ASSIUT UNIVERSITY

FACULTY OF COMPUTERS AND INFORMATION

Comprehensive Guide to Brain Tumor Analysis Using MRI: Classification and Segmentation Techniques

A Graduation Project Team:

Mahmoud Mohamed Ahmed

Ahmed Magdy Mohamed

Mohamed Eid Saber Saleh

Mahmoud Hussein Mohamed

Mahmoud Mohamed Abdelhalim

Shrouk Sobhy Abdelsabour

Salma Hamada Abdelnaem

Madonna Mamdoh Louis

Under Supervision of

Dr. Sara Salah Mohamed

July 2024

Table of Content

Chapter (1)

Proposal

Contents

1. Project Abstract

This project aims to develop a sophisticated system for the analysis of brain tumors using MRI, employing advanced classification and segmentation techniques, specifically utilizing the UNET model. The system includes a comprehensive web and mobile application that allows doctors to upload, analyze, and document MRI images of patients. Additionally, the application provides functionalities for patient medication management, communication between patients and doctors, and access to medical histories. The entire project is developed in Arabic to cater to Arabic-speaking users, ensuring accessibility and usability.

2. Project Objectives

1- Develop an Automated MRI Analysis System:

- Create a system to classify MRI images as brain or non-brain images.
- Develop functionality to detect and segment brain tumors in MRI images using the UNET model.

2- Enhance Doctor-Patient Interaction:

- Provide a platform for doctors to upload MRI images, analyze them, and document their findings.
- Enable patients to access their medical reports and manage their medication schedules.

3- Improve Patient Data Management:

- Implement a database to store patient medical history, including MRI reports and doctor's observations.
- Facilitate seamless communication between doctors and patients through a chat feature.

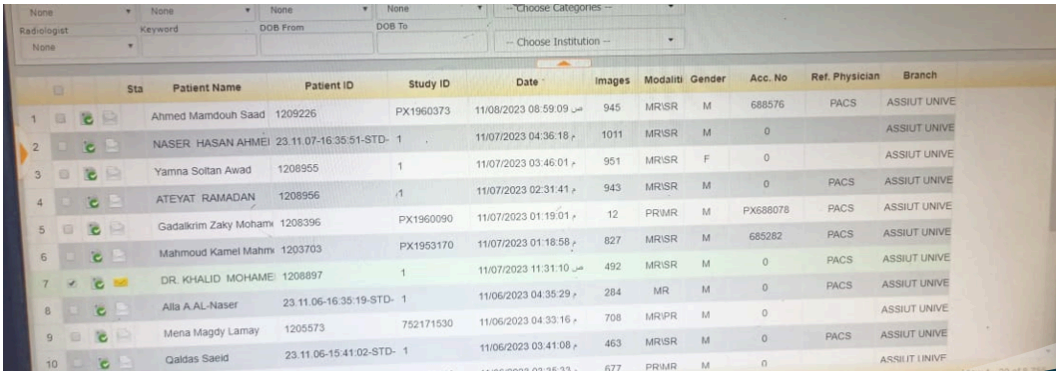
4- Develop an Arabic Language Interface:

- Ensure the entire system, including user interfaces and documentation, is available in Arabic to serve Arabic-speaking users.

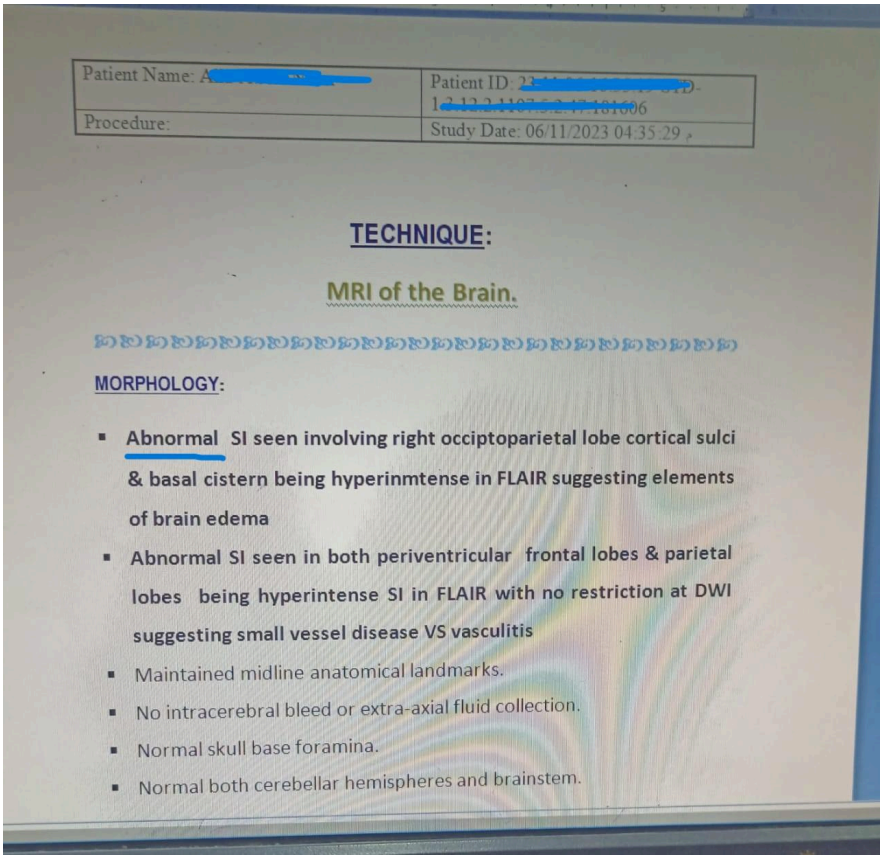
3.Data Driven

In collaboration with Assiut University Hospital, the Department Of MRI Radiology, Our team has a partnership that allows us access to a wealth of MRI images and patient reports :

1. Providing a robust dataset for analysis from the Hospital Server**(PAXE)**.



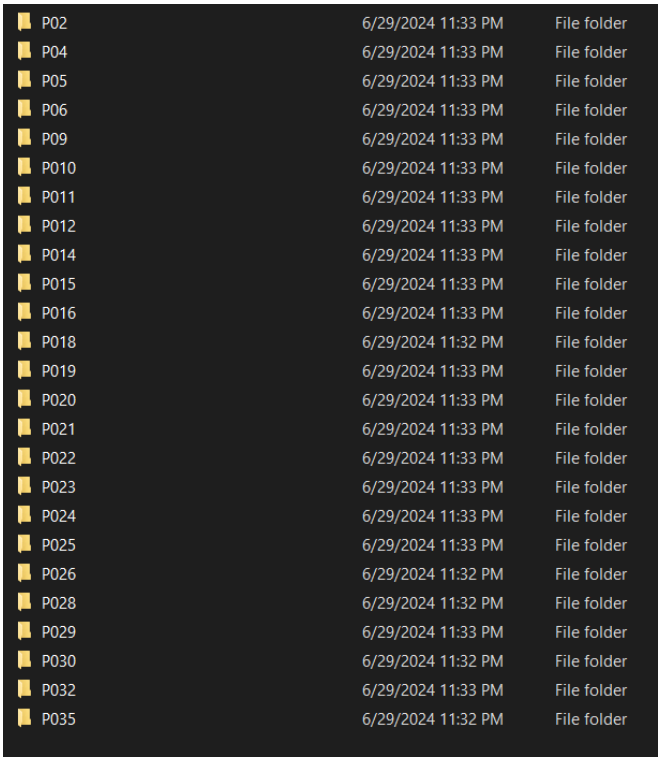
2. Searching for a Brain Tumor patients from it's report to collecting his data after:



3. Gathering patients data in a sheet for extracting them from IT department:

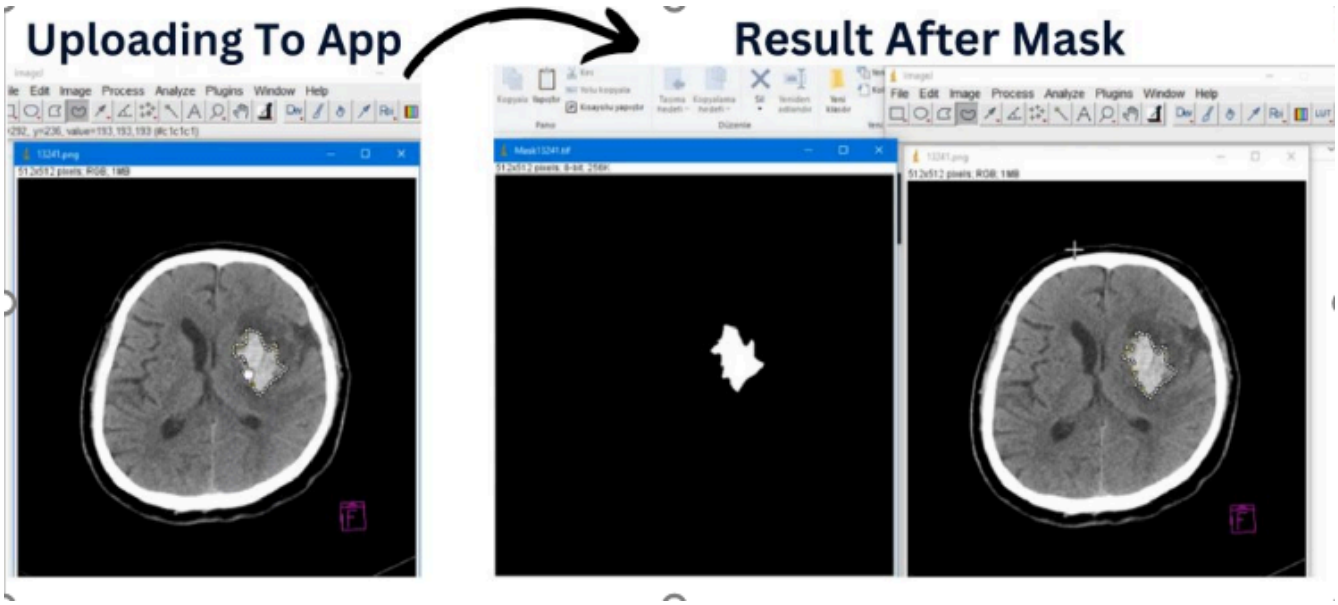
	A	B	C	D
1	Name	ID	Date	Status
2	Fatma mohamed Ali	1138152	2023-12-07	meningoma
3	Asmaa Mohamed	983154	2023-12-07	glioma
4	Yasser Mohamed	1150862	2023-12-07	Granulamatous lesions
5	FatmaMahmoud Ahmed	1150514	2023-12-07	sol
6	Maged samir	1060317	2023-12-07	lesion
7	yara sayed Anter	1151361	2023-12-07	lesion
8	Haythem Ali Amin	1207999	2023-12-06	lesion
9	Ibrahim Thabet	1221574	2023-12-06	mass
10	Ahmed Khalid Abd Alnaser		2023-12-05	Sol
11	Abdel Zaher Abou Zaid	906207	2023-12-04	Schwanoma
12	karam thabet	1093270	2023-12-03	carcinoma
13	Samar Gamal	1212531	2023-12-03	sol
14	Em AlRahman salah	1220197	2023-12-02	lesion

4. Receiving the sequence of 512 MRI images of each patient.

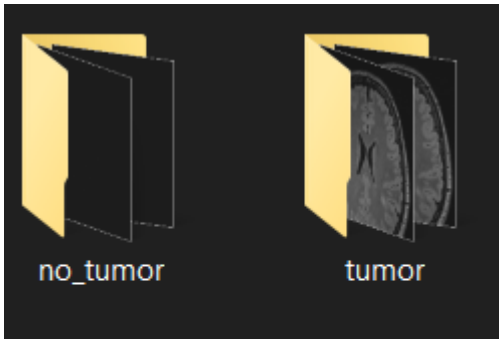


5. Find the sequence of the tumor in each image with the cooperation of the radiologists to.

6. Then Using an application called ImageJ for creating the mask for each image that has a tumor, Under the supervision of expert doctors and radiologists we create masks (Labeling Stage)



7. AI model checks if the image is of a brain or not.
8. AI model splits the images into two folders: tumor and no tumor.
9. Here the segmentation need is ended and we start preparing the data for classification task needs!
10. Dividing each patient images to tumor and no tumor folders to create the data for the classification model



11. With this extensive dataset and expert guidance, our automated system achieves high accuracy in segmentation and classification tasks.

