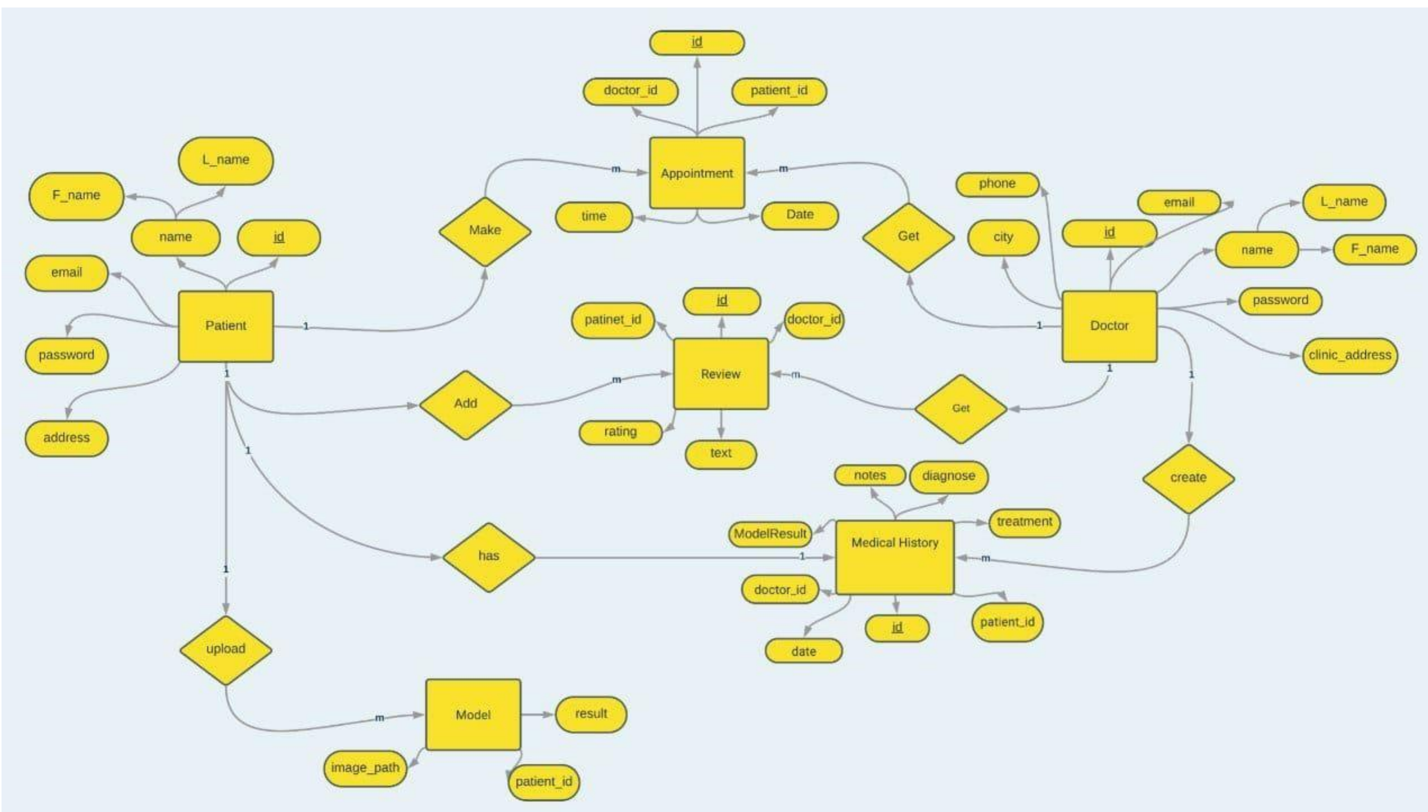


Figure 6.18 ERD

CHAPTER SEVEN
PROPOSED MODEL

7.1 Overview

We discuss the need for automated diagnosis of brain tumors due to the increasing number of cases. The manual diagnostic process is prone to errors and can lead to dangerous situations. We propose a deep neural network model that uses ResNet-50 and global averaging to address fading gradient and overfitting problems. The model was evaluated using a brain MRI dataset of three tumors consisting of 5732 images. Key performance measures are used to analyze the performance of the proposed model and its competitive models. The proposed model achieves a classification accuracy of up to 99.1% without increasing the data, and after increasing the data to 11450 by using other data, the model achieved a classification accuracy of up to 99.5%, outperforming the current models in terms of accuracy. The study has important implications for automating brain tumor diagnosis and could improve patient outcomes through faster and more accurate diagnoses. Our proposed model is superior to the existing models in classification accuracy. A chart is introduced below as the methodology used. [10*]

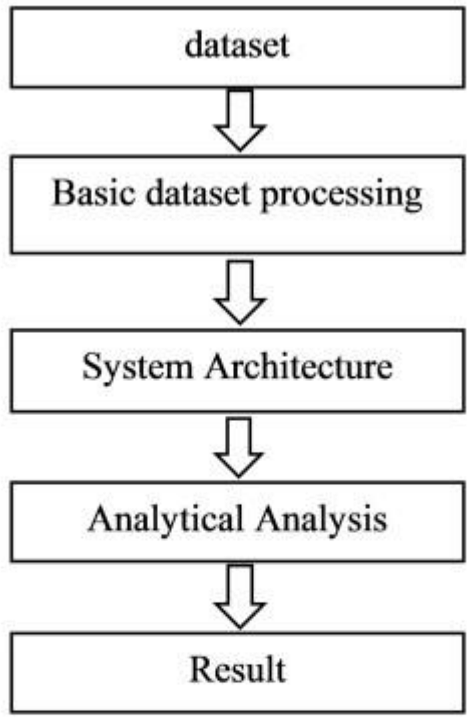


Figure 7.1 Model Architecture

7.2 Image Acquirement

Brain tumor dataset containing a total of 5732 MRI images is available at the American Brain Tumor Association [11*] : Brain Tumor Education website. The dataset consists of three types of brain tumors: meningioma (1642 slices), glioma (1652 slices), pituitary (1645 slices), and normal brain tissue (790 slices). The images are a combination of T1, T2, and Flair types of MRI, where T1 highlights fat tissue and appears brighter, T2 highlights fat tissue and water and appears brighter, and Flair is similar to T2 but with free-flowing water and fat appearing dark.

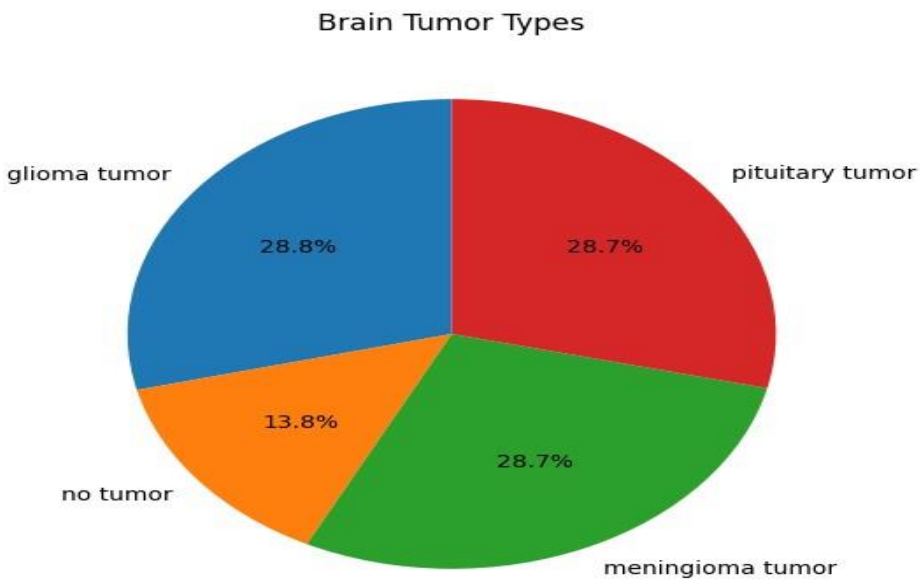


Figure 7.2 Brain Tumor Types

Brain tumor Dataset-2 containing a total of 5712 MRI images is available at Shenzhen University Association [11*]: Brain Tumor Education website. The dataset consists of three types of brain tumors: meningioma (slices 1339), glioma (1321 slices), pituitary (slices 1457), and normal brain tissue (slices 1595).

Merging Dataset-1 and Dataset-2 would result in a new dataset with a total of 11450 MRI images. The new dataset would have 2981 meningioma slices, 2973 glioma slices, 3102 pituitary slices, and 2385 normal brain tissue slices.

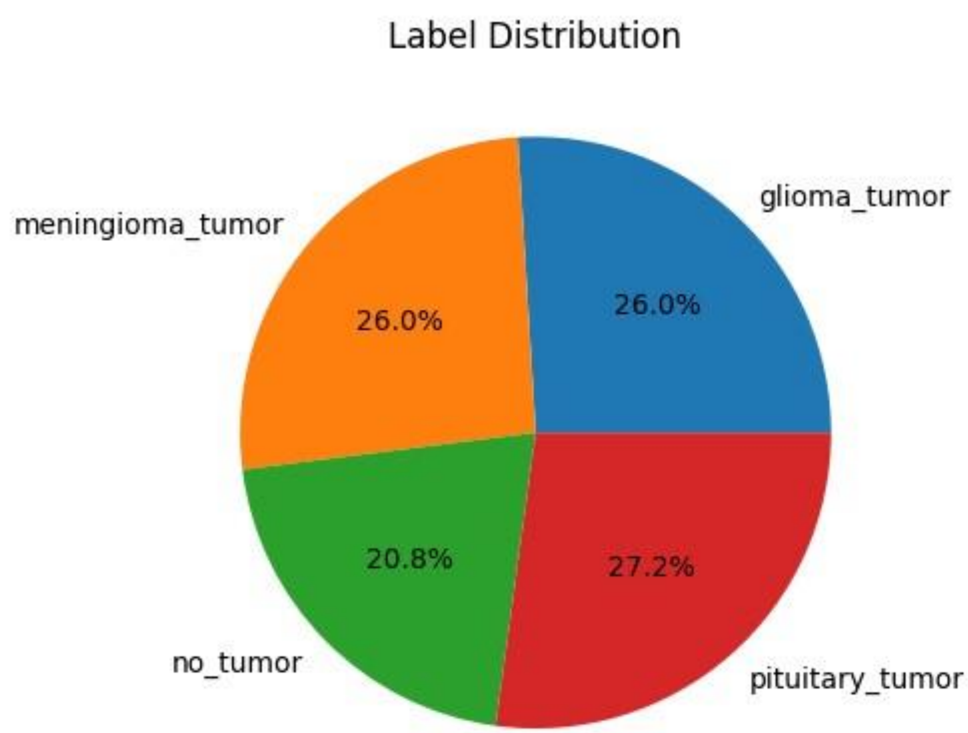


Figure 7.3 Label Distribution

7.3 Data processing

We describe the minimum data preprocessing steps that are performed on the datasets before training the brain tumor classification model.

These steps include:

Image resizing: All the images in the datasets are resized to 150×150 pixels to ensure consistency and reduce computational complexity during training.

Enhancing the image: The "enhance image" function is called on each image to enhance its brightness using the OpenCV "conversates" color transformation. This operation takes two arguments, "alpha" and "beta", which are set to 1.5 and 0 in this case, respectively. The "alpha" parameter controls the contrast of the image, while the "beta" parameter controls its brightness. This step is performed to improve the quality of the images and enhance the features that are important for classification.

