CS492: Senior Design Project II

Design Project Final Report

MRacle

Mirza Özgür Atalar 22003455
Orhun Aysan 22103005
Kaan Türkoğlu 22102105
Ege Karaahmetoğlu 22102622

Advisor: Doruk Öner

Course Instructors: Atakan Erdem, Mert Bıçakçı

Table of Contents

1. Introduction

We have identified critical shortcomings within the current healthcare system that pose challenges for both individuals and institutions. In particular, radiology departments are overwhelmed by the high volume of medical imaging data, especially MRI scans, while facing a shortage of radiologists. This imbalance leads to diagnostic delays, increased workload, and a higher risk of human error. As patient populations grow and brain-related diseases become more significant, existing systems struggle to meet demand, impacting healthcare quality and patient outcomes [1].

Technological advances in artificial intelligence and medical imaging provide an opportunity to address these challenges. Yet, despite the availability of high-performing AI models, their integration into clinical practice remains limited due to concerns about transparency, accountability, and the difficulty of seamlessly incorporating AI into the medical workflow [2][3]. MRacle seeks to address these issues by introducing a human-centered AI support tool that is suitable for clinical environments.

Our project focuses on brain tumor detection, a domain in which early and accurate diagnosis is vital. Brain tumors often present complex visual patterns on MRI scans, requiring expert interpretation across multiple sequences [1]. However, with limited time per scan and high diagnostic burden, even experienced radiologists can face challenges in maintaining consistent performance.

MRacle is an end-to-end AI-assisted platform designed to support radiologists in the detection and segmentation of brain tumors from MRI scans. MRacle highlights regions of interest on the scan, assigns likelihood scores, and assists in prioritizing high-risk cases. Rather than replacing radiologists, MRacle enhances their capabilities by serving as a second reader and enabling more efficient resource allocation [2].

In MRacle's workflow, MRI scans are automatically retrieved from the hospital's PACS system and analyzed by our AI model. The system then visualizes the segmentation overlays and tumor scores, which the radiologist reviews alongside patient data. If the AI and the radiologist's assessment align, the diagnosis can be quickly confirmed. If discrepancies arise, the radiologist is equipped with explainable model outputs to guide further investigation, ensuring both transparency and confidence in the decision-making process [2][3].

By offering rapid, reliable, and interpretable assistance, MRacle provides a scalable solution to a pressing healthcare challenge and aims to make brain tumor diagnosis more efficient, accurate, and accessible [1].

2. Requirements Analysis

This section explains detailed information about how we built MRacle. In general, our design is based on two main features of our application: time-saving for radiologists and extended accuracy.

2.1 Functional Requirements

- The system supports files in various formats (NIfTI [4], DICOM [5]) and multiple MRI sequences (T1-CE, T2-FLAIR).
- The system allows synchronized navigation through different MRI sequences for comprehensive analysis.
- The system can initiate AI-driven analysis on selected scans and view segmentation overlays.
- After finishing the analysis, the system evaluates potential tumor risks highlighted by the AI model according to their severity.
- The system analyzes changes in tumor size and characteristics by comparing current scans with previous ones.
- Radiologists are able to add corrections or confirmations to AI results and submit feedback to refine AI performance.
- The system only keeps essential patient data following anonymity rules and privacy rules.
- Radiologists are presented with statistics and information on the number of patients and scans remaining and completed.
- Radiologists can view patients with relative information such as age, gender, the MR scan date, and the results gathered from the analysis.
- The system lists the patient scans according to the disease severity, where the most severe cases are listed at the highest level.

2.2 Non-Functional Requirements

2.2.1 Usability

The system features an intuitive interface designed to align seamlessly with radiologists' workflows, creating a consistent and user-friendly experience. Tasks are implemented to minimize radiologists' efforts, allowing them to focus on their core responsibilities. The application is responsive and can interact with the user when something is done successfully or goes wrong.

2.2.2 Reliability

MRacle has consistent performance and accuracy in its AI analyses. The system has high uptime, with reliable error-handling mechanisms to prevent and recover from failures. Regular maintenance and updates are conducted to maintain system reliability and address any emerging issues promptly.

2.2.3 Performance

MRacle is designed to deliver high performance, completing AI analyses within a short interval per scan to ensure timely support for radiologists. The system will maintain user interface (UI) response times of under one second for interactions.

2.2.4 Scalability

The MRacle system scales to support simultaneous use. The system can handle a growing number of users and an increasing amount of data efficiently. It is able to accommodate the increase in patients and data. This growth is managed without sacrificing the speed or functionality that users rely on, ensuring the system remains fast and reliable as it expands. The application can be containerized to facilitate easy deployment across multiple servers, ensuring load distribution and service availability even as demand fluctuates.

2.2.5 Privacy and Security

The system is designed to focus on the sensitivity of medical data, ensuring compliance with healthcare-specific data protection standards and regulations. Given the critical nature of patient information, decisive measures are used to protect privacy and confidentiality. Data is protected during transmission and storage, minimizing interception or unauthorized access risks.

3. Final Architecture and Design Details

3.1 Overview

MRacle is designed to enhance brain tumor detection by leveraging artificial intelligence to support radiologists in clinical workflows. The system integrates with existing Picture Archiving and Communication Systems (PACS) to acquire MRI scans and applies a deep learning model to detect, segment, and highlight potential tumor regions [6]. By prioritizing cases based on the severity of findings, MRacle ensures that high-risk patients are reviewed first, helping radiologists allocate their time and expertise effectively.