4. Approaches and Methodology

AI Models

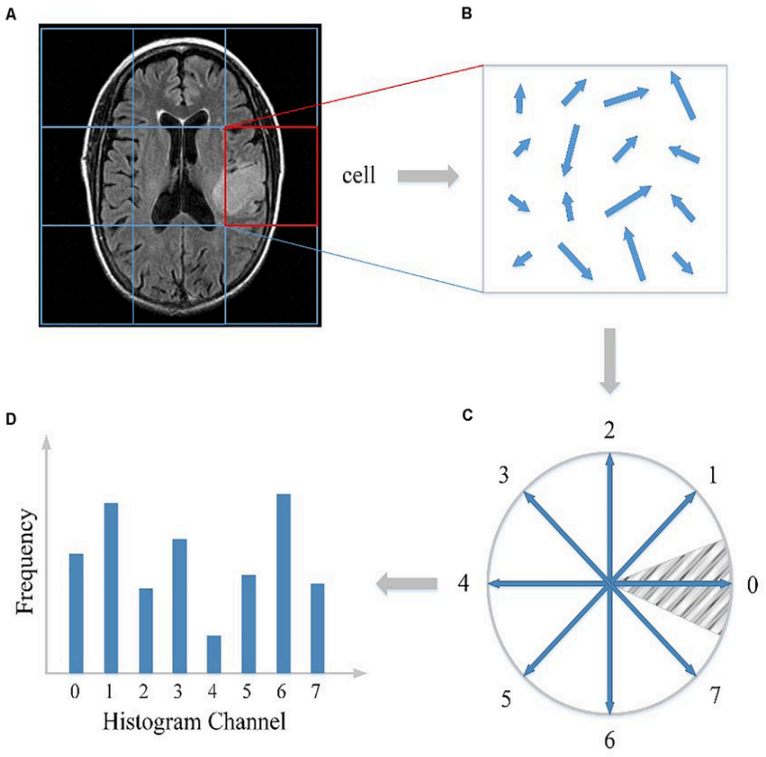
1- Brain MRI Detection Model

Data Preprocessing

1. **Image Loading**: Each image is converted to grayscale for uniformity.
2. **Resizing**: Images are resized to a consistent dimension (128x128 pixels) to ensure uniform feature extraction.
3. **Normalization**: Images are scaled to reduce variability in illumination and contrast, enhancing model robustness.

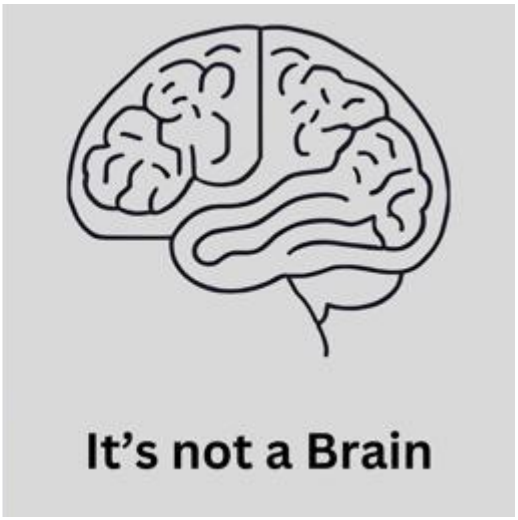
Model Architecture:

The model uses a Support Vector Machine (SVM) with a linear kernel. Features are extracted from images using the Histogram of Oriented Gradients (HOG) technique, which captures essential shape and texture information. This setup allows the model to classify images into two categories: "Brain MRI" and "Not Brain."



Training the Model

1. **Feature Extraction**: HOG features are computed for each image, focusing on gradients and edge directions.
2. **SVM Training**: The extracted features are used to train an SVM, which separates the two classes in a high-dimensional space.
3. **Model Saving**: The trained model is saved for future predictions, allowing quick classification of new images.



2- Brain Tumor Classification Model

1. **Image Augmentation:**
   - To enhance the diversity of the training data and prevent overfitting, data augmentation techniques are used. The ImageDataGenerator from Keras is utilized for this purpose.
2. **Resizing:**
   - Images are resized to a consistent dimension (256x256 pixels)

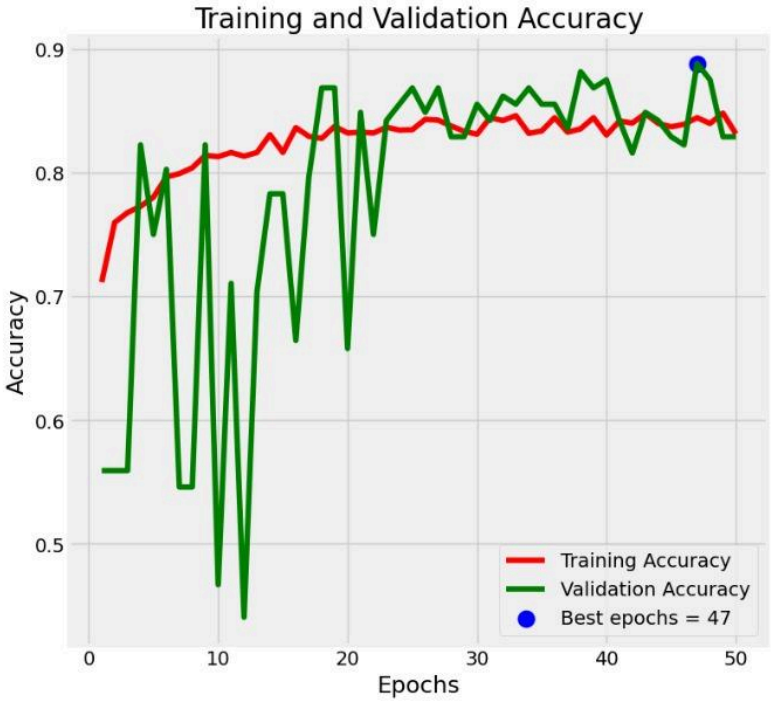
Model Architecture:

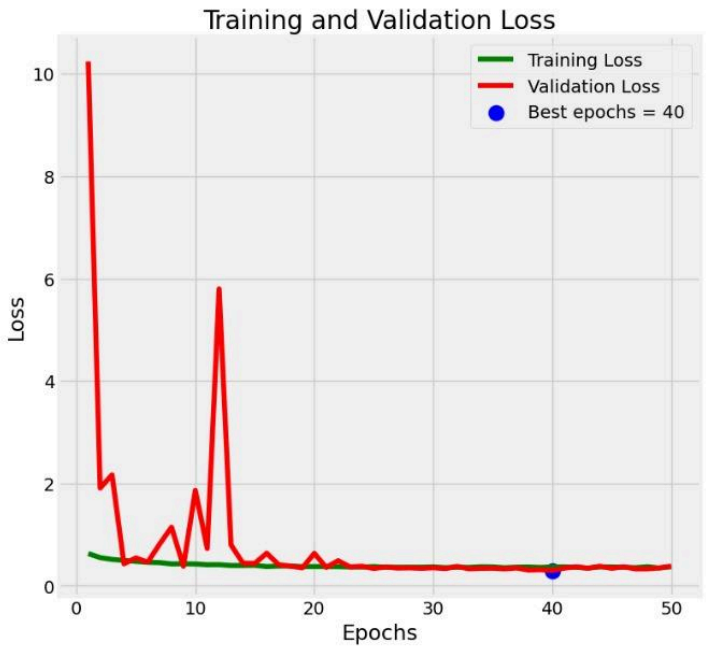
The model uses ResNet50, a deep convolutional neural network pre-trained on the ImageNet dataset. ResNet50 is chosen for its ability to learn deep representations and handle the vanishing gradient problem effectively.

Training the Model

The model is trained using augmented data. Early stopping and model checkpointing are employed to save the best model and prevent overfitting.

Evaluation





3- UNET Model for Segmentation with VGG16

Data Preprocessing

To ensure that the MRI images are clean and suitable for analysis, the following preprocessing steps will be implemented:

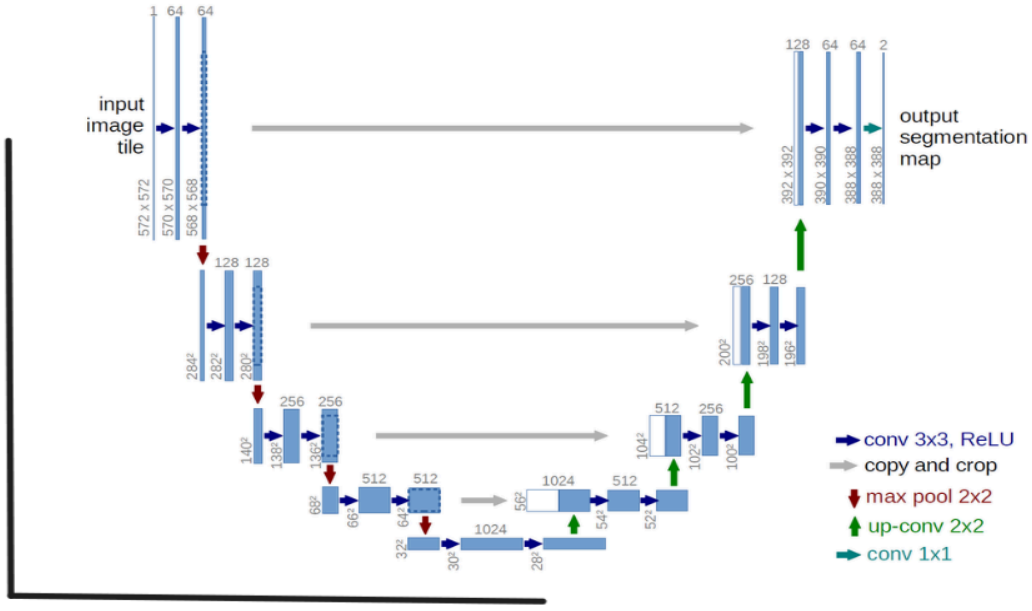
1. **Image Normalization:**
   - Standardize the pixel intensity values to a consistent range to ensure uniformity across the dataset.
2. **Image Augmentation:**
   - Generate additional training samples by applying transformations like rotation, flipping, and scaling to improve the model's robustness.