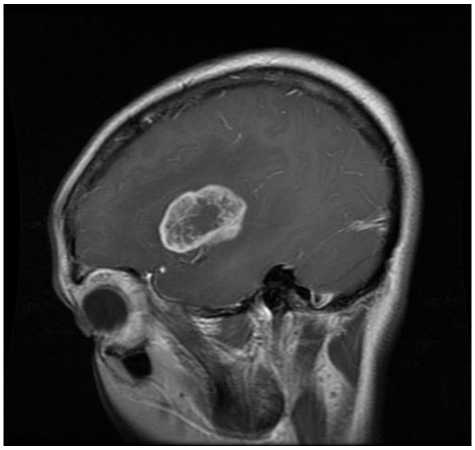


Figure 7.4 ***Glioma tumor***

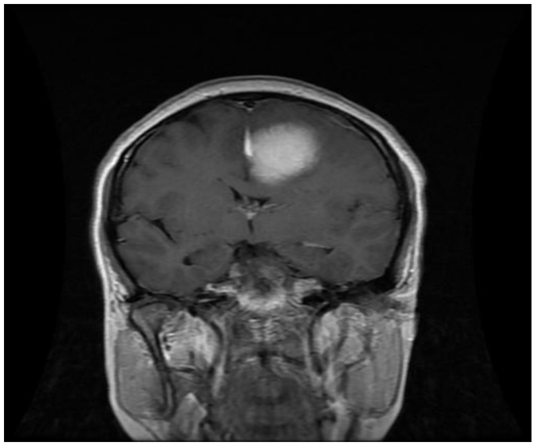


Figure 7.5 ***Meningioma tumor***

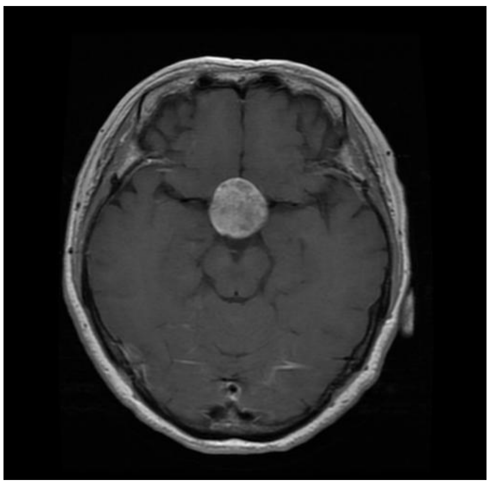


Figure 7.6 ***Pituitary tumor***

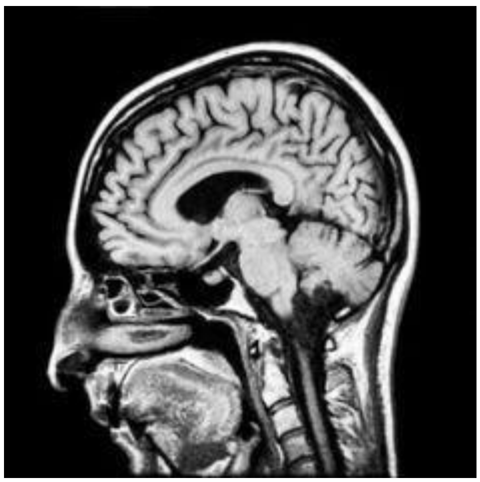


Figure 7.7 ***Normal Brain***

7.4 Feature Extractors

feature extractors are techniques used in deep learning to automatically extract relevant features from raw input data, such as images. Learned features are those that are learned directly from the input data by training a neural network with a labeled dataset. Convolutional Neural Networks (CNNs) are one example of a deep neural network that can be used for feature extraction.

The main idea behind the learned features approach is to discover data representations with multiple levels of abstraction to enable higher-level features of representing the semantics of the data, which provides better robustness to intraclass variability.

For feature extraction, pre-trained models can be used via transfer learning, which is a process where a pre-trained model that has been trained on a particular problem is used on a similar other problem. This has the advantage of taking less training time as the pre-trained model has already learned relevant features. In the case of image classification problems, many CNN models have been recognized through the ImageNet challenge, and those pre-trained models can be used via transfer learning in different image classification problems.

In this work, ResNet-50 [12*] is a pre-trained CNN model that has been modified and used for feature extraction. All the pre-trained models used in this work are applied to the datasets mentioned above.

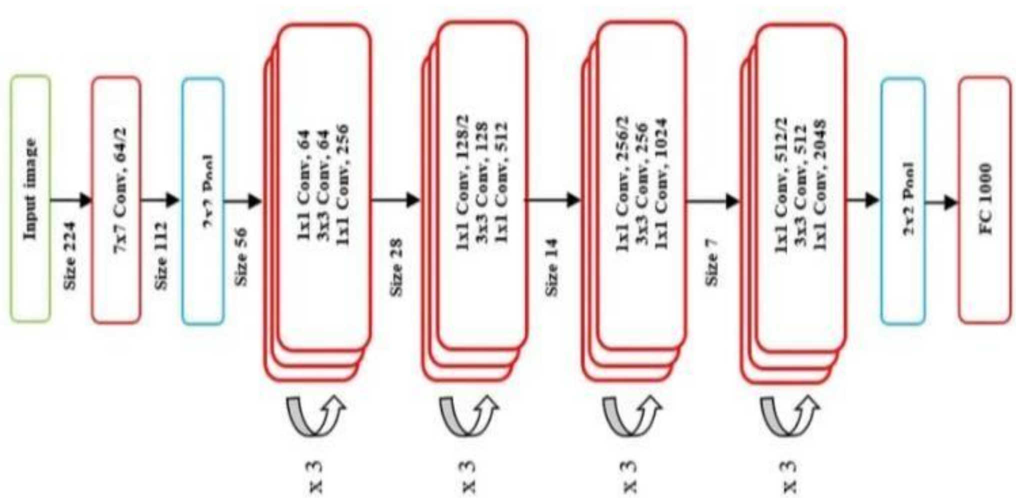


Figure 7.8 ResNet-50 architecture

7.5 Convolutional Neural Networks

Convolutional Neural Networks (CNNs) are a type of deep neural network commonly used for image classification tasks. CNNs consist of multiple layers, including convolutional layers, pooling layers, and fully connected layers.

The convolutional layer is the core component of a CNN, and it operates by convolving an input image with a set of learnable filters to extract features. These filters are also called kernels or receptive fields, and they are typically small matrices of weights that slide over the input image to compute a set of output feature maps. The filters are learned during training using backpropagation and gradient descent and they capture different local patterns and textures in the input image.

However, deep CNNs can suffer from the vanishing gradient problem, where the gradients of the loss function become very small as they propagate through many layers, making it difficult to update the weights of the early layers. To address this issue, Residual Networks (ResNets) were introduced, which use skip connections to allow the gradients to flow directly through the network. ResNets have achieved state-of-the-art performance on many image classification benchmarks, and ResNet50 is a popular variant of ResNets that balances performance and computational efficiency.

In summary, CNNs are a powerful type of deep neural network that can automatically learn useful features from images for classification tasks, and ResNets is a popular variant of CNNs that address the vanishing gradient problem and achieve state-of-the-art performance.

7.6 Pooling Layer

The pooling layer is another important aspect of CNNs, which performs a downsampling operation on the output feature maps of the convolutional layer. The main goal of the pooling layer is to reduce the size of the feature maps while retaining the most important information. This helps to reduce the computational cost and prevent overfitting during training.

There are several types of pooling operations, such as max pooling and average pooling. In max pooling, the maximum value in each sub-region of the feature map is selected and passed to the next layer. In average pooling, the average value of each sub-region is computed instead. From the survey, it has been found that maxpooling outperforms average pooling in image classification tasks.

7.6.1 Activation Layer

A key component of CNNs is that applies a non-linear transformation to the output of the convolutional layer. The purpose of the activation layer is to introduce nonlinearity into the model, which enables CNNs to learn complex representations of the input data and avoid learning trivial linear combinations of inputs.

Several types of activation functions can be used, such as sigmoid, hyperbolic tangent (tanh), and Rectified Linear Unit (ReLU). The ReLU function has become the most commonly used activation function in CNNs due to its simplicity and effectiveness. ReLU is a non-linear function that sets all negative values to zero and leaves positive values unchanged. The output of ReLU is expressed as

$$R(z) = \max(0, z), \text{ where } z \text{ is the input value.}$$

In the proposed model, the second layer represents the non-linearity layer, and the

ReLU activation function is used to introduce non-linearity into the model. The ReLU function is simple to compute and has been shown to improve the model's accuracy and efficiency.

In summary, the activation layer is a critical component of CNNs that applies a nonlinear transformation to the output of the convolutional layer. The ReLU function is a popular choice for the activation function due to its simplicity and effectiveness in improving the model's accuracy and computational efficiency.

7.6.2 SoftMax:

a commonly used layer in neural networks, especially in classification tasks. It is typically located at the end of the network and is responsible for producing a probability distribution over the possible classes for a given input.

The SoftMax function is used to convert the output of the previous layer into a probability distribution over the classes. It takes as input a vector of real numbers, often referred to as logits or activations, and applies the following formula to each element:

$$\mathbf{yk = exp(\varphi k) / (sum(exp(\varphi j))\ for\ all\ j)}$$

where yk is the output probability for class k, φk is the k-thon element of the input vector, and the denominator is the sum of the exponentials of all the elements in the input vector. This ensures that the output probabilities sum to 1.

The SoftMax layer is used in conjunction with a loss function, such as crossentropy, to train the network. During training, the network adjusts its weights to minimize the loss between the predicted probabilities and the true labels. Once the network is trained, the SoftMax layer can be used to classify new input data by selecting the class with the highest probability.

In summary, the SoftMax layer is an important component of a neural network used for classification tasks. It converts the output of the previous layer into a probability distribution over the possible classes using the SoftMax function. During training, it is used with a loss function to adjust the network's weights, and during inference, it is used to classify new input data.

7.7 The Used CNN Deep Learning Model [13*]

ResNet was developed to address several pitfalls of deep neural networks, including difficulty in training due to the increasing number of layers and parameters, and the vanishing/exploding gradient problem. As the number of layers increases, the number of parameters in the model increases exponentially, making it challenging to train deep neural networks with a large number of layers. Additionally, adding multiple layers can make the network more expressive but less discriminative, resulting in decreased performance for deeper networks.

7.7.1 ResNet-50 [14*] :

The ResNet-50 architecture contains the following elements:

ConvB1 block: A combination of a 7x7 convolutional layer with 64 filters and a max-pooling layer.

ConvB2, ConvB3, ConvB4, and ConvB5 blocks: Each block consists of three convolutional layers with 1x1, 3x3, and 1x1 kernel sizes, respectively, and a varying number of filters. The blocks differ in the number of filters used in each convolutional layer. ConvB2 uses (64, 64, 256) filters, ConvB3 uses (128, 128, 512) filters, ConvB4 uses (256, 256, 1024) filters, and ConvB5 uses (512, 512, 2048) filters.

Skip connections: Each convolutional block uses a skip connection to reduce the vanishing gradient problem.

Global average pooling layer: At the end of the network, there is a global average pooling layer that calculates the average value of each feature map and outputs a single value for each feature map.

SoftMax layer: The global average pooling layer is followed by a SoftMax layer for classification. In the proposed model mentioned in the passage, the output layer has been replaced with global average pooling and SoftMax layers for three-class classification.

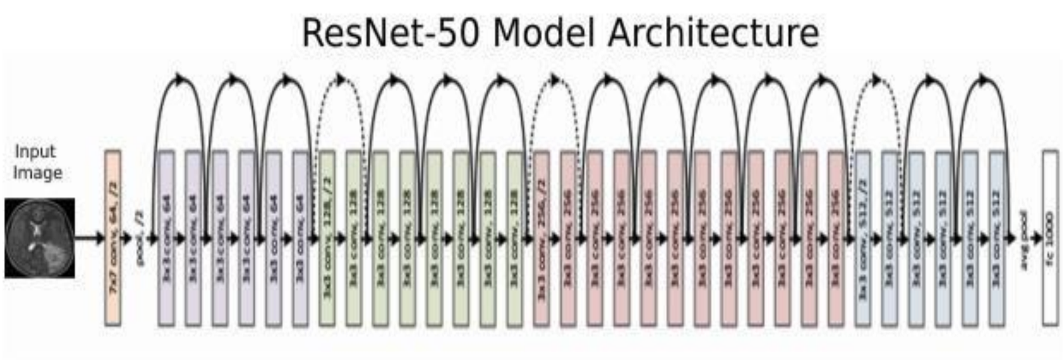


Figure 7.9 ***ResNet-50 model architecture***

7.7.2 skip connections:

In ResNet, skip connections are implemented by adding the identity mapping (x) to the output of a convolutional layer (F(x)), resulting in the final output y = x + F(x). This approach introduces a residual connection between the input and output of each layer, allowing the network to learn the residual mapping between the two, which can be seen as the difference between the input and output.

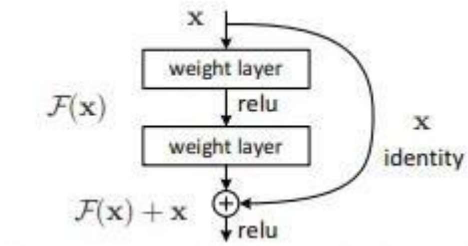


Figure 7.10 Skip Connection

By adding the residual connection, the network can learn to focus on the residual mapping, making it easier to optimize and train deeper networks without suffering from vanishing/exploding gradients.

Mathematically, the residual connection can be expressed as **y**

$$= \mathbf{H(x) + F(x)}$$

where H(x) represents the identity mapping of the input x. The goal of the network is to learn the mapping F(x) that transforms the input to the desired output, while the residual connection ensures that the input signal is preserved and can flow directly to the output.