At its core, MRacle utilizes an advanced U-Net based model architecture for automated analysis of brain MRI scans, delivering visual and quantitative insights that assist in identifying critical areas. The case prioritization mechanism is tailored to clinical urgency, streamlining decision-making in high-volume diagnostic environments.

Designed with interoperability in mind, MRacle requires minimal modifications to existing infrastructure and offers a user-friendly interface for reviewing AI-generated results, uploading new scans, and managing patient records.

3.2 Subsystem Decomposition

The MRacle system architecture comprises four principal layers, each containing specialized modules with clearly defined interfaces:

1. **UI Layer**: Implements client-side interfaces for system interaction
2. **Controller Layer**: Manages HTTP request processing and routing
3. **Service Layer**: Implements domain logic and workflow orchestration
4. **Repository Layer**: Manages data persistence operations

The architecture integrates with external PACS infrastructure for medical image acquisition and utilizes MongoDB for data access.

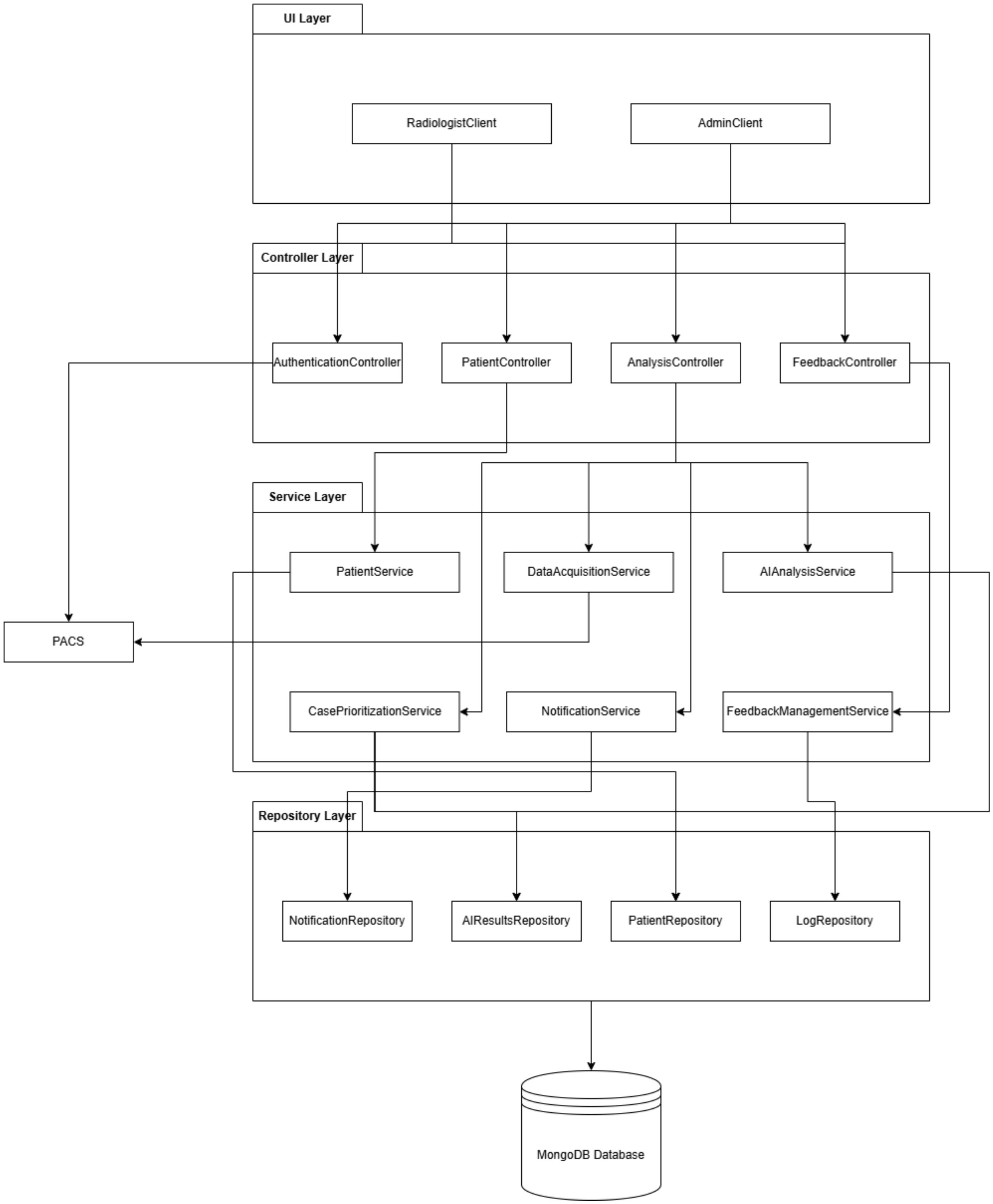

Figure 1: Subsystem Decomposition Diagram

3.3 Hardware/Software Mapping

MRacle requires specific hardware components to ensure optimal performance and integration with hospital infrastructure. The computational needs center around GPU-based servers for AI model inference in tumor detection and segmentation. These inference servers can be deployed on-premises or in secure cloud environments that comply with healthcare regulations. Standard application servers host the backend system's containerized microservices, while a dedicated database server supports MongoDB operations.

The system integrates with various hospital equipment, including MRI machines, which generate standard DICOM-compliant images. Connection with hospital PACS infrastructure occurs through established DICOM