3. **Resizing:**

   - Resize images to a standard size compatible with the UNET model input requirements.

Model Architecture:

- Implement the UNET architecture with an encoder for downsampling and a decoder for upsampling, including skip connections to retain spatial information.
- Fine-tune the encoder using a pre-trained VGG16 model to leverage learned features.



VGG16

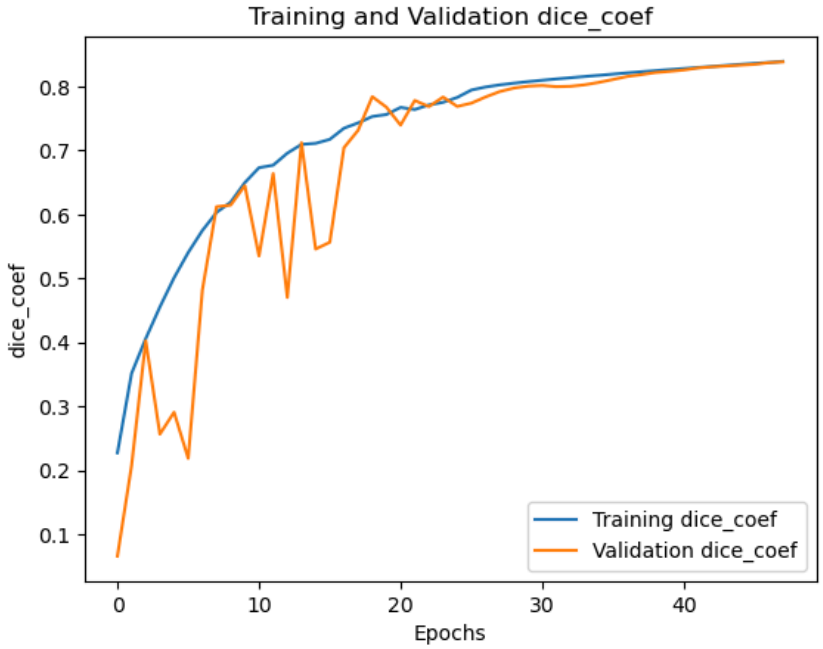
Training the Model:

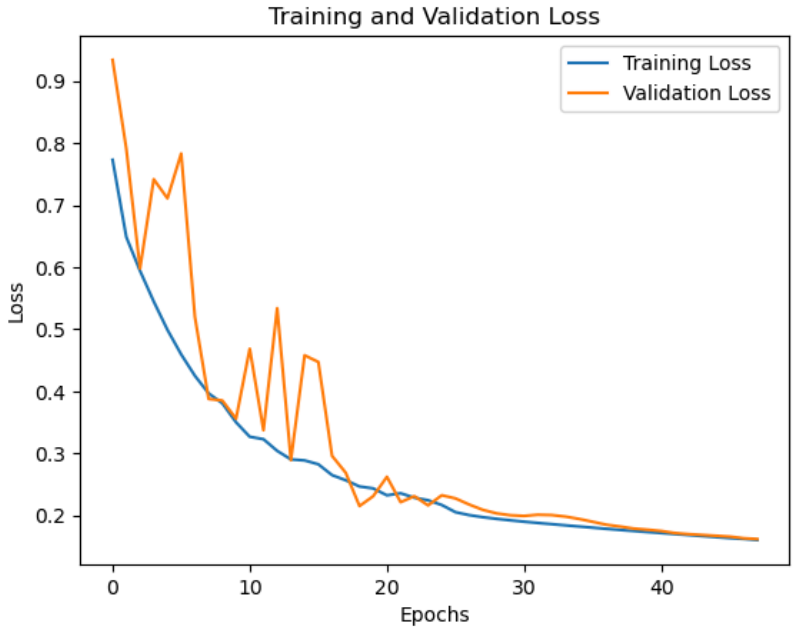
- Use a labeled dataset of MRI images with annotated brain tumors.
- Split the dataset into training, validation, and test sets.
- Train the model using appropriate loss functions (Dice coefficient, Intersection over Union) and optimization algorithms (Adam).

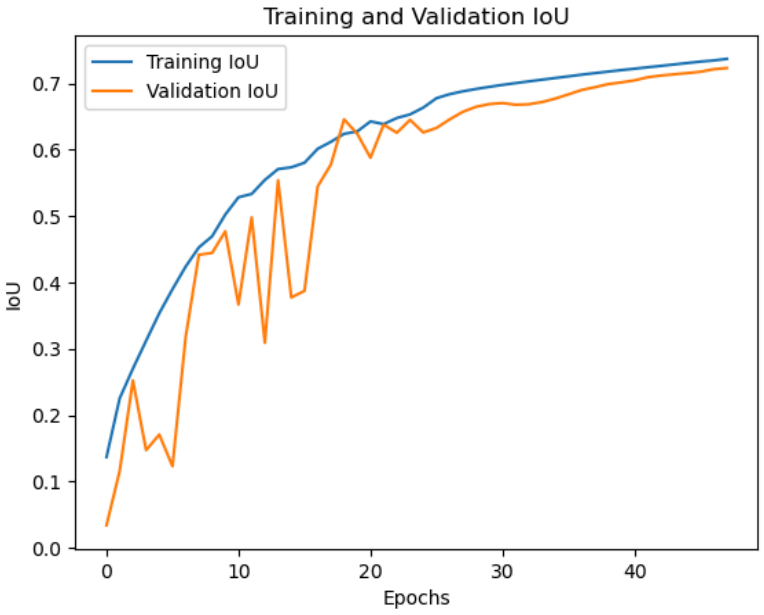
Validation and Testing:

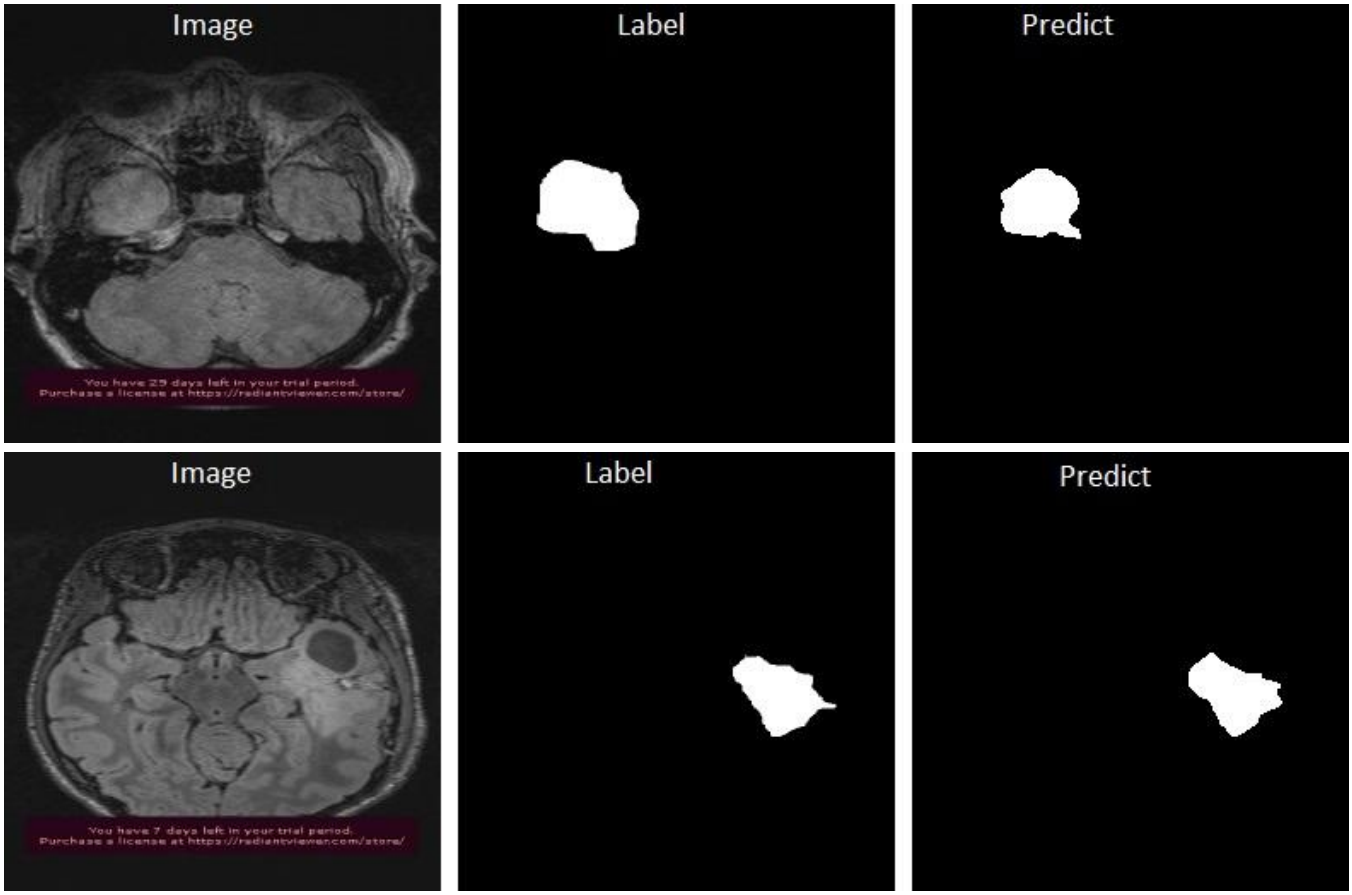
- Validate the model on a separate validation set to tune hyperparameters.
- Test the model on an unseen test set to evaluate its performance and generalization capability.

Evaluation


Training and Validation dice_coef
dice_coef
Epochs
Training dice_coef
Validation dice_coef


Training and Validation Loss
Loss
Epochs
Training Loss
Validation Loss


Training and Validation IoU
IoU
Epochs
Training IoU
Validation IoU



Web and Mobile Application Development

Frontend Development

The frontend for both web and mobile applications will be developed using HTML and CSS:

1. **User Interface Design:**
   - Design user-friendly and intuitive interfaces for doctors and patients.
   - Ensure the layout is responsive and compatible with different devices.
2. **Implementation:**
   - Develop static web pages using HTML.
   - Style the pages with CSS to ensure a professional and accessible design.

Backend Development

The backend will be developed using Node.js to handle the business logic and interactions with the database:

1. **API Development:**
   - Develop RESTful APIs to handle requests from the frontend.
   - Implement endpoints for uploading MRI images, retrieving reports, managing patient data, and facilitating chat functionalities.
2. **Database Integration:**
   - Use a database (MongoDB) to store patient information, MRI reports, and chat logs.
   - Ensure secure and efficient data retrieval and storage mechanisms.

Model Deployment

The trained UNET model will be deployed using a Flask API:

1. **Model Serialization:**
   - Serialize the trained model using library keras.
2. **API Development:**
   - Develop a Flask API to serve the model for inference.
   - Implement endpoints for image classification, tumor detection, and segmentation.
3. **Integration:**
   - Integrate the Flask API with the Node.js backend to facilitate communication between the frontend and the model.