7.7.3 Modified ResNet-50 [15*] :

The Residual Networks, in short ResNet 50 won the ImageNet challenge in 2015 and is being used in many computers vision-related tasks. The main idea here is to

train an extremely deep neural network that overcomes the vanishing gradients problem and also reduces the number of parameters to a great extent. It uses skip connections.

The original architecture uses the input size of 224 × 224 which was modified to 150× 150. Besides, at the top of the base model, replace fully connect with a global average pooling layer, dropout layer, and have been added that improved the model's performance.

Model hyper-parameters:

Finding the optimal values of the hyper-parameters is one of the crucial and significant tasks for building a robust model. Besides good features extractor and classifiers, the hyper-parameters values have a great influence on the fast convergence of the model. During the training of the proposed model, different values of the hyper-parameters were tried randomly and tested with all our datasets. We focused on activation function, optimizer, learning rate, dropout rate, batch size, number of epochs, train-test splitting ratio, etc. All the optimal values that have been chosen and used in the proposed model are shown in Table.

Hyper-parameter	Value
Output activation function	SoftMax
Optimizer	Adam
Initial learning rate	0.001
Learning rate decay	.3
Dropout rate	0.15
Early stopping	17,19
Batch size	32
No of epochs	200
Train-test split	80%–20%

Table 7.1

7.7.4 Transfer Learning [16*] :

with the ResNet-50 pre-trained model to perform a specific task.

Step-by-step transfer learning process:

1. **Obtain a pre-trained model**: The pre-trained model used in this code is ResNet-50, which has been trained on the ImageNet dataset.

2. **Create a base model**: The ResNet-50 model is used as the base model in this code.
3. **Freeze layers**: The initial layers of the ResNet-50 model are frozen to avoid re-learning the basic features that have already been learned. This is achieved by setting the "include top" argument to False when loading the model

4. **Add new trainable layers**: New trainable layers are added on top of the frozen layers to learn the specialized features required for the target task. In this code, a GlobalAveragePooling2D layer, a Dropout layer, and a Dense layer with a SoftMax activation function are added.

5. **Train the new layers**: The new layers are trained with the target task dataset. The input shape of the ResNet-50 model is set to (150,150), which means the images in the dataset have a size of 150x150.

6. **Fine-tune your model**:

Fine-tuning involves unfreezing some parts of the base model and training the entire model again on the whole dataset at a very low learning rate. This is done to adapt the pre-trained model to the target task and improve its performance. In this model, the first 133 layers of the ResNet-50 model are frozen, while the remaining layers are left trainable for fine-tuning. The low learning rate helps to prevent the pre-trained weights from being drastically changed while allowing the model to learn new patterns from the target task.

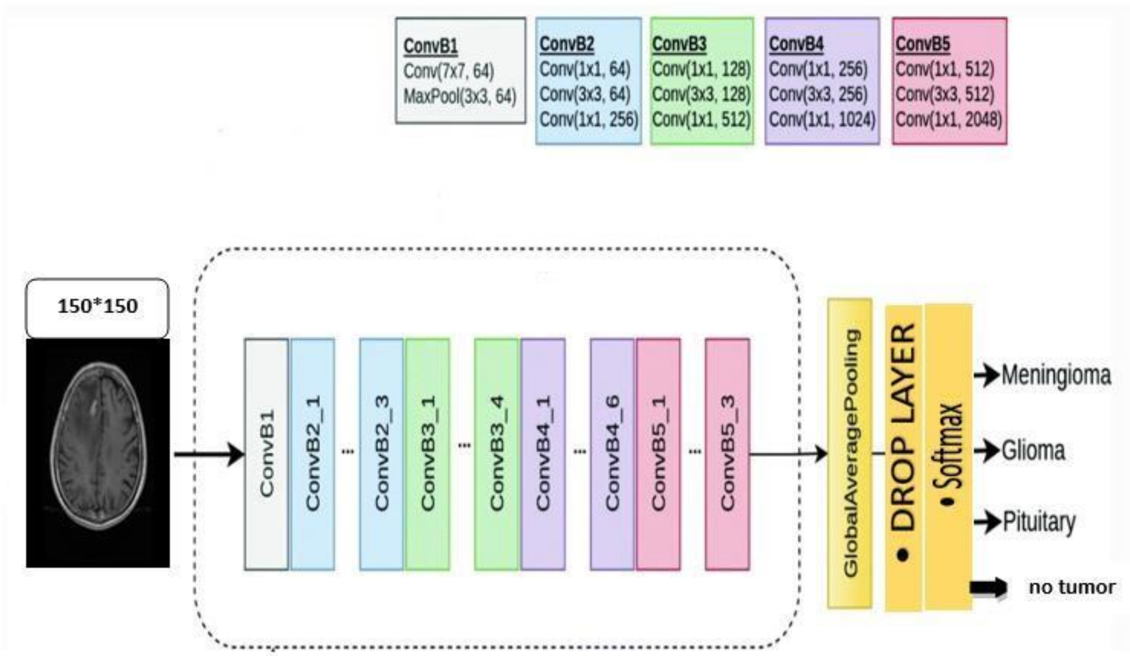


Figure 7.11 ***Resnet-50***

7.8 Experimental Results and Analysis

7.8.1 Materials:

This section presents and discusses all the details related to the experiments carried out to investigate and evaluate the performance of the proposed approaches. In this project, simulation experiments were performed on Google Collab with K80 GPU and 12 GB memory and 16 GB RAM, Intel Core i7-5610M CPU (3.00 GHz, 1600 MHz, 4 MB L3 Cache, 2 cores, 37W). The proposed approach is designed with TensorFlow, and Keras using Python.

7.8.2 Evaluation Metrics:

Performance metrics are commonly used in classification problems. Here are the definitions and formulas for each of them:

Accuracy: This metric measures the overall correctness of the model's predictions. It is calculated as the ratio of the number of correct predictions (true positives and true negatives) to the total number of observations.

Accuracy = (TP + TN) / (TP + TN + FP + FN)

Recall (Sensitivity): This metric measures the proportion of positive cases that were correctly identified by the model. It is calculated as the ratio of true positives to the sum of true positives and false negatives.

Recall = TP / (TP + FN)

Precision: This metric measures the proportion of correct positive predictions. It is calculated as the ratio of true positives to the sum of true positives and false positives.

Precision = TP / (TP + FP)

F1-score: is a commonly used metric to evaluate the performance of the classification model. It is the harmonic mean of precision and recall and provides a balance between the two metrics. The formula for F1-score is:

F1-score = 2 * (precision * recall) / (precision + recall)

TP, FP, TN, and FN terms are defined as follows:

True Positive (TP): This occurs when the model correctly predicts a positive example as positive (belonging to class X). For example, if an image is actually of class X and the model correctly classifies it as class X, this is a true positive.

False Positive (FP): This occurs when the model predicts a negative example as positive (belonging to class X) incorrectly. For example, if an image is actually of class Y and the model incorrectly classifies it as class X, this is a false positive.

True Negative (TN): This occurs when the model correctly predicts a negative example as negative (belonging to class Y). For example, if an image is actually of class Y and the model correctly classifies it as class Y, this is a true negative.

False Negative (FN): This occurs when the model predicts a positive example as negative (belonging to class Y) incorrectly. For example, if an image is actually of class X and the model incorrectly classifies it as class Y, this is a false negative.

7.10 Results of The System

The proposed approaches scenario:

7.10.1 First scenario:

Experimental results of the scenario using both fine-tuned pre-trained Resnet 50 models will be discussed in this section. The proposed Resnet -50 model is trained using hyper-parameters (batch size = 32, epoch = 19).

the training log for the neural network:

The accuracy of the model increased steadily from 87.65% in the first epoch to 100% in the eighth epoch. After that, the accuracy remained at 100% for the rest of the epochs. This suggests that the model has learned to fit the training data very well and can predict the correct class for each image in the training set.

The validation accuracy also increased steadily from 91.94% in the first epoch to 99.78% in the 19th epoch. This suggests that the model can generalize well to new data, as it can predict the correct class for the majority of the images in the validation set.

The learning rate was reduced using the ReduceLROnPlateau callback function when the validation loss did not improve for several epochs. This helped to prevent overfitting and allowed the model to continue to improve its performance on the validation set.

The training log suggests that the neural network was able to learn the features of the brain tumor dataset and achieve high accuracy and validity.

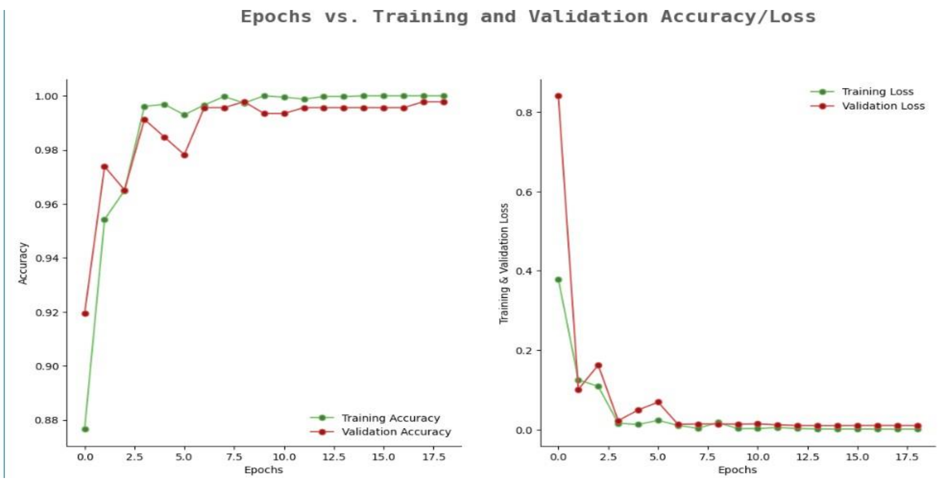


Figure 7.12 The learning rate

This scenario presents the classification of the brain tumor Model as accuracy = 99%. The results of applying resnet50 on the brain dataset are discussed in this section. The table shows the performance metrics of the model on the dataset of the first scenario.

class	precision	recall	f1-score
Glioma	.99	.99	.99
No tumor	.100	.100	.100
meningioma	.99	.98	.98
Pituitary	.99	.99	.99

Table 7.2 scenario presents the classification of the brain tumor Model

The results show that the model achieved high-performance metrics, with an accuracy of 0.99 and high precision, recall, and F1-score for each class. This indicates that the model was able to accurately classify the brain tumor images into their respective categories.

In the first scenario "brain tumor classification" was obtained in 5738 images (80% train and 20% test).

The confusion matrix figures provide a visual representation of the performance of the model. They show that the majority of the images were correctly classified, with only a small number of misclassifications. This suggests that the model is robust and can generalize well to new data.

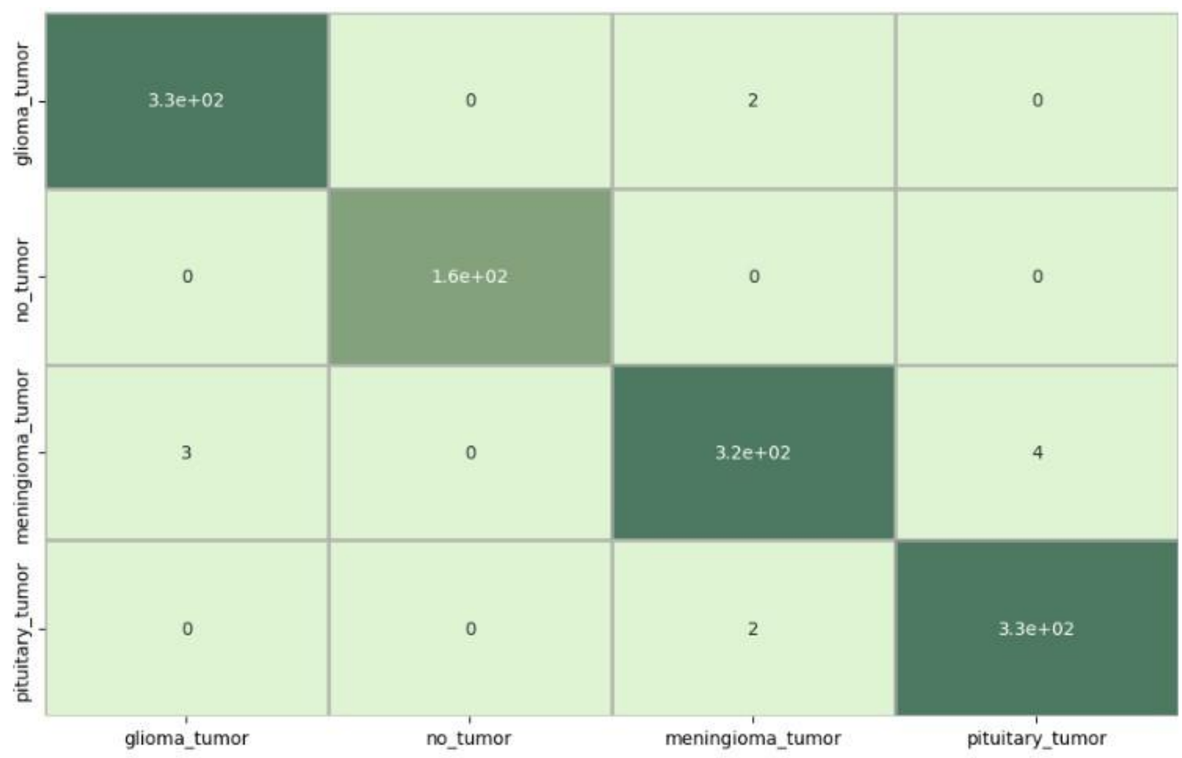


Figure 7.13 Confusion matrix

In addition to the performance metrics and confusion matrix, log loss is also an important evaluation metric for classification models.

The log loss measures the performance of the model in terms of the predicted probabilities of the classes. A lower log loss indicates better performance, as it means the model is more confident in its predictions.

The log loss equation is given by:

-1/N * sum (y_i * log(y_hat_i) + (1 - y_i) * log (1 - y_hat_i))

where:

N is the number of samples in the dataset y_i is the true label (0 or 1) of the i-th sample

y_hat_i is the predicted probability of the positive class (i.e., the probability that the sample belongs to class 1) for the i-th sample

The log loss measures the difference between the true labels and the predicted probabilities and penalizes the model for making confident incorrect predictions. A lower log loss indicates that the model is making more accurate and confident predictions.

In the scenario, the log loss value of 0.03 suggests that the model is performing very well in terms of predicting the probabilities of the classes and is making accurate and confident predictions with a low amount of error.

7.10.2 Second scenario:

Experimental results of the second scenario using both fine-tuned pre-trained Resnet 50 models will be discussed in this section. The proposed Resnet -50 model is trained using hyper-parameters (batch size = 32, epoch = 17). **the training log for the neural network:**

This is the training log for a deep-learning model with 17 epochs

The model starts with an initial learning rate of 0.001 and is trained for 3 epochs with high accuracy and low loss on both the training and validation sets. At the end of epoch 3, the ReduceLROnPlateau callback reduces the learning rate to 0.0003 as there is no improvement in the validation loss.

The model is then trained for another 3 epochs with the reduced learning rate, where the validation accuracy reaches 99.5% at epoch 6. At the end of epoch 8, the learning rate is reduced again to .0000945, and the model is trained for another 4 epochs.

At the end of epoch 12, the learning rate is further reduced to .0000275, and the model is trained for another 5 epochs. At the end of epoch 16, the learning rate is further reduced to .00000262.

Finally, the training is stopped early at epoch 17, which suggests that the model starts overfitting the training data after that epoch. Early stopping can prevent the model from memorizing the training data too well and improve its ability to generalize to new data.

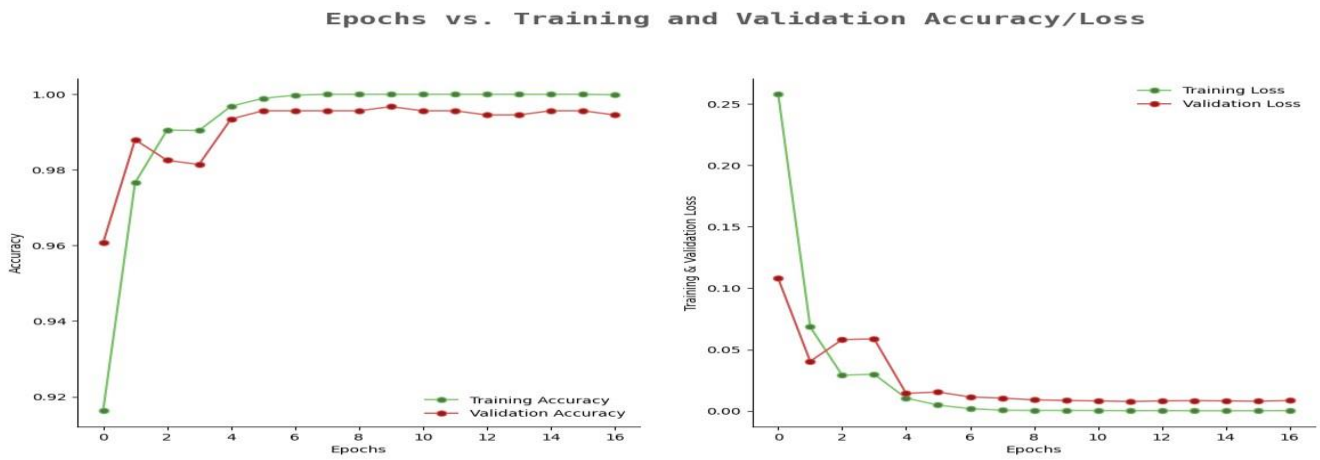


Figure 7.14 Epoch Vs. Training and Validation Accuracy/Loss

This scenario presents the classification of the brain tumor Model as accuracy = 99.5%. The results of applying resnet50 on the brain dataset are discussed in this section. The table shows the performance metrics of the model on the dataset of the second scenario.

class	precision	recall	f1-score
Glioma	.100	.99	.100
No tumor	.100	.100	.100
meningioma	.99	.100	.99
Pituitary	.100	.100	.100

Table 7.3

The second scenario "brain tumor classification" was obtained in 11450 images (80% train and 20% test).

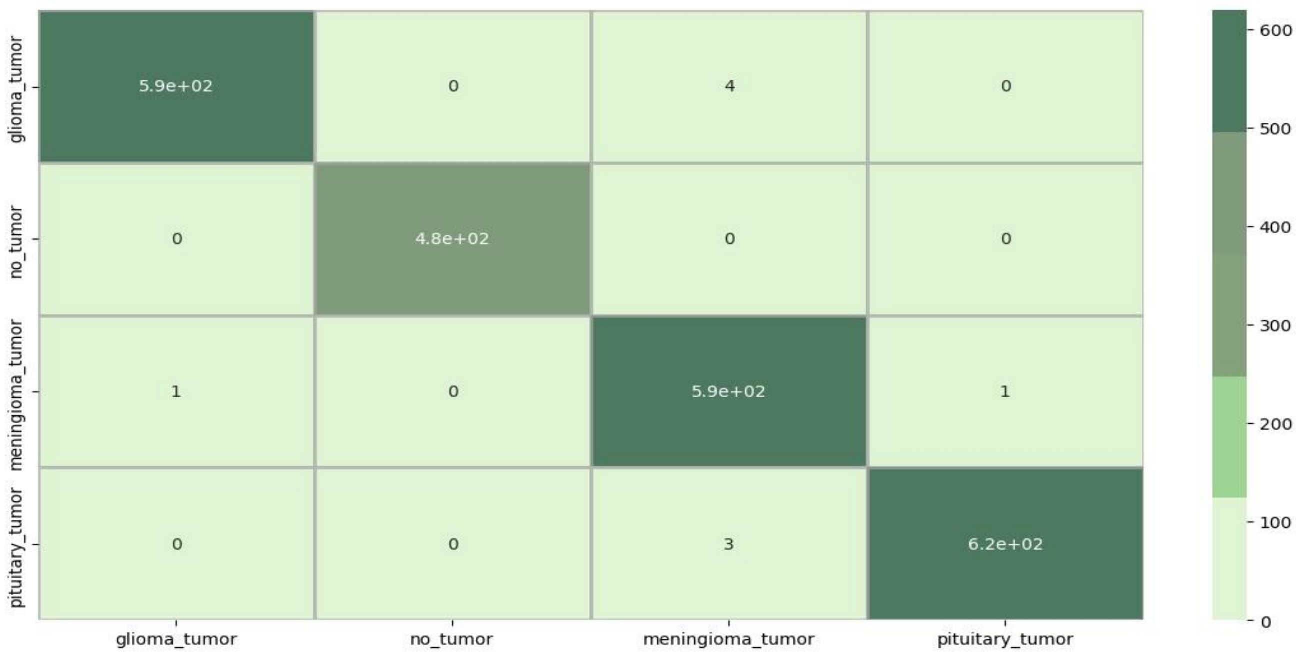


Figure 7.15 Confusion matrix

It is important to carefully evaluate model performance and use a range of techniques to improve its performance. Therefore, data containing 1311 images of brain tumors were used to test the model.

The model was tested on a dataset of 1311 brain tumor images and achieved an overall accuracy of 0.98. The precision, recall, and F1 score for each class are also provided.

class	precision	recall	f1-score
Glioma	**.99**	**.97**	**.98**
No tumor	**.98**	**.95**	**.97**
meningioma	**.99**	**.99**	**.99**
Pituitary	**.94**	**.100**	**.97**

Table 7.4

The confusion matrix figures provide a visual representation of the performance test data:

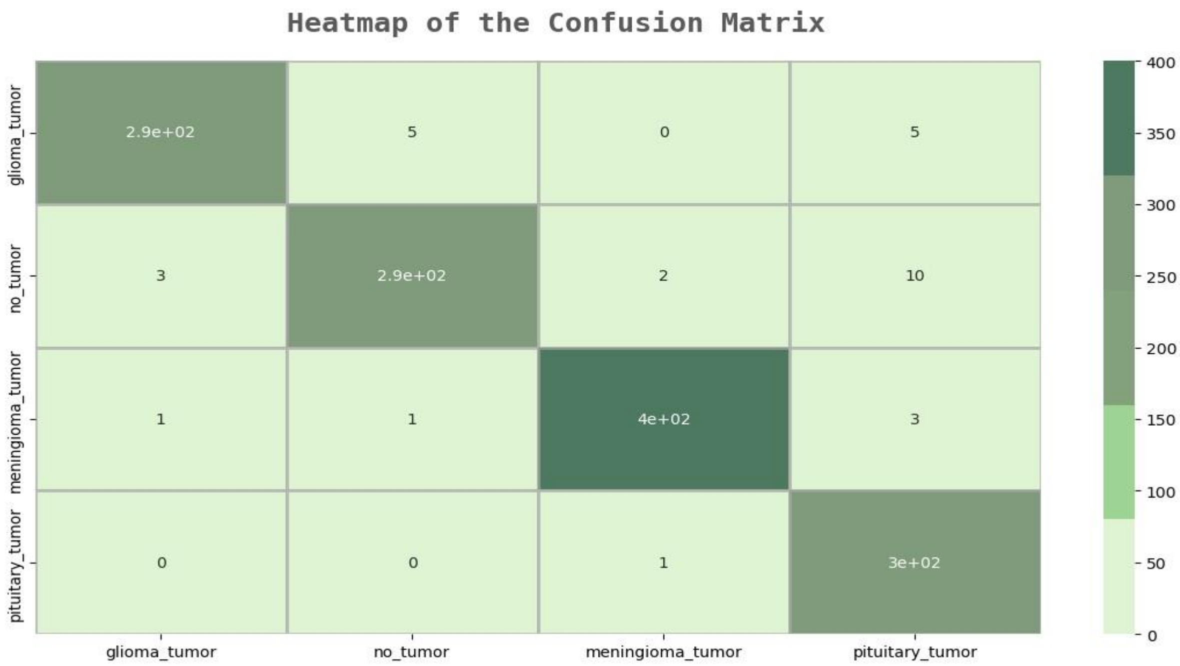


Figure 7.16
Confusion matrix

Overall, the results demonstrate the effectiveness of transfer learning using a pretrained Resnet-50 model for brain tumor classification. By leveraging the pre-

trained weights and features of the Resnet-50 model, the proposed approach was able to achieve high-performance metrics with a relatively small amount of training

data. This approach can be useful for other medical image classification tasks where labeled data is limited and expensive to obtain.

CHAPTER EIGHT

SYSTEM DEVELOPMENT

8.1 Overview

System development is the stage in which the system is implemented using different technologies in this chapter we will try to cover all technological tools and methodologies we used to implement our website.

8.2 Methodological assumptions

User and system requirements to activate the system.

8.2.1 User Requirements

- Users should have basic computer skills in operating systems and internet browsers.
- Users must have a connection to access the internet.

8.2.2 System Requirements

Hardware

✓ Basic computer hardware such as core i3 processor, 4 GB RAM, and Microsoft Windows operating system. Software

✓ Internet Browsers such as Internet Explorer, Mozilla Firefox, Google Chrome, etc.

8.3 Used Technologies 8.3.1 JavaScript

JavaScript is a dynamic computer programming language. It is lightweight and most commonly used as a part of web pages, whose implementations allow clientside scripts to interact with the user and make dynamic pages. It is an interpreted programming language with object-oriented capabilities.

8.3.2 Mean Stack

The MEAN stack is a JavaScript-based framework for developing web applications. MEAN is named after Mongo DB, Express, Angular, and Node, the four key technologies that make up the layers of the stack. The MEAN architecture is designed to make building web applications in JavaScript and handling JSON incredibly easy.

8.3.3 Mean Stack Components

- **Angular.js Front End**

Angular.js allows you to extend your HTML tags with metadata to create dynamic, interactive web experiences much more powerfully than, say, building them yourself with static HTML and JavaScript (or jQuery).

• **Express.js and Node.js Server Tier**

Express.js has powerful models for URL routing (matching an incoming URL with a server function) and handling HTTP requests and responses. By making XML HTTP requests (XHRs), GETs, or POSTs from Angular.js front end, that can connect to Express.js functions that power the application. Those functions in turn use MongoDB's Node.js drivers, either via callbacks or using Promises, to access and update data in the Mongo DB database.

• **Mongo DB Database Tier**

Mongo DB comes in: JSON documents created in Angular.js front end can be sent to the Express.js server, where they can be processed and (assuming they're valid) stored directly in Mongo DB for later retrieval.

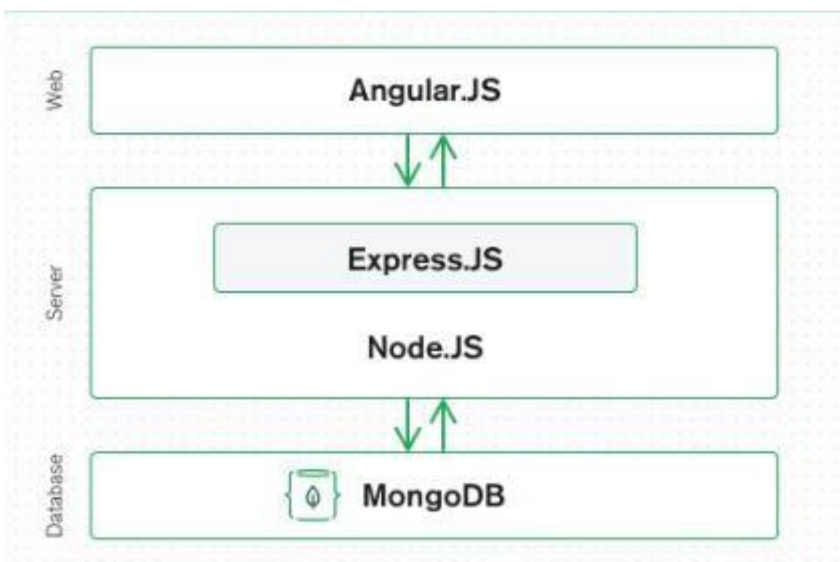


Figure 8.1 Mean stack architecture

8.4 Web Development

When the user opens the site, this is the first screen shown to him, and he cannot use any service within the site except by logging in.

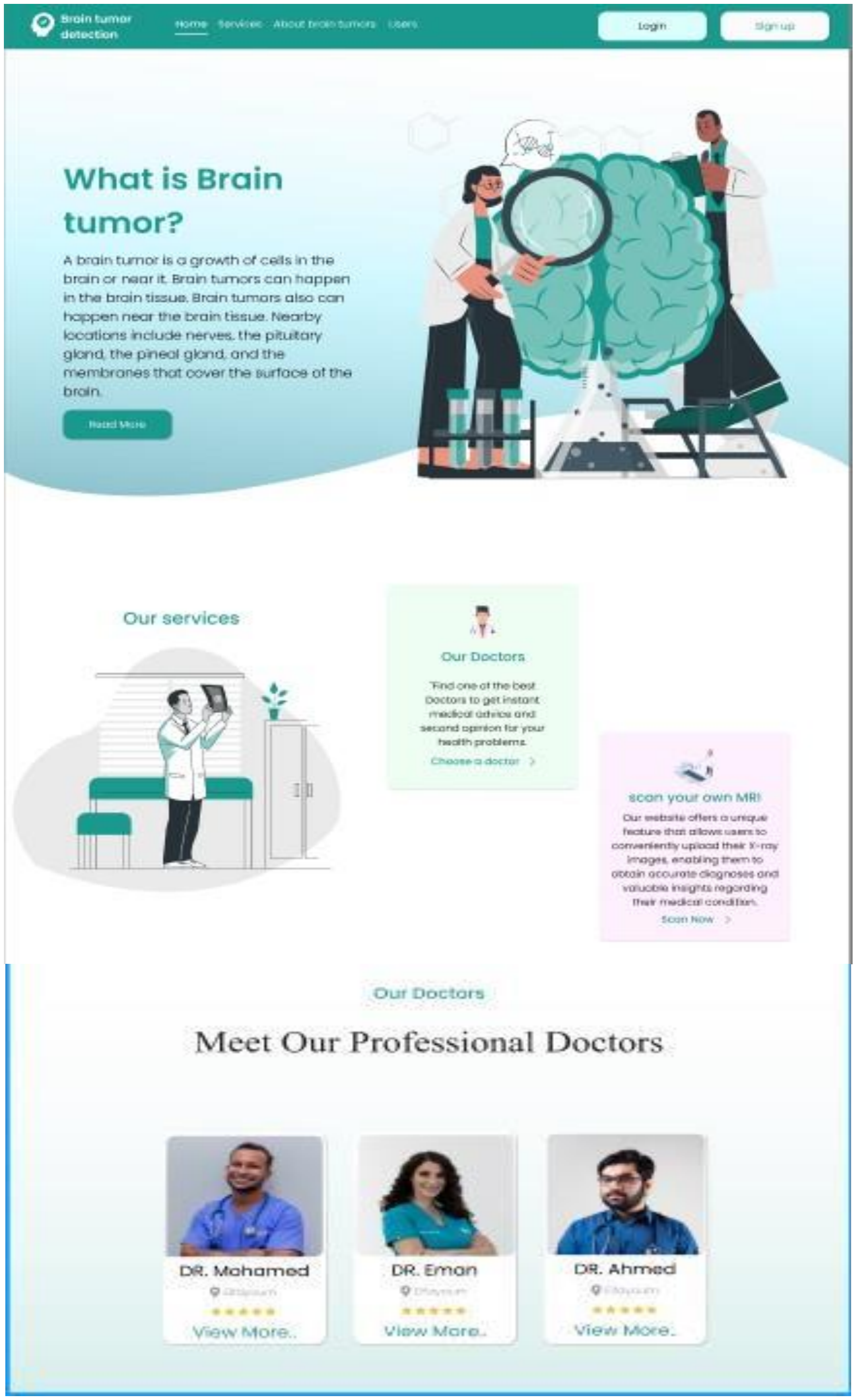


Figure 8.2 Pre-login home page

If the user pressed any of these buttons (choose a doctor – scan now) on the pre-login page it will lead him to the log-in page if he does not have an account he can sign up for an account as a patient or doctor.

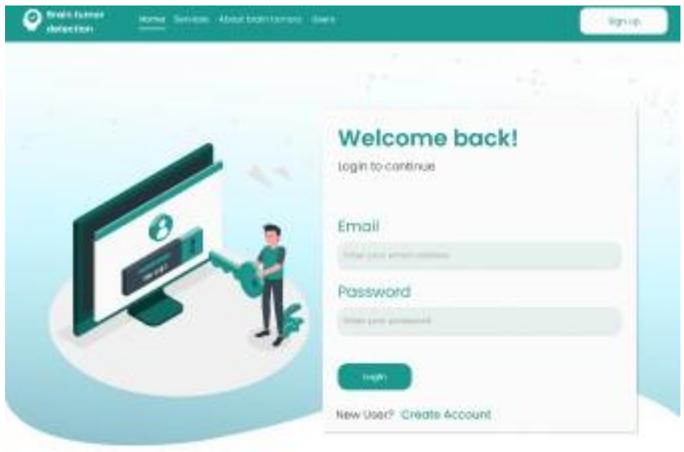


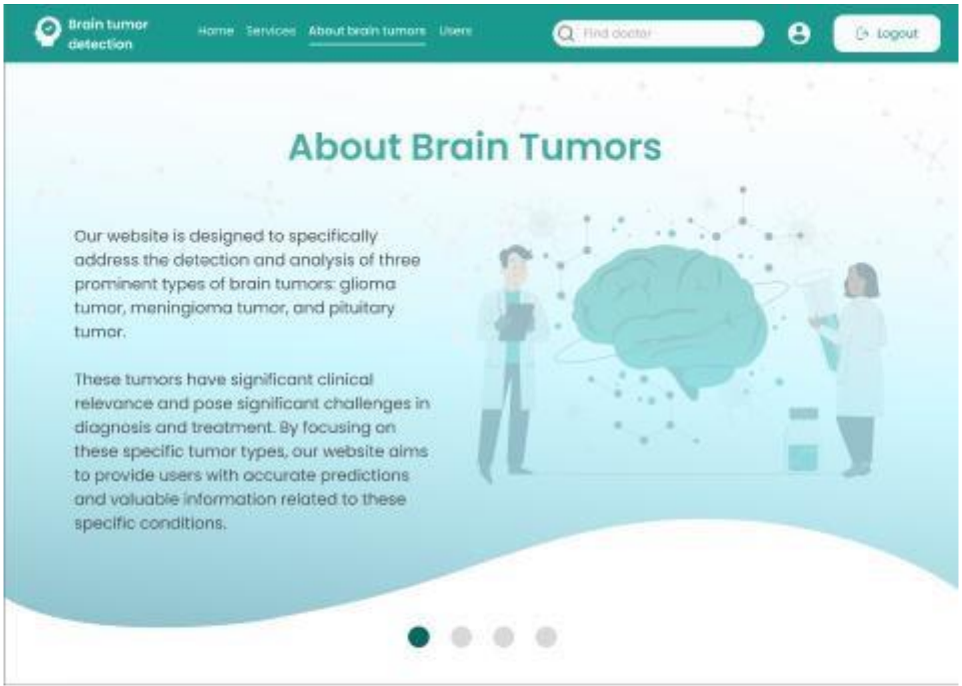
Figure 8.3 Login page

Figure 8.4 Sign-up page

This home page after logging in now users can access website services and search for doctors.

Figure 8.5 Home page after login

The About Brain Tumors, tab shows some information about brain tumors



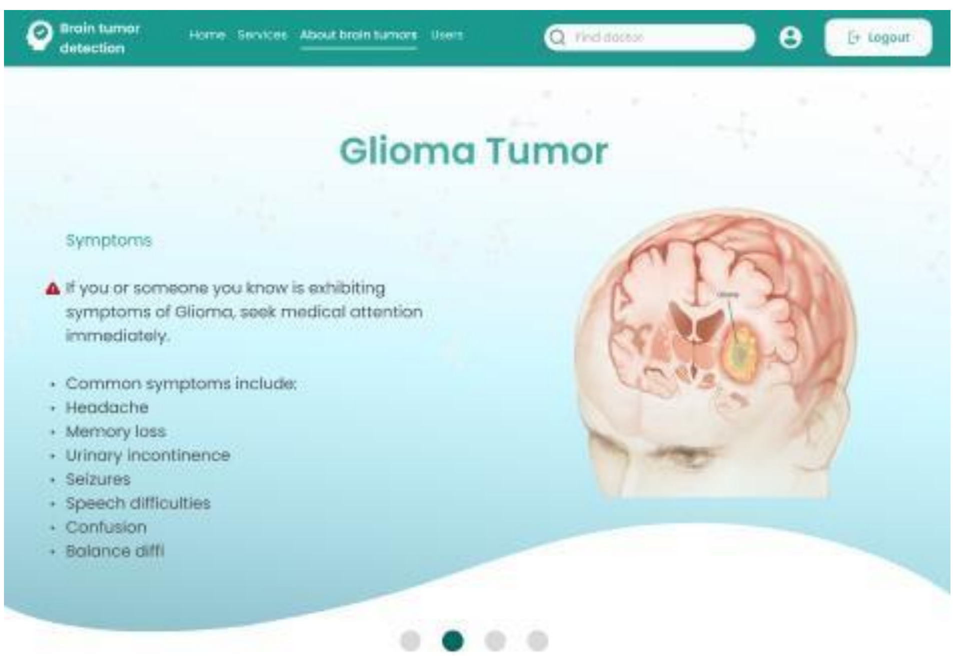


Figure 8.6 About brain tumors tab

The services tab enables users to scan their MRI brain images by browsing the image from their local files and pressing scan then they get the result of their MRI scan, also services tab shows recommended doctors to take an appointment.

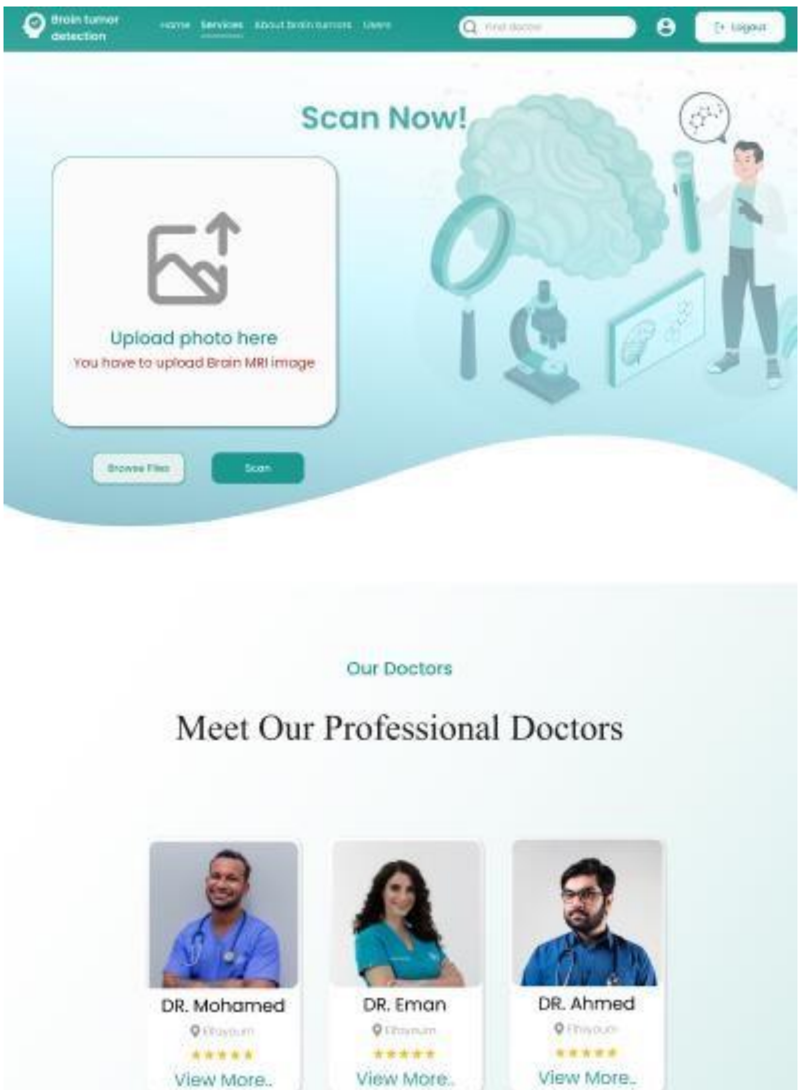


Figure 8.7 Services tab

Scan result appears to the user as a dialog on the same page.

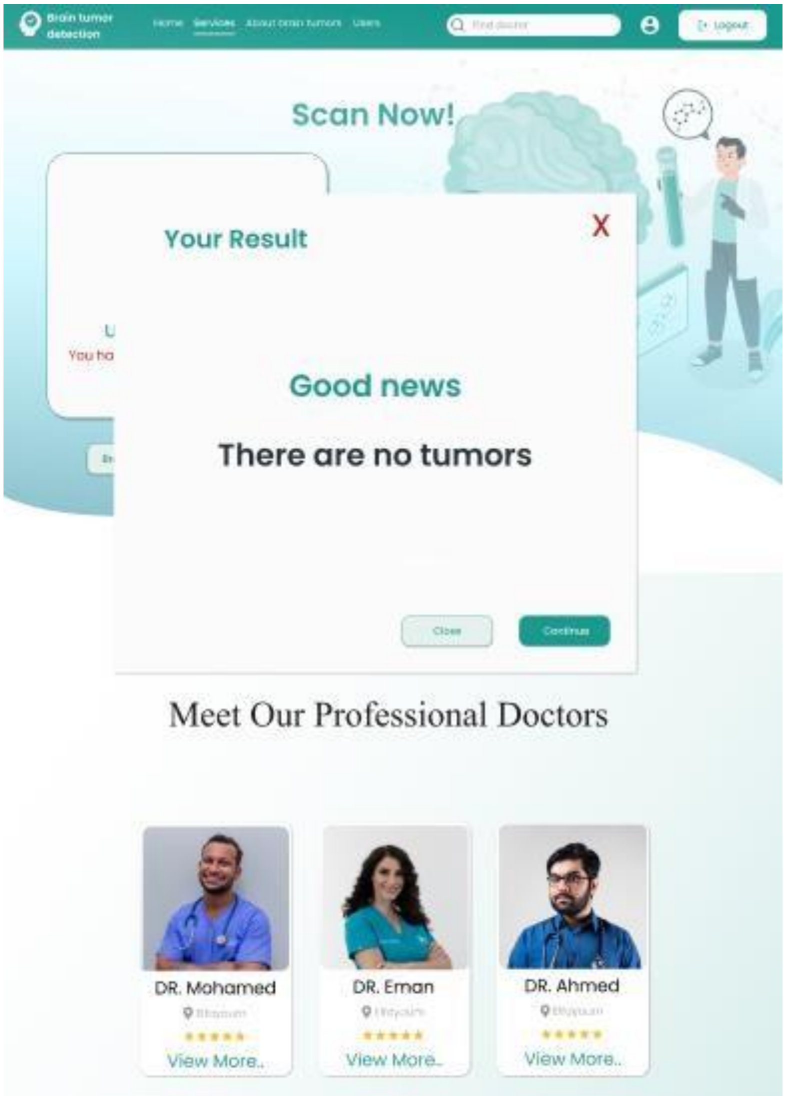


Figure 8.8 Result dialog

The patient profile shows patient info an MRI image of his brain, information, and notes of his physician.

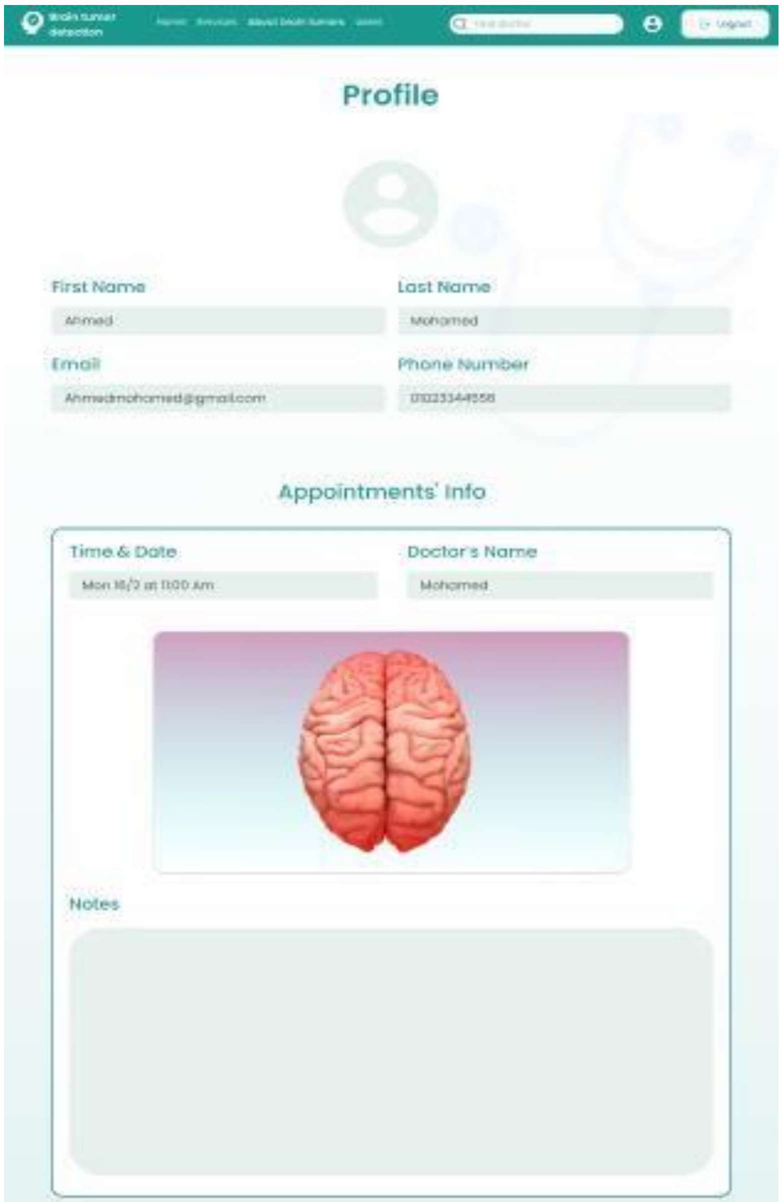
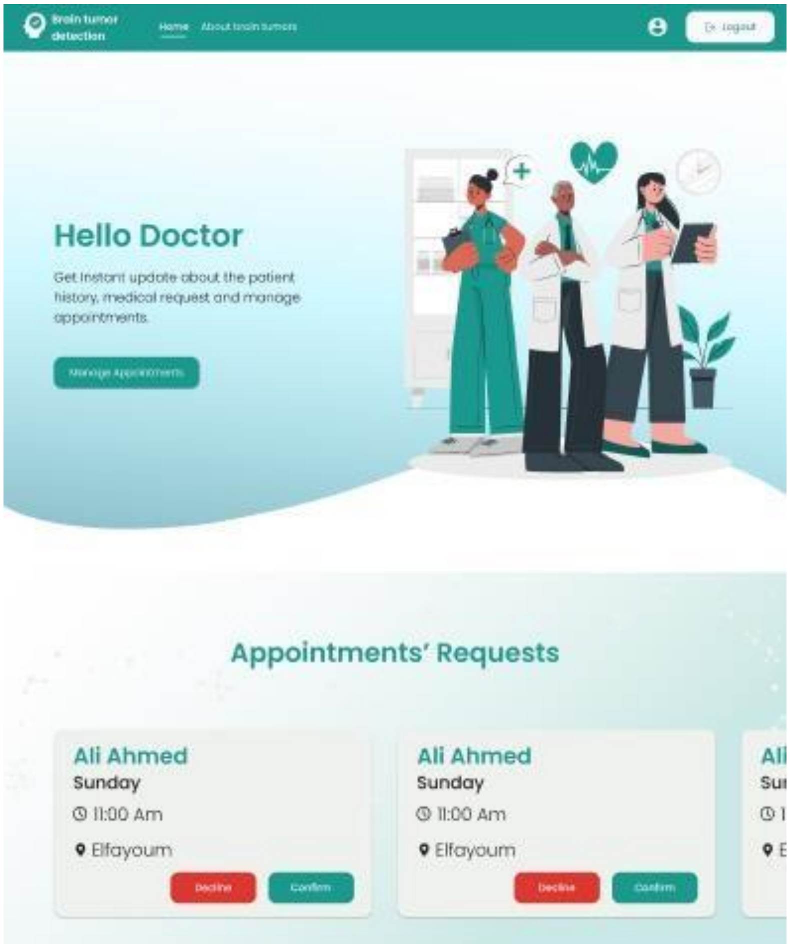


Figure 8.9 Patient profile

Doctor’s home screen provides services for the doctors who are registered to the site,

Doctors can manage their booked appointments they can confirm or decline the appointments requests, and they can see their upcoming appointments.



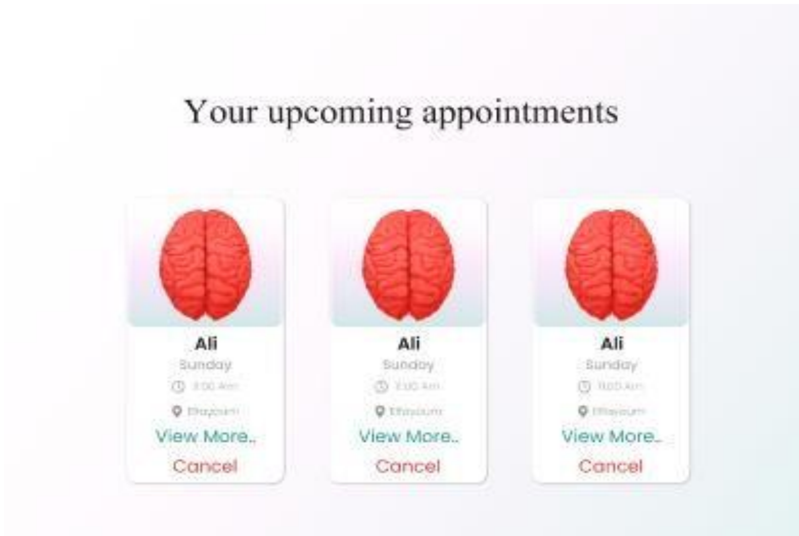


Figure 8.10 Doctor home page

Doctors can edit their profiles and add their available appointments for patients

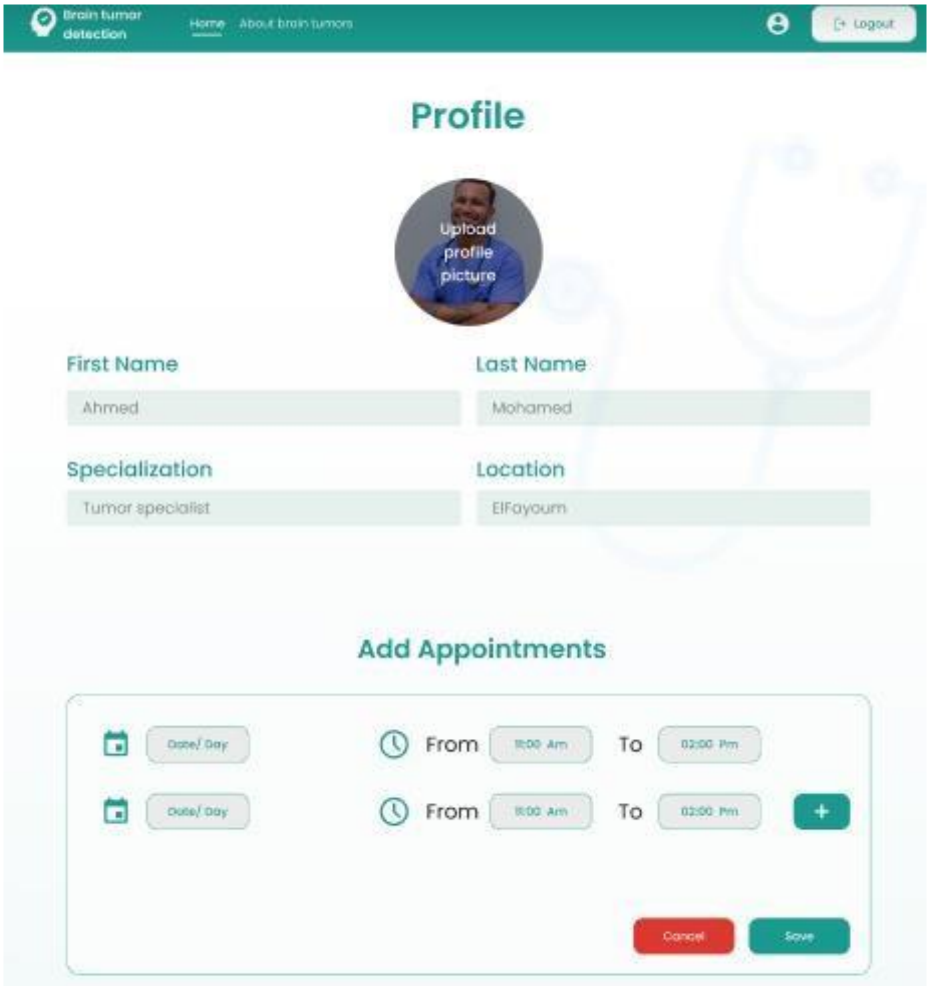


Figure 8.11 Doctor profile

Doctors also can view profiles of their patients and see their medical history and they can add notes to them.

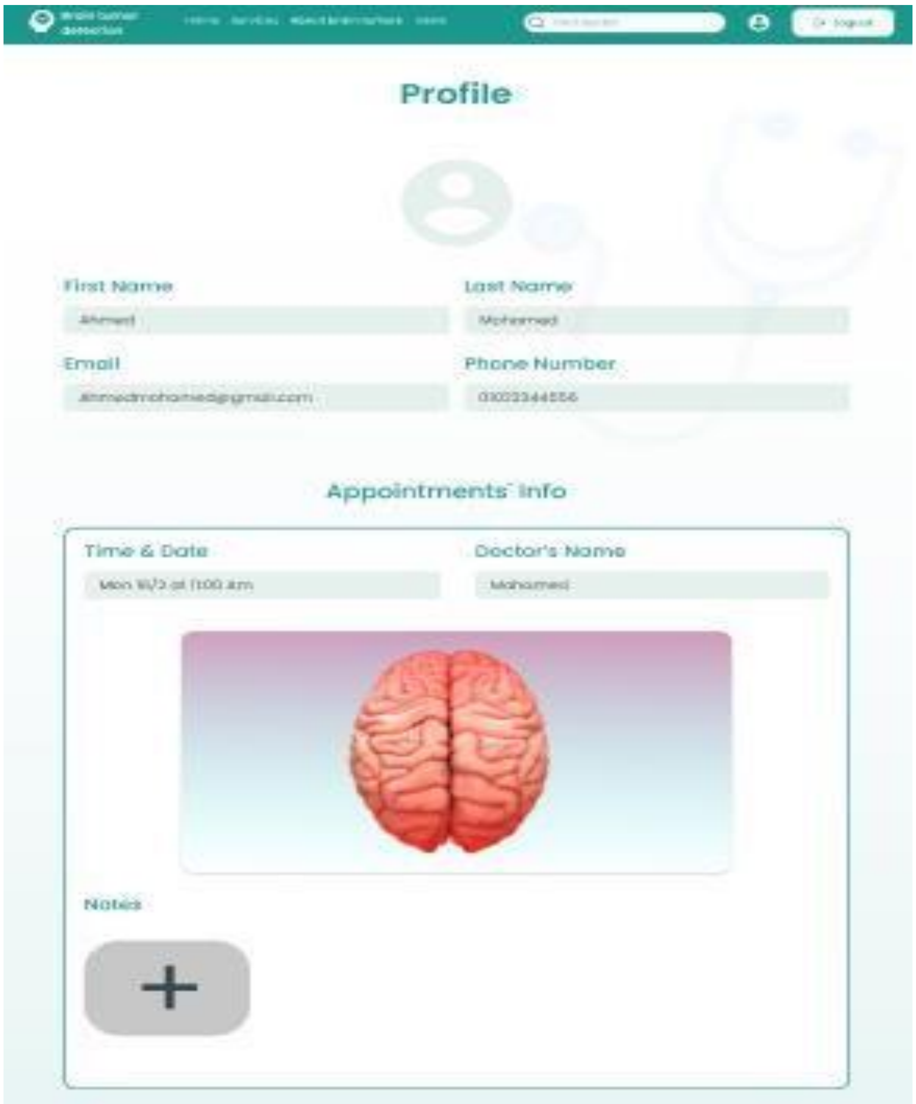


Figure 8.12 Patient profile

CHAPTER NINE

CONCLUSION

9.1 Conclusion

In conclusion, our graduation project introduces a groundbreaking website that utilizes advanced technology to address the challenge of early detection and diagnosis of brain tumors. By focusing on three prominent types of brain tumors, namely glioma tumor, meningioma tumor, and pituitary tumor, our website offers users a powerful tool for assessing their risk and gaining valuable insights into their health.

Through the seamless process of uploading MRI images, our website employs cutting-edge algorithms and expert analysis to provide rapid and accurate diagnoses. This early detection plays a crucial role in facilitating timely intervention and improving treatment outcomes.

Furthermore, our website goes beyond diagnosis by streamlining the appointment booking process. Users have the convenience of choosing from a curated list of specialized doctors, allowing them to connect with the most suitable healthcare professionals for their specific needs.

The motivation behind our project lies in improving accessibility, empowering individuals, and leveraging advancements in AI and medical imaging to combat the challenges posed by brain tumors. By providing a user-friendly platform, we aim to enhance the healthcare experience, enabling users to make informed decisions and take proactive measures for their well-being.

We would like to express our deep gratitude to our research supervisor, Dr. Rasha El Badry, for their guidance and support throughout this project. Their expertise and mentorship have been instrumental in our success.

In conclusion, our project contributes to the field of medical diagnostics by offering a comprehensive website that enables early detection of brain tumors, facilitates appointments with specialized doctors, and empowers individuals in their healthcare journey. We believe that our website has the potential to make a significant impact in improving patient outcomes and positively influencing the fight against brain tumors.

9.2 Difficulties

To address the challenges encountered in the development of a website for brain tumor detection and appointment scheduling, the following solutions can be implemented:

1. Data Availability:	Obtaining a diverse and reliable dataset of brain images for training the AI algorithms can be difficult, requiring access to highquality and annotated data.	Collaborate with healthcare institutions and research organizations to acquire a diverse and reliable dataset. Implement data augmentation techniques to increase dataset size. Ensure proper data anonymization and obtain necessary permissions and consents.
2. Ethical and Legal Considerations:	Adhering to ethical and legal guidelines, such as patient privacy and data protection regulations, is crucial and may involve obtaining permissions and consent.	Strictly adhere to ethical guidelines and data protection regulations. Implement robust security measures to safeguard patient privacy and ensure
		compliance with regulations like HIPAA. Seek legal counsel to navigate any complex legal considerations.
3. Algorithm Development and Validation:	Creating accurate AI algorithms for brain tumor detection requires rigorous validation and testing across different scenarios.	Invest in research and development to create robust AI algorithms. Validate algorithms using large and diverse datasets. Collaborate with medical professionals to obtain expert opinions and feedback during the development and validation processes.
4. Integration with Medical Systems:	Integrating the website with existing medical systems, like electronic health records or appointment scheduling platforms, may require collaboration and technical alignment.	Establish partnerships with healthcare institutions to facilitate integration with their existing medical systems. Coordinate with IT departments and developers to align technical requirements and ensure seamless data exchange between systems.

5. Performance and Scalability:	Ensuring the website can handle a large volume of user requests, process images efficiently, and provide accurate predictions is essential and may require infrastructure optimization.	Optimize the website's infrastructure to handle high volumes of user requests. Utilize cloudbased solutions for scalability. Employ efficient algorithms and parallel processing techniques to reduce processing time and improve performance.
6. Continuous Updates and Maintenance:	Keeping the website up to date with the latest advancements, integrating new algorithms, and addressing bugs require ongoing maintenance and updates.	Establish a dedicated team for ongoing maintenance and updates. Stay informed about advancements in medical imaging and AI algorithms. Regularly monitor the website for vulnerabilities and address any bugs promptly. Provide regular updates to users to enhance the website's functionality and performance.

Table 9.1

By implementing these solutions, the challenges encountered in the development of the website can be effectively addressed, ensuring a successful and reliable platform for brain tumor detection and appointment scheduling.

9.3 Future Work

In addition to the existing capabilities of our website, a potential avenue for future work is the development of a mobile application that extends the functionality to smartphones. This expansion would enable users to access brain tumor detection and diagnosis services directly from their smartphones, providing increased convenience and accessibility. Some potential areas of focus for incorporating the website into a smartphone application include:

1. Expansion of Tumor Types: Currently, our website focuses on glioma tumors, meningioma tumors, and pituitary tumors. Future work can involve expanding the scope to include additional types of brain tumors, such as astrocytoma, medulloblastoma, or ependymoma. This expansion would broaden the utility of the website and provide more comprehensive diagnostic capabilities.

2. Mobile Image Capture: Integrating the smartphone's camera capabilities to allow users to capture brain images directly within the application. This feature would eliminate the need for users to transfer images from their cameras to the application separately.

3. Offline Functionality: Implementing offline functionality that allows users to access certain features and information even without an internet connection. This would be particularly beneficial for users in areas with limited connectivity or during travel.

4. Push Notifications: Incorporating push notifications to provide users with timely updates, reminders for appointments, and notifications about the availability of new features or improvements to the application.

5. Integration with Mobile Health Platforms: Exploring integration with mobile health platforms or electronic health record systems commonly used on smartphones. This integration would enable the seamless sharing of relevant medical information between the application and other healthcare providers, promoting coordinated and comprehensive care.

6. Collaborative Research and Validation Studies: Conducting collaborative research studies and validation trials with healthcare institutions can further validate the accuracy and effectiveness of the website in real-world clinical settings. These studies can provide valuable insights, feedback, and opportunities for refinement and optimization.

By extending the capabilities of our website to a smartphone application, we can reach a wider audience and provide individuals with a convenient and accessible tool for brain tumor detection and diagnosis. This expansion would further empower users to take control of their health and facilitate early intervention, leading to improved outcomes and better management of brain tumors.

Appendix

pre-processing:

Train data:

```python
training_folder = '/content/drive/MyDrive/Training'
output_folder = '/content/drive/MyDrive/traning_new'
def enhance_image(img):
  alpha = 1.5 #  (1.0-3.0)
  beta = 0 #  (0-100)
  adjusted = cv2.convertScaleAbs(img, alpha=alpha, beta=beta)
  return img
for class_name in os.listdir(training_folder):
    class_folder = os.path.join(training_folder, class_name)
    output_class_folder = os.path.join(output_folder, class_name)
    os.makedirs(output_class_folder, exist_ok=True)
    for filename in os.listdir(class_folder):
        img = cv2.imread(os.path.join(class_folder, filename))
        enhanced_img = enhance_image(img)
        cv2.imwrite(os.path.join(output_class_folder, filename), enhanced_img)
```

Python

Test Data:

```python
test_path = '/content/drive/MyDrive/Testing'
output_folder = '/content/drive/MyDrive/test_new'
def enhance_image(img):
  alpha = 1.5 #  (1.0-3.0)
  beta = 0 #  (0-100)
  adjusted = cv2.convertScaleAbs(img, alpha=alpha, beta=beta)
  return img
for class_name in os.listdir(training_folder):
    class_folder = os.path.join(training_folder, class_name)
    output_class_folder = os.path.join(output_folder, class_name)
    os.makedirs(output_class_folder, exist_ok=True)
    for filename in os.listdir(class_folder):
        img = cv2.imread(os.path.join(class_folder, filename))
        enhanced_img = enhance_image(img)
        cv2.imwrite(os.path.join(output_class_folder, filename), enhanced_img)
```

Python

Preparation data:

1-load Data

```python
labels = ['glioma_tumor','no_tumor','meningioma_tumor','pituitary_tumor']
```

```python
X_train = []
y_train = []
image_size = 150
for i in labels:
    folderPath = os.path.join('/kaggle/input/brain-tumor/traning_new-20230508T145734Z-001','traning_new',i)
    for j in tqdm(os.listdir(folderPath)):
        img = cv2.imread(os.path.join(folderPath,j))
        img = cv2.resize(img,(image_size, image_size))
        X_train.append(img)
        y_train.append(i)

for i in labels:
    folderPath = os.path.join('/kaggle/input/brain-tumor/test_new-20230508T145945Z-001','test_new',i)
    for j in tqdm(os.listdir(folderPath)):
        img = cv2.imread(os.path.join(folderPath,j))
        img = cv2.resize(img,(image_size,image_size))
        X_train.append(img)
        y_train.append(i)

for i in labels:
    folderPath = os.path.join('/kaggle/input/brain-br32/Training',i)
    for j in tqdm(os.listdir(folderPath)):
        img = cv2.imread(os.path.join(folderPath,j))
        img = cv2.resize(img,(image_size,image_size))
        X_train.append(img)
        y_train.append(i)
```

2-One Hot Encoding:

```
y_train_new = []
for i in y_train:
    y_train_new.append(labels.index(i))
y_train = y_train_new
y_train = tf.keras.utils.to_categorical(y_train)


y_test_new = []
for i in y_test:
    y_test_new.append(labels.index(i))
y_test = y_test_new
y_test = tf.keras.utils.to_categorical(y_test)
```

4-Train and Test Split:

```
X_train,X_test,y_train,y_test = train_test_split(X_train,y_train, test_size=0.2,random_state=101,stratify=y_train)
```

```
X_train.shape,X_test.shape,y_train.shape,y_test.shape
```

```
((9160, 150, 150, 3), (2290, 150, 150, 3), (9160,), (2290,))
```

MODEL ResNet-50:

1-Load model:

```
weights_path = '/kaggle/input/resnet-50/resnet50_weights_tf_dim_ordering_tf_kernels.h5'
```

```
resne_2 = tf.keras.applications.ResNet50(weights=weights_path, include_top=False, input_shape=(150,150,3)
```

```
resne_2.summary()
```

2-Fine-tuning

```
# Freeze the ResNet-50 layers
for layer in resne_2.layers[:133]:
    layer.trainable=False
```

3-Summery of model:

```
resne_2.summary()
```

```
Model: "resnet50"
__________________________________________________________________________________________________
 Layer (type)                   Output Shape         Param #     Connected to
==================================================================================================
 input_3 (InputLayer)           [(None, 150, 150, 3  0           []
                                )]

 conv1_pad (ZeroPadding2D)      (None, 156, 156, 3)  0           ['input_3[0][0]']

 conv1_conv (Conv2D)            (None, 75, 75, 64)   9472        ['conv1_pad[0][0]']

 conv1_bn (BatchNormalization)  (None, 75, 75, 64)   256         ['conv1_conv[0][0]']

 conv1_relu (Activation)        (None, 75, 75, 64)   0           ['conv1_bn[0][0]']

 pool1_pad (ZeroPadding2D)      (None, 77, 77, 64)   0           ['conv1_relu[0][0]']

 pool1_pool (MaxPooling2D)      (None, 38, 38, 64)   0           ['pool1_pad[0][0]']

 conv2_block1_1_conv (Conv2D)   (None, 38, 38, 64)   4160        ['pool1_pool[0][0]']

 conv2_block1_1_bn (BatchNormal  (None, 38, 38, 64)  256         ['conv2_block1_1_conv[0][0]']
 ization)

 conv2_block1_1_relu (Activatio  (None, 38, 38, 64)  0           ['conv2_block1_1_bn[0][0]']
...
Total params: 23,587,712
Trainable params: 16,094,720
Non-trainable params: 7,492,992
```

4-Sequential model :

```
model_2 = tf.keras.models.Sequential()
model_2.add(resne_2)
model_2.add(tf.keras.layers.GlobalAveragePooling2D())
model_2.add(tf.keras.layers.Dropout(0.15))
model_2.add(tf.keras.layers.Dense(4, activation='softmax'))
```

5-Train model:

```
Epoch 1/200
258/258 [==============================] - 39s 107ms/step - loss: 0.2577 - accuracy: 0.9163 - val_loss: 0.1078 - val_accuracy: 0.9607 - lr: 0.0010
Epoch 2/200
258/258 [==============================] - 24s 92ms/step - loss: 0.0684 - accuracy: 0.9766 - val_loss: 0.0405 - val_accuracy: 0.9880 - lr: 0.0010
Epoch 3/200
258/258 [==============================] - 25s 96ms/step - loss: 0.0292 - accuracy: 0.9905 - val_loss: 0.0581 - val_accuracy: 0.9825 - lr: 0.0010
Epoch 4/200
258/258 [==============================] - ETA: 0s - loss: 0.0299 - accuracy: 0.9904
Epoch 4: ReduceLROnPlateau reducing learning rate to 0.0003000000142492354.
258/258 [==============================] - 24s 93ms/step - loss: 0.0299 - accuracy: 0.9904 - val_loss: 0.0588 - val_accuracy: 0.9814 - lr: 0.0010
Epoch 5/200
258/258 [==============================] - 25s 97ms/step - loss: 0.0106 - accuracy: 0.9968 - val_loss: 0.0145 - val_accuracy: 0.9934 - lr: 3.0000e-04
Epoch 6/200
258/258 [==============================] - 24s 92ms/step - loss: 0.0049 - accuracy: 0.9989 - val_loss: 0.0155 - val_accuracy: 0.9956 - lr: 3.0000e-04
Epoch 7/200
258/258 [==============================] - 25s 96ms/step - loss: 0.0020 - accuracy: 0.9998 - val_loss: 0.0115 - val_accuracy: 0.9956 - lr: 3.0000e-04
Epoch 8/200
258/258 [==============================] - ETA: 0s - loss: 7.2958e-04 - accuracy: 1.0000
Epoch 8: ReduceLROnPlateau reducing learning rate to 9.000000427477062e-05.
258/258 [==============================] - 24s 93ms/step - loss: 7.2958e-04 - accuracy: 1.0000 - val_loss: 0.0106 - val_accuracy: 0.9956 - lr: 3.0000e-04
Epoch 9/200
258/258 [==============================] - 25s 97ms/step - loss: 5.5850e-04 - accuracy: 1.0000 - val_loss: 0.0091 - val_accuracy: 0.9956 - lr: 9.0000e-05
Epoch 10/200
258/258 [==============================] - 25s 96ms/step - loss: 5.7939e-04 - accuracy: 1.0000 - val_loss: 0.0086 - val_accuracy: 0.9967 - lr: 9.0000e-05
Epoch 11/200
...
258/258 [==============================] - 24s 92ms/step - loss: 2.7961e-04 - accuracy: 1.0000 - val_loss: 0.0080 - val_accuracy: 0.9956 - lr: 8.1000e-06
Epoch 17/200
258/258 [==============================] - 24s 92ms/step - loss: 3.7890e-04 - accuracy: 0.9999 - val_loss: 0.0086 - val_accuracy: 0.9945 - lr: 2.4300e-06
Epoch 17: early stopping
```

References

[1] DAVID CROSBY, SANGEETA BHATIA, KEVIN M. BRINDLE, LISA M. COUSSENS, CAROLINE DIVE, MARK EMBERTON, SADIK ESENER, REBECCA C. FITZGERALD, SANJIV S. GAMBHIR,PETER KUHN, TIMOTHY R. REBBECK, and SHANKAR BALASUBRAMANIAN "Early detection of cancer", Published: SCIENCE, Posted: 18 Mar 2022

[2] Brain Anatomy and How the Brain Works | Johns Hopkins Medicine

[3] Msoud Nickparvar "Brain Tumor MRI Dataset", Published: Kaggle, Updated: 2 years ago

[4] Nahid Ferdous Aurna, Mohammad Abu Yousuf, Kazi Abu Taher, A.K.M. Azad, Mohammad Ali Moni "A classification of MRI brain tumor based on two stage feature level ensemble of deep CNN models", Published: ScienceDirect, Posted: July 2022

[5] Wadhah Ayadi, Wajdi Elhamzi, Mohamed Atri, Imen Charfi," Deep CNN for Brain Tumor Classification", Published: Springer, Posted: February 2021

[6] Sumeet Saurav, Ayush Sharma, Ravi Saini & Sanjay Singh ," An attention-guided convolutional neural network for automated classification of brain tumor from MRI", Published: SpringerLink, Posted: 01 September 2022

[7] Manoj Kumar Panda , Akhilesh Sharma , Vatsalya Bajpai , Badri Narayan Subudhi , Veerakumar Thangaraj , Vinit Jakhetiya, " Encoder and decoder network with ResNet-50 and global average feature pooling for local change detection" , Published: ScienceDirect, Posted: September 2022

[8] Rajat Mehrotra, M.A. Ansari, Rajeev Agrawal, R.S. Anand," A Transfer Learning approach for AI-based classification of brain tumors" ,Published: ScienceDirect, Posted: 15 December 2020

[9] Laith Alzubaidi, Jinglan Zhang, Amjad J. Humaidi, Ayad Al-Dujaili, Ye Duan, Omran Al-Shamma, J. Santamaría, Mohammed A. Fadhel, Muthana Al-Amidie & Laith Farhan," Review of deep learning: concepts, CNN architectures, challenges, applications, future directions" , Published: SpringerOpen, Posted: 31 March 2021

[10] He Huang, Guang Yang, Wenbo Zhang, Xiaomei, Weiji Yang, Weiwei Jiang and Xiaobo Lai," A Deep Multi-Task Learning Framework for Brain Tumor Segmentation" , Published: frontiers, Posted: 04 June 2021

[11] Beatriz Ocaña-Tienda, Julián Pérez-Beteta, José D. Villanueva-García, José A. Romero-Rosales, David Molina-García, Yannick Suter, Beatriz Asenjo, David Albillo, Ana Ortiz de Mendivil, Luis A. Pérez-Romasanta, Elisabet González-Del

Portillo, Manuel Llorente, Natalia Carballo, Fátima Nagib-Raya, Maria Vidal-Denis, Belén Luque, Mauricio Reyes, Estanislao Arana & Víctor M. Pérez-García, “A comprehensive dataset of annotated brain metastasis MR images with clinical and radiomic data” , Published: Scientific Data, Posted: 14 April 2023

[12] Aakash Kaushik, “Understanding ResNet50 architecture” , Published: OpenGenus

[13] Kaiming He Xiangyu Zhang Shaoqing Ren Jian Sun, “Deep Residual Learning for Image Recognition”, Published: Microsoft Research, Posted: 10 Dec 2015

[14] [ResNet-50: The Basics and a Quick Tutorial (datagen.tech)]

[15] Suvaditya Mukherjee, “The Annotated ResNet-50”, Published: Data Science, Posted: Aug 18, 2022

[16] Niklas Donges, “What Is Transfer Learning? Exploring the Popular Deep Learning Approach.”, Published: Built In