Language Localization

The entire application will be developed in Arabic to ensure accessibility for Arabic-speaking users**:**

1. **Translation:**
   - Translate all static text and user interface elements into Arabic.
   - Ensure right-to-left (RTL) text alignment and layout adjustments.
2. **Localization Testing:**
   - Test the application thoroughly to ensure all Arabic text is correctly displayed and functional.

5. Project Plan and Management

Project Phases

1. **Initiation and Planning:**
    - Define project scope, objectives, and deliverables.
    - Develop a detailed project plan and timeline.
2. **Data Collection and Preprocessing:**
    - Collect MRI image data and perform preprocessing steps.
    - Ensure data quality and prepare datasets for model training.
3. **Model Development:**
    - Train and validate the UNET model for brain tumor segmentation.
    - Fine-tune the model to achieve optimal performance.
4. **Application Development:**
    - Develop web and mobile applications with distinct interfaces for doctors and patients.
    - Implement all required functionalities and ensure seamless integration.
5. **Testing and Quality Assurance:**
    - Conduct thorough testing of the application to ensure functionality and usability.
    - Perform user acceptance testing with a focus on Arabic language users.
6. **Deployment and Maintenance:**
    - Deploy the application on web and mobile platforms.
    - Provide ongoing support and maintenance to address any issues and implement updates.

6. REFERENCE

1. Image J Documentation :https://imagej.net/ij/docs/index.html

Chapter (2)

Software Requirements Specification (SRS)

Table Of Contents

1. Introduction

Brain tumor analysis and diagnosis are critical components of modern medical practice, requiring precision, accuracy, and timely intervention. Magnetic Resonance Imaging (MRI) is a widely used imaging technique that provides detailed brain images, helping medical professionals identify and assess abnormalities such as tumors. However, the manual analysis of MRI images can be time-consuming and subject to human error, necessitating the development of automated solutions that can assist in detecting and segmenting brain tumors.

This project aims to leverage advanced machine learning techniques, particularly the UNET model, to develop a system that can automate the classification and segmentation of brain tumors in MRI images. By integrating this model into a web and mobile application, the project seeks to enhance the efficiency and accuracy of brain tumor analysis. This system will not only aid doctors in making informed decisions but also empower patients by providing them with easy access to their medical information and history.

The web and mobile applications will be designed with a user-friendly interface, ensuring that both doctors and patients can navigate the system with ease. The application will allow doctors to upload MRI images, automatically classify and segment tumors, and allow doctors to generate detailed reports. Patients will be able to view their reports, manage their medications, and communicate with their doctors. The entire system will be developed in Arabic, catering specifically to Arabic-speaking users and ensuring accessibility and usability for this demographic.

1.1 Purpose

The purpose of this document is to outline the comprehensive approach for developing a system that facilitates brain tumor analysis using MRI images through advanced classification and segmentation techniques. The system will be implemented using a combination of web and mobile applications, designed to streamline the workflow for medical professionals and enhance the accessibility of medical information for patients. The primary goals are to automate the detection and segmentation of brain tumors, enable seamless doctor-patient interaction, and provide an intuitive user experience in Arabic.

1.2 Document Conventions

This document adheres to the following conventions to ensure clarity and consistency:

- This document uses Times New Roman font.
- **Bold text** is used to emphasize important terms and headings with size 22.
- Normal text and subheading with size 13.
- Sections and subsections are numbered for easy reference.
- Visual elements such as diagrams and charts are labeled and referred to by their figure numbers.

1.3 Intended Audience and Reading Suggestions

This document is intended for a diverse audience, including project stakeholders, developers, and end-users. The following reading suggestions are provided to guide different audience segments through the document:

1. **Project Stakeholders:**
    - Focus on sections that outline the project's objectives, methodologies, and expected outcomes.
    - Key sections: Project Abstract, Project Objectives, Project Plan and Management.
2. **Developers:**
    - Pay attention to the technical details in the Approaches and Methodology section, which provides a comprehensive guide to implementing the system.
    - Key sections: Data Preprocessing, UNET Model for Segmentation, Web and Mobile Application Development, Model Deployment.
3. **End-Users (Doctors and Patients):**
    - Review the sections related to the functionalities and user interfaces of the web and mobile applications to understand how the system will enhance their interaction and workflow.
    - Key sections: Project Objectives, Frontend Development, Backend Development, Language Localization.

1.4 Product Scope

The system is designed to facilitate the administration and management of brain tumor services. The admin will have the ability to easily update the database, ensuring that the information remains accurate and up-to-date. Users of the system will have the option to search for and select a specific doctor by using their username. This feature enables users to find and connect with the doctor of their choice for consultation or treatment purposes. To access the system's services, users must register

themselves as either a doctor, patient, or admin. During the registration process, users will be required to provide essential credentials such as their first name, last name, email address, national ID, phone number, date of birth, address, and password.

1.5 References

1.5.1 Standards: IEEE Std 830 IEEE Recommended Practice for Software Requirements Specifications

1.5.2 Python: Python Release Python 3.11.0 | Python.org

1.5.3 Python API: Welcome to Flask — Flask Documentation (3.0.x) (palletsprojects.com)

1.5.4 NodeJS: https://nodejs.org/en/download/package-manager

1.5.5 MongoDB: https://www.mongodb.com/try

2. Overall Description

2.1 Product Perspective

This is a new, self-contained project. Initially, there is no framework or system in place to build upon; therefore, this software will be built from the ground up. This system will consist of a back end and a front end. In the front end, the user will interact directly with a web page using the interface to retrieve all data requested. Additionally, the front end will be used by the users to enable any required permissions required by the electric companies of our application to access the relevant user data. In the backend, our software engine will allow us to aggregate the data, store the data, and produce results that will be of value to the users. The web application will be built on top of NodeJS + Python and interface with express, HTML, CSS, and MongoDB to store and pull relevant data.

2.2 Product Functions

- The user must be able to upload his Data to our website.
- The user must be able to authenticate with our system.
- The product must be able to collect and store data results in the MongoDB database.
- The product must be able to provide calculations and analysis of the user's data.
- Our website helps your doctor detect changes in brain tissues through MRI images, aiding in brain tumor segmentation and classification.

2.3 User Classes and Characteristics

The initial users of our software front end will be doctors and patients. The user may take a moment to familiarize themselves with the layout and interface of our web service but should easily understand what they are looking at.

2.4 Operating Environment

- Our website will primarily act as a web service that will be supported on web browsers including Google Chrome, We assume that the underlying operating system is Windows.

2.5 Design and Implementation Constraints

2.5.1 Design

- Language: Python, NodeJS, HTML, Express, CSS
- Database: MongoDB

2.5.2 Implementation Constraints

- Must run on Google Chrome
- The user must authenticate himself at the login.
- Processing requirement: as little processing as possible

2.6 User Documentation

Everything will be as simple as possible. Ideally, it will require no extra explanation. Every step for account setup and viewing options will be clearly labeled and described as the client progresses through the web page.

2.7 Assumptions and Dependencies

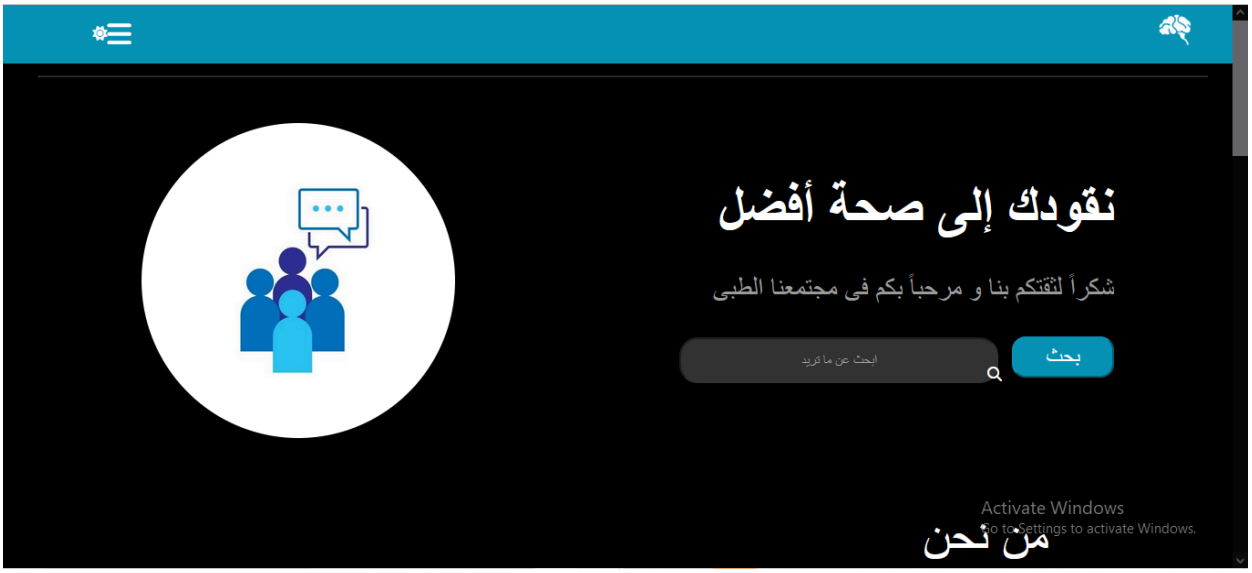
- We assume the user has or has access to our website (login)
- We assume the user has an internet connection to navigate the website.
- We depend on the correct operation between Python + NodeJS+HTML+CSS+JS and MongoDB

3. External Interface Requirements

3.1 User Interface

System Web Application contains the following user interfaces: ▪

- Home page contains the latest news about Brain Tumor, in addition to some scientific articles that the user can view, through which it is possible to go to the rest of the site pages such as the patient and doctor profile pages, about us and contact us.
- Login page for authenticating registered users. This screen should accept username and password. It also provides features for "New user registration" and “Forgot password”.
- Services page contains the services provided by the site to the user, all of which are based on helping radiologists in diagnosing brain tumor diseases.
- The About Us page contains information about the team admin, team members, founders, and site administrators .
- Contact Us page contains contact information through which the user can communicate with us if they encounter any problem



Home Page

3.2 Hardware Interfaces

- Laptop: Dell and HP
- Operating system: Windows 11
- Hard disk: DHH 1TB, SSD 512GB
- RAM: 8GB,16 GB
- Processor: Intel(R) Core (TM) i7-8565U CPU

- Server: Intel(R) Xeon(R) Silver 4214 CPU @ 2.20GHz 2.19 GHz (6 processors), 20 GB RAM

3.3 Software Interfaces

- Framework: NodeJS, Express, HTML,CSS Web Framework
- Python Interpreter: TensorFlow, Microsoft Visual Studio Code Community, and Jupyter Notebook.
- Database: MongoDB

3.4 Communications Interfaces

We use the DICOM standard to retrieve MRI scans from a hospital, preprocess with tools openCV TensorFlow, integrate segmentation and classification models via TensorFlow, manage model training and evaluation using scikit-learn and TensorBoard, provide a user interface developed in React for image upload and result visualization, exposing functionalities through secure RESTful APIs built with Flask, handling data storage with MongoDB.

4. System Features

4.1 Segment Tumor through MRI:

4.1.1 Description and Priority:

This feature is considered the distinctive feature of the Sadek website, as deep learning techniques and neural networks are used to segment the brain tumor by having the doctor upload the MRI image and then the segmentation result appears on the interaction page

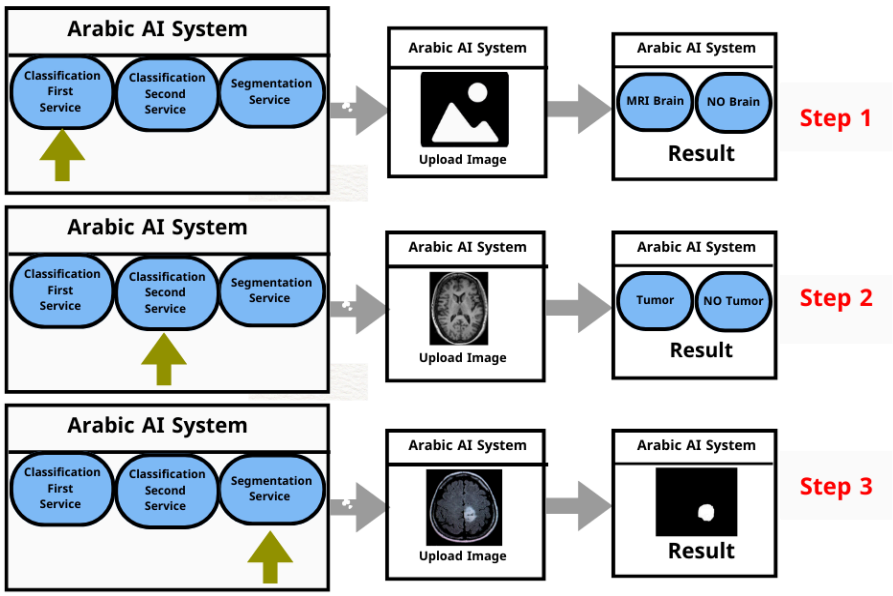
On the other hand, the result of the segmentation and the medications proposed for the patient are kept on his personal page in the application, and remind him of medication appointments

4.1.2 Response Sequences

The process of Segmentation and Classification is carried out through two steps as follows:

1. The doctor uploads images.

2. Deep learning algorithms will be applied to the image to segment and classify tumor.



4.1.3 Functional Requirements

For this feature to be achieved, some requirements must be met:

1. The doctor has already logged in to the site using the correct username and password
2. The doctor has already clicked on the category of uploading the image and replying with a text to
3. The doctor should be able to upload an MRI image and not another type of image.
4. The computer on which the image was uploaded must be connected to the Internet.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

- Clicking any link on the navigation bar should take less than 1 second
- Updating and saving any personal information should take less than 1 second.
- Displaying any results should take less than 15 seconds.
- Uploading a profile picture should take less than 15 seconds.
- Retrieving new and updated energy data should take less than 30 seconds.
- The system should support at least 200 concurrent users.
- The times when the system will be most stressed are likely during assignment submissions. Therefore, it must be able to handle at least 200 concurrent users.
- A substantial number of users must be able to access the system at the same time

5.2 Safety Requirements

- The user's login information should be secure enough so that anyone except the developers will know that user's username/email and password.
- When the user types his or her password to log in, it should be hidden.
- The system will be protected against vulnerabilities such as injection attacks, even though we use MongoDB instead of SQL databases
- If there is extensive damage to a wide portion of the database due to catastrophic failure, such as a disk crash, the recovery method restores a past copy of the database that was backed up to archival storage (typically tape) and reconstructs a more current state by reapplying or redoing the operations of committed transactions from the backed-up log, up to the time of failure

5.3 Security Requirements

- The system's back-end servers shall never display a customer's password. The customer's password may be reset but never shown.
- The system's back-end servers shall only be accessible to authenticated administrators.
- The system's back-end databases shall be encrypted.
- The system should be secure enough to protect against denial-of-service attacks.
- The system should be secure enough to prevent data corruption from unauthorized users.
- The system should be secure enough so that the user's personal information will not be disclosed to unauthorized users.
- The program is protected from hacking and viruses.

5.4 Software Quality Attributes

5.4.1 Availability

- The system must deliver services to the client when requested.
- The system should be always available, meaning the user can access it using a web browser, only restricted by the downtime of the server on which the system runs. In case of a hardware failure or database corruption, a replacement page will be shown. Also, in case of a hardware failure or database corruption, Backups of the MongoDB database should be retrieved by the administrator and saved, ensuring continuity and recovery in case of issues.

5.4.2 Dependability

- The website must be reliable enough so that when different users perform the same task, the expected outcome does not differ.
- The website must be reliable enough so that when users perform normal functions, the system does not fail.

5.4.3 Usability

- The website must be easy enough to learn so that users know how to use the product entirely on their first try.
- The website must be more efficient than similar products.

5.4.4 Flexibility

- The website must be flexible enough to support Google Chrome and Firefox 18 browsers.
- The website must be able to support Windows.
- The website must be flexible enough to support users with different power companies.

5.5 Business Rules

- No one can use the application (without the username and password, the user cannot use the application.)
- The opinion of maintaining and improving the application due to the customer.
- add rules to your form controls that validate the data. For example, an Email field validates that a user has entered a valid email address.

Chapter (3)

Software Design Document

(SDD)

Table Of Contents

Revision Page

a. Overview

This version of the system helps doctors to diagnose diseases such as brain tumor accurately and in a short time from the MRI images

b. Target Audience

Radiologist, Patient, Admin

1. Introduction :

The purpose of this software design system is to provide a low level description of the System providing insights into the structure and design of each component Topics covered include the following

1.1. Purpose

This document provides a detailed description of the software architecture of the system and will serve as a road map showing future developers how the system was designed and implemented. It specifies the structure and design of some of the modules discussed in the Software Requirements Specifications, cited above.

1.2. Scope

Our application was developed using state-of-the-art deep learning techniques and algorithms to diagnose 2D axial MRI images.

The primary focus of our project is to accurately diagnose a sequence of 512 patient images, including the image of the brain from the top to the neck with every detail of the brain, and more. Our team of experts trained our algorithm using a large dataset of MRI images, making our application highly effective at identifying patterns in these images that may indicate a particular condition.

By leveraging the power of deep learning, our application provides fast, accurate diagnoses that are essential for effective treatment and management of Brain tumor diseases.

1.3. Goals and Objectives

The purpose of our system is to help doctors to diagnose Tumors accurately and in a short time. Thus, the primary objective of the project is to create a model that can automatically classify and segment tumors.

Accordingly, the final product must be quick, efficient, and extremely easy to use. It must offer useful features without overwhelming the user with options. The user interface must be intuitive and have little or no learning curve. Beyond these general design principles, the application must also provide the following concrete functionalities:

Algorithm to determine the diagnosis.
Support for multiple transactions.
Automatic data synchronization
Detecting history.

1.4. Definitions

ERD	**The entity-relationship diagram**
CRUD	**Creat, Read, Update, Delete**
ORM	**Object-Relational Mapping**

1.5. Reference

NodeJS & Express & MangoDB: Udemy Course
Website Design: Youtube Link

1.6. Overview

Firstly

a. We spoke about the introduction of the project to describe what the project does and how it will work.
b. We spoke about the purpose of the project and what are the benefits from it.
c. We spoke about the scope that defines the classes and functions of the application and what is each function definition.
d. We spoke about the goals and objectives of the project and what are the benefits from it.
e. We spoke about definitions that define what the system has that acts as the radiologist and patient roles in the project.
f. We spoke about references that define what sources we use to initiate this website from scratch, courses to learn NodeJS & Express & MangoDB & learn how to design, and channels to teach us how to code.

Secondly

a. We spoke about the architectural style that is used while designing the Software and why we use this architecture.
b. We spoke about the architectural models that are used in our system and the relation between the components.
c. We spoke about the use case diagram; we explained how the system will work exactly between radiologist, patients and the Database.

Thirdly

a. We added a subsystem diagram for our project
b. In the Detailed Description, we gave two examples for our modules, which are and .

Fourthly

a. We added our project's schema.
b. We described the database sections and every table on it and its Schema Content.
c. We describe how the major data or system entities are stored, processed, and organized.

Fifthly

a. We spoke about our GUI and what it is consisting of and gave screenshots for each.
b. We describe the functionality of the system from the user’s perspective.

2. System Architectural Design:

2.1. Architecture Style and Rationale:

The proposed architectural style for the system is a client-server architecture with a layered approach. The system will consist of two main components: a web-based user interface for radiologists to upload images and view diagnosis results, and a server-side application responsible for processing image data and generating diagnosis reports.
The layered approach will provide several benefits, including improved scalability, maintainability, and flexibility. By separating the presentation, application logic, and data storage layers, developers can modify individual components without affecting the rest of the system. Additionally, this approach allows for the use of standardized protocols and interfaces, which can simplify integration with other systems in the future.

2.2. Architecture Model

The component model for the system will consist of the following components

2.2.1. Web User Interface:

Responsible for providing the user interface to radiologists.
Allow Radiologists to Upload images, view diagnose results, and Chat with Patient
Allow Admin to access Database. o Built using HTML, CSS, and JavaScript.

2.2.2. Image Segmentation Service:

Responsible for processing image data and generating diagnosis results.
Uses deep learning algorithms to analyze images and identify tumor's location.
Built using Python (TensorFlow, keras, opencv python, pillow, numpy)

2.2.3. Image Classification Service:

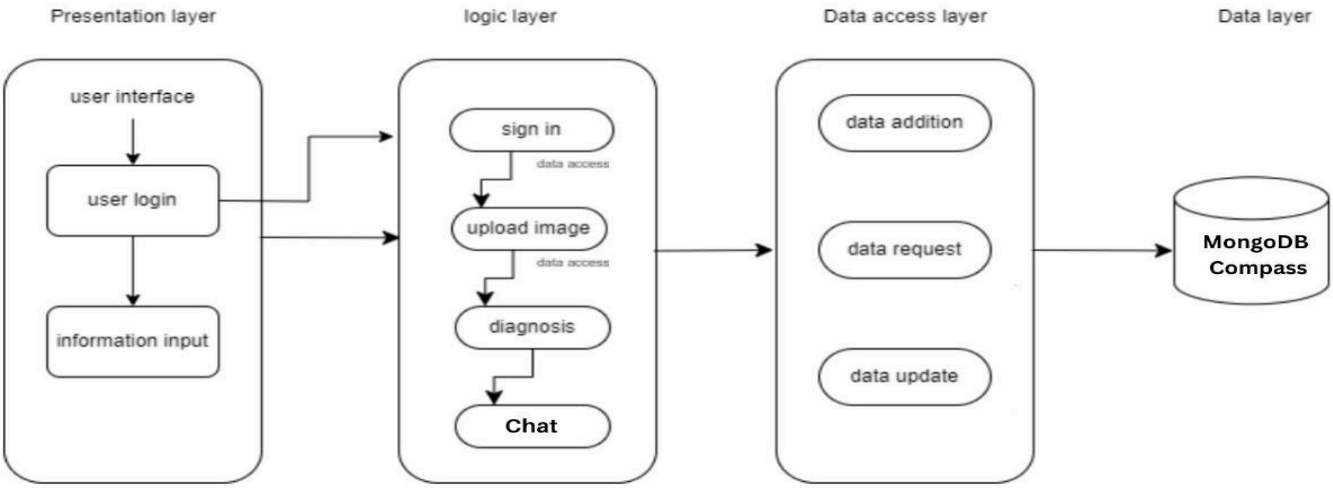
Responsible for classify images to a tumor and no tumor classes
Use deep learning algorithms to analyze images and classify tumor existence
Built using python (RESNET-101, TensorFlow numpy)

2.2.4. Chat Service:

Responsible for sending messages between Doctors and patients.
Built Using Node JS and Web socket (Socket.io).

2.2.5. Database Service:

Responsible for storing, and retrieving data from the database, and it provides a way to (CRUD) create, read, update, and delete data stored in a relational database.
Utilizes a relational database management system (MongoDB),3-Built using MongoDB Compass

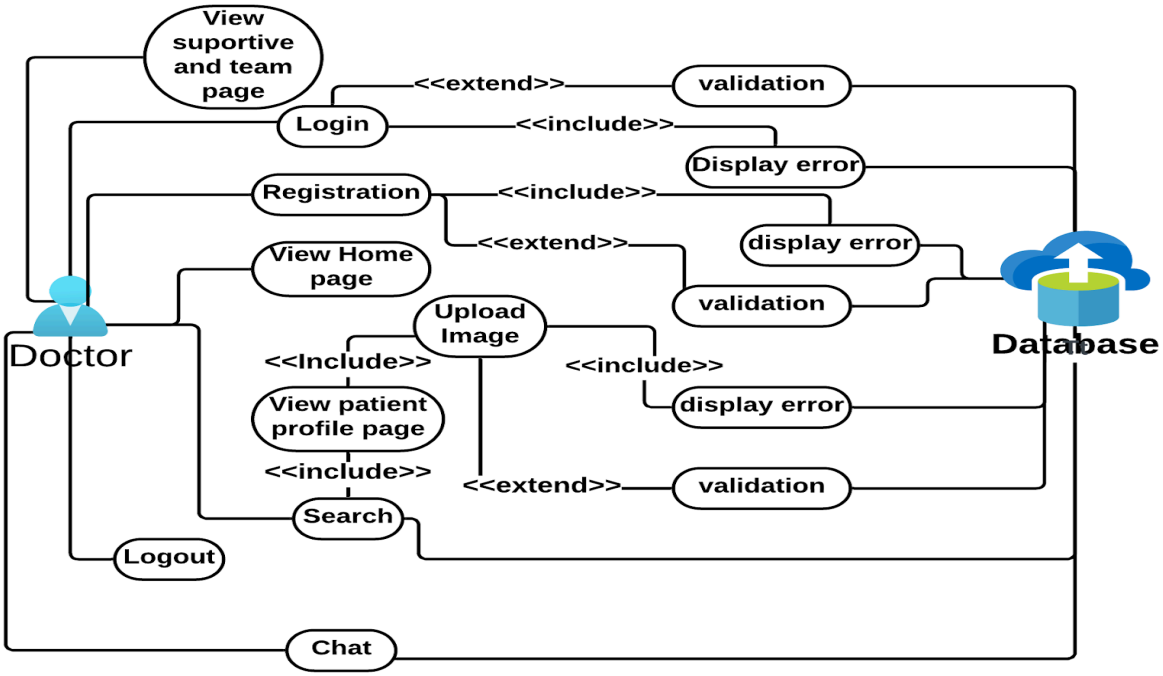


Component Model for the system

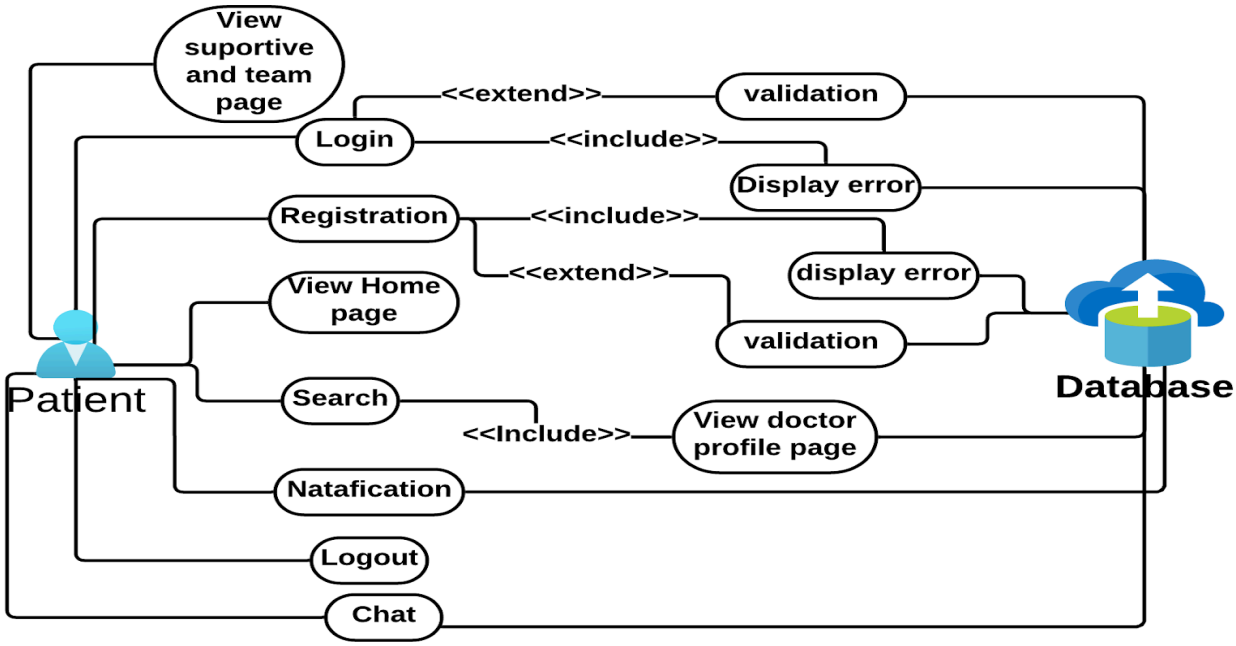
2.3. Use Case Diagram:

When a Radiologist uploads an image, the Web User Interface component sends the image data to the Image segmentation and classification Service for analysis. The web user interface then gives access to the doctor a diagnosis report and stores it in the database.

Use Case Diagram for the Radiologist

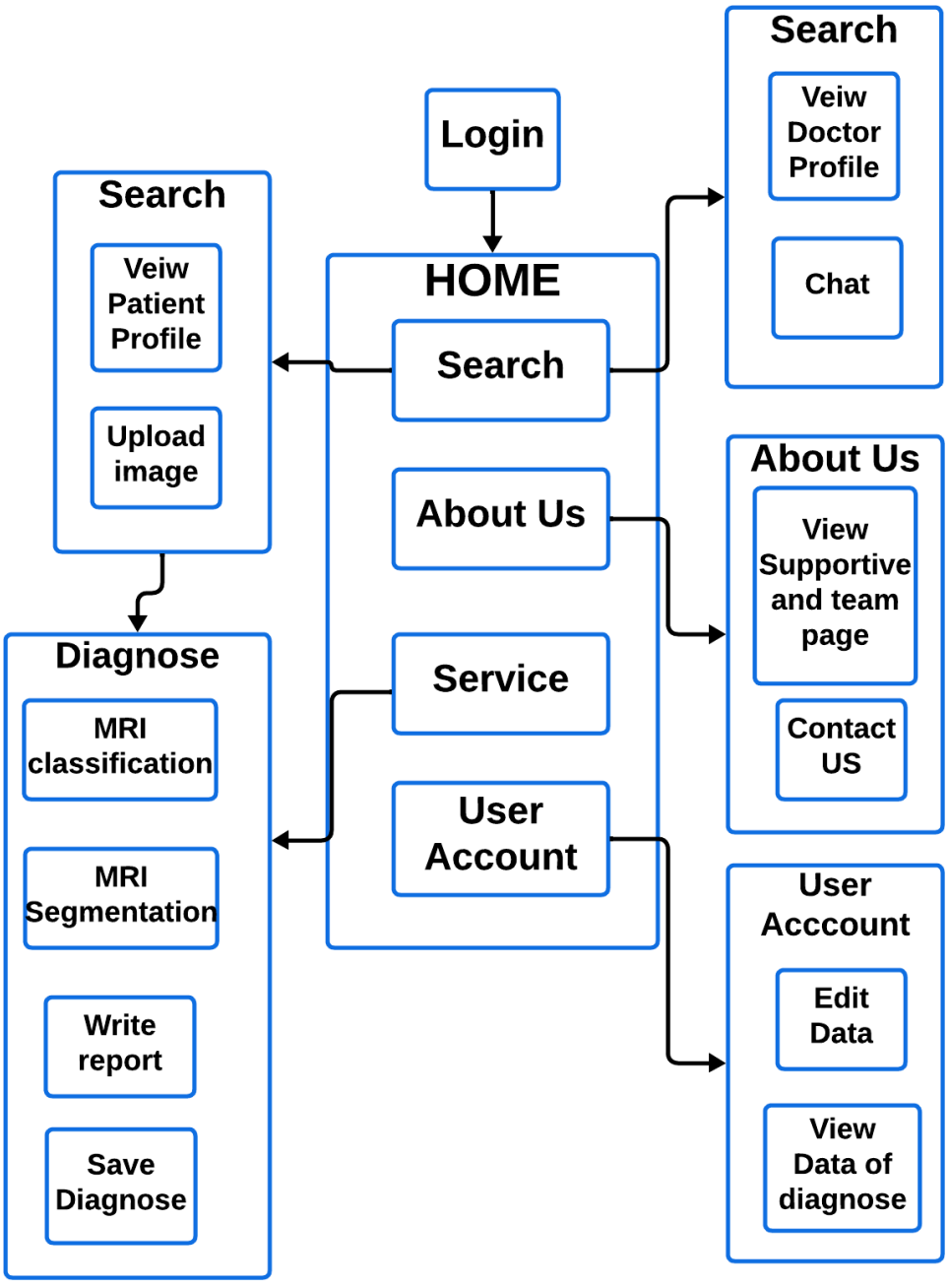


Use Case Diagram for the patient



3. Detailed Description of Components

3.1. Complete Package Diagram



Subsystem of the Website

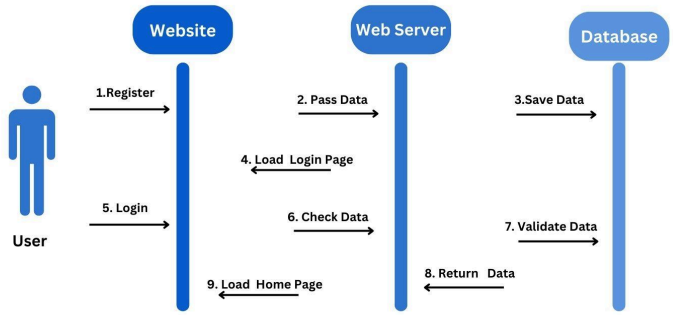
3.2. Detailed Description:

3.2.1. Model <Register>:

The Target audience is the Doctor and the patient

The user enters his basic information with an attribute of a check box that the user chose if he is doctor or patient, next attributes depend on this attribute

3.2.1.1. Sequence Diagrams

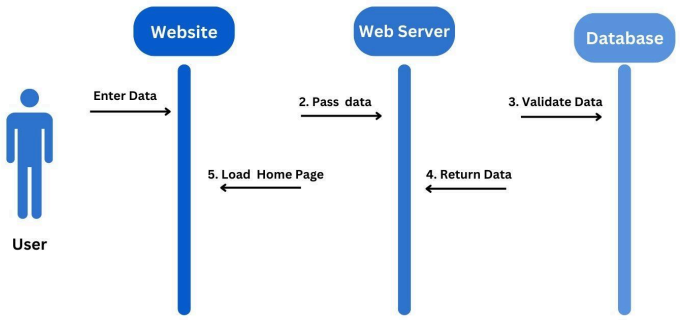


SEQUENCE DIAGRAM OF REGISTRATION NEW USER SCENARIO

3.2.2. Model <LOGIN>:

The Target audience is the Doctor, the patient, and the admin

3.2.2.1. Sequence Diagrams

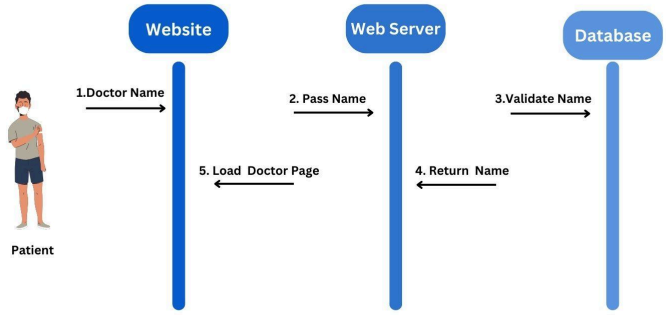


SEQUENCE DIAGRAM OF LOGIN USER SCENARIO

3.2.3. Model <Search for doctor>:

The Target audience is the patient

3.2.3.1. Sequence Diagrams:

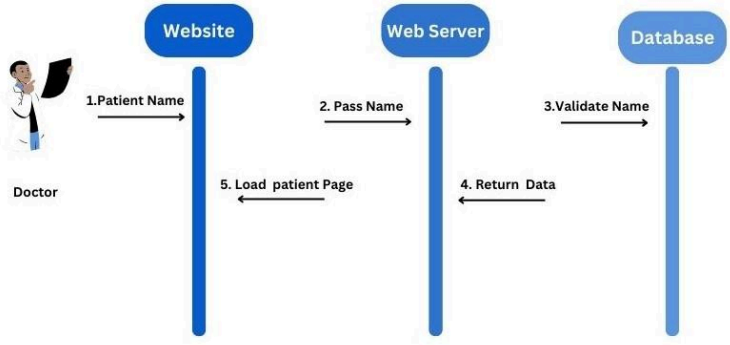


SEQUENCE DIAGRAM OF SEARCHING ON DOCTOR FOR PATIENT SCENARIO

3.2.4. Model <Search for Patient>:

- The Target audience is the doctor

3.2.4.1. Sequence Diagrams:

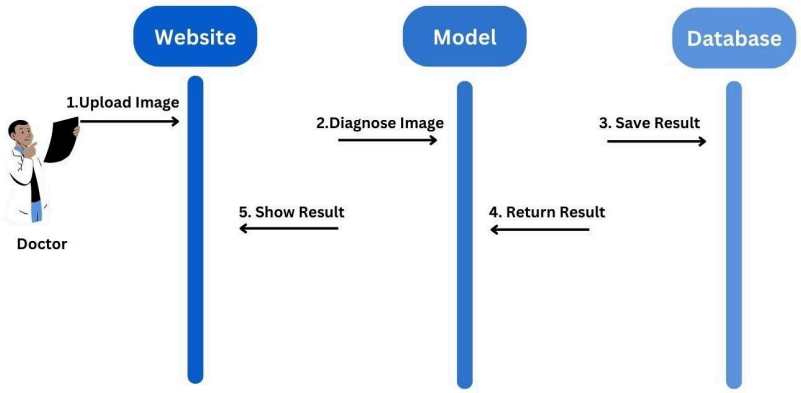


SEQUENCE DIAGRAM OF SEARCHING ON PATIENT FOR DOCTOR SCENARS

3.2.5. Model <Upload>:

- The Target audience is the doctor

3.2.5.1. Sequence Diagrams:

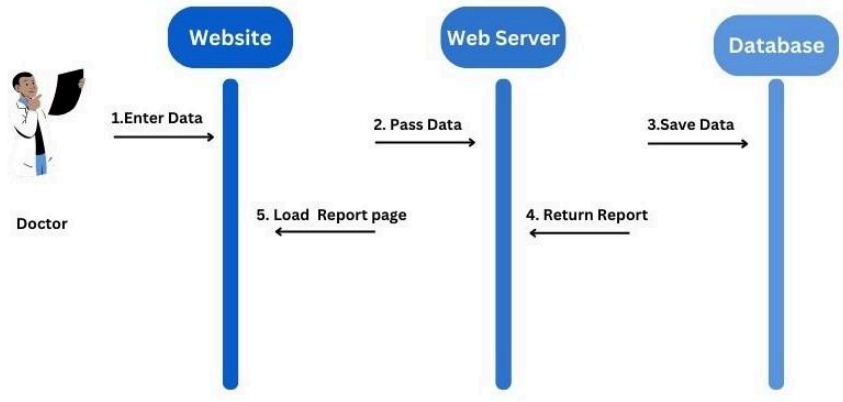


SEQUENCE DIAGRAM OF UPLOADING IMAGE SCENARIO

3.2.6. Model <Add Details>:

- The Target audience is the doctor

3.2.6.1. Sequence Diagrams:



SEQUENCE DIAGRAM OF ENTERING PATIENT'S SCENARIO

4. Data Design:

4.1. Data Description:

The data for this project will be stored in a non-relational database using **MongoDB** and accessed through **Nodejs**. The database design for this website consists of three schemas with a one-to-one relationship between them**.** This schema will store information about patients , doctors,and their diagnostic operations For brain tumors. as well as the associations between doctors, patients , and operations

4.2. Schema:

4.2.1. User Schema:

Field name	*Data type*	*Constraints*
Id	ObjectId	Primary key ,unique
First-name	String	required
Last-name	String	required
address	String	required
email	String	unique
Phone_number	Number	required
gender	String(enum)	required
password	String	required
password_Confirm	String	required
ssn	Number	required(14number)
date-of-birth	Date	required
image	String	required
role	String	Required(admin,patient, doctor)

4.2.2. Doctor Scheme:

Field name	Data Type	*Constraints*
user	ObjectId	ref(user)
specialty	String	required
hospital-name	String	required

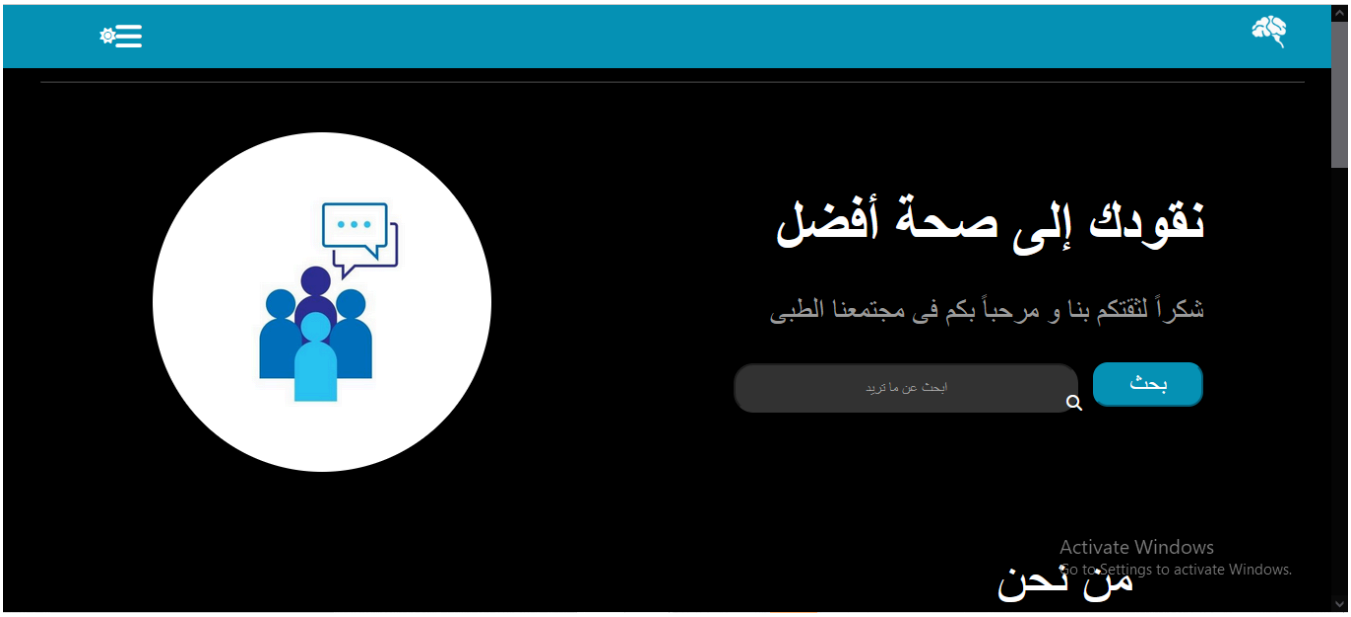
4.2.3. Patient Schema:

Field name	Data Type	*Constraints*
user	ObjectId	ref(user)
history_type	String	required
responsible_doctor	String	required
initial_diagnosis	String	required
complain	String	required
medical_test	String	required
rays	String	required
treatment	String	required
consult_another_doctor	String(enum)	required
smoking	String(enum)	required
allergy	String(enum)	required
infection	String(enum)	required
diagnosis_result	String	required
final_report	String	required
comments	String	required
segmented_image	String	required

1. USER INTERFACE DESIGN

The Website Interface Design

1.1. When a user opens the site, the user will be shown the HOME PAGE first if the user want to Sign In or Register he will find it above as a button:



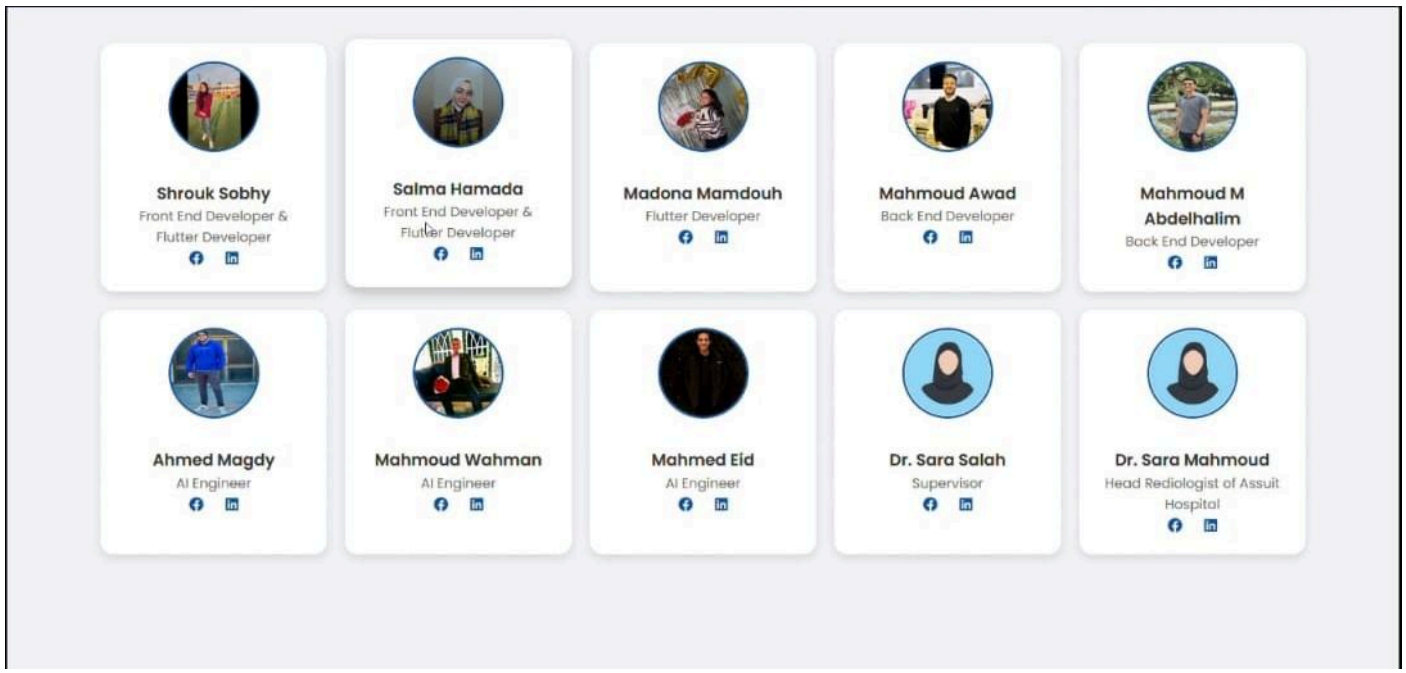
1.2. When the user wants to subscribe to the webSite . he or she will be shown the ‘Sign in ’page first:

If he was previously registered on Site ,he will enter the email and password.

1.3. If the user is new, there are some data to be entered for registration:

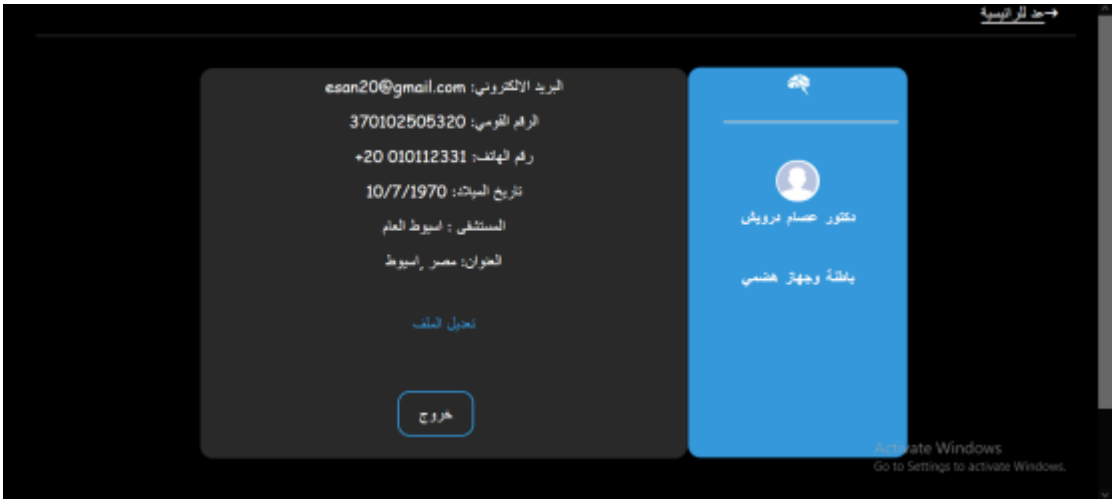
1.4. Password reset page. We write the new password and confirm it:

1.5. On the "About Us" page, you will find information about the supervisors and contact information for the team the project:



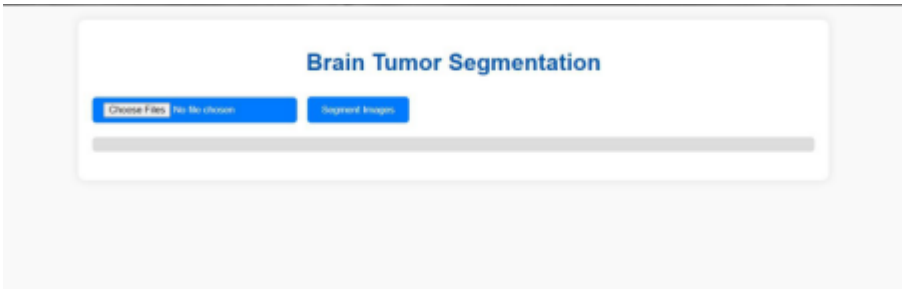
1.6. On the Help page, the user can contact us if he has any problem

1.7. On the Profile page, the User can edit his email, change the password and see the history:

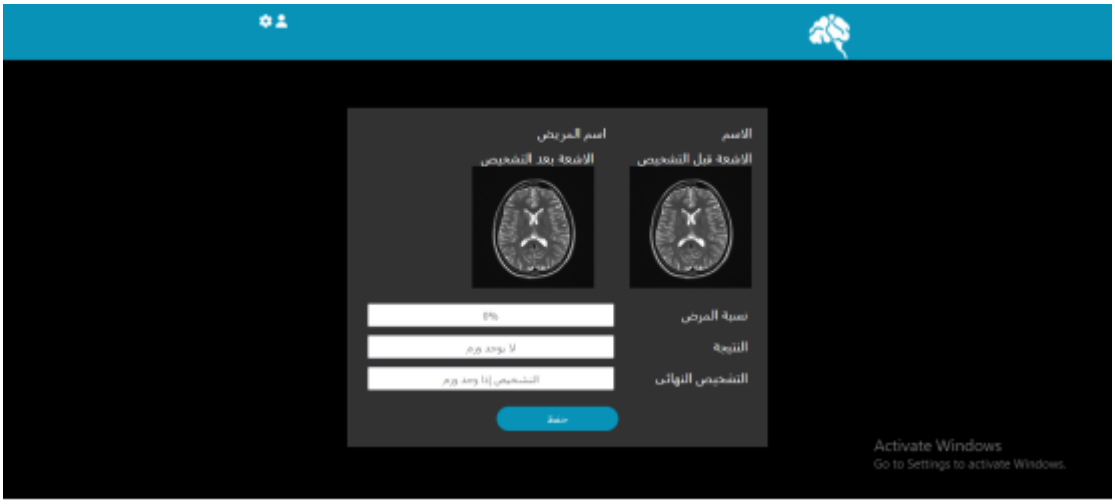


1.8. **For the Diagnosing Process after Doctor searches for the patient he can take the information about patient complain before the diagnosing then start the diagnosing serves:**

1.9. **For the Uploading Images, the doctor upload the patient sequence of images that is already taken from MRI Device, the sequence is from Axial Type in 2D array about 512 image:**



1.10. **After completing the Process, doctor can write the report and save:**



The Application Interface Design

1.11. When a user opens the Application:

1.12. When the user wants to subscribe to our application ,he or she will be shown the ‘Sign in ’page first and he must choose if he a doctor or patients , this page if he a patient

1.13. If the Registrator is a doctor, he will have a different data to be entered with the general data:

1.14. If he was previously registered on App, he will enter the username and password:

1.15. If he forget the password:

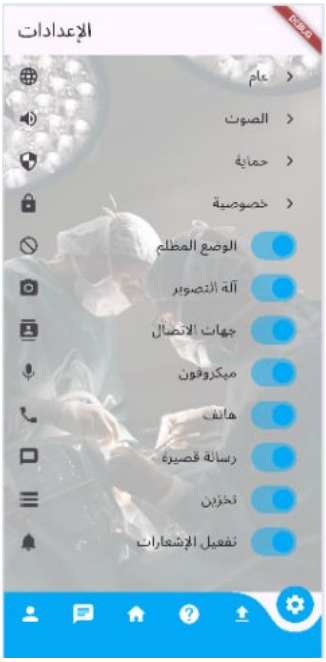
1.16. If the user want to contact with doctor(Searching Process):

1.17. If the user wants to see his profile and modify it:

1.18. If the patient wants to chat with doctors to ask him some question:

1.19. The page to help the user to use our application:

1.20. To control the setting of the Application



1.21.

1.22.

References

- MOHAMMAD ASHRAF OTTOM,HANIF ABDUL RAHMAN,AND IVO D. DINOV,"Znet: Deep Learning Approach for 2D MRI BrainTumor Segmentation",p1-6,May2022.
- Abhishta Bhandari, Jarrad Koppen & Marc Agrarian,"Convolutional neural Network for brain Tumor Segmentation",P1_8, 8 June 2020.
- Akmal Bek bobomirza evich abdusalomov,"Brain Tumor Detection Based On Deep Learning Approaches and Magnetic Resonance Imaging",18 Aug 2023.
- Yasmina Al Khalila, Aymen Ayaza, Cristian Lorenz, Jürgen Weeseb, Josien Pluima, Marcel Breeuwer,"Multi-modal brain tumor segmentation via conditional synthesis with Fourier domain adaptation",P1-10 ,11 January 2024.
- Faizan Ullah , Muhammad Nadeem , Mohammad Abrar , MunaAl-Razgan ,Farhan Amin ,and AbduSalam , Taha Alfakih ,"Brain Tumor Segmentation from MRI Images Using Handcrafted Convolutional Neural Network"،P1-13,11 August 2023.
- KhTohidul Islam , Sudanthi Wijewickrema and Stephen O'Leary,"KADeepLearning Framework for Segmenting Brain Tumors Using MRIandSynthetically Generated CT Images",P1-10,11 January 2022.
- YahyaM.A.Mohammed1,*Said El Garouani2 and Ismail Jellouli,"A survey of methods for brain tumor segmentation-based MRI images",P266_ 286,11 January 2023.
- Ramin Ranjbarzadeh, Abbas Bagherian Kasgari, Saeid Jafarzadeh Ghoushchi, Shokofeh Anari, Maryam Naseri & Malika Bendechache , "Brain tumor segmentation based on deep learning and an attention mechanism using MRI multi-modalities brain images",25 May 2021.

- Taranjit kaur& Tapan Kumar Gandhi,"Deep convolutional neural networks with transfer learning for automated brain image classification",P1-16,27 March 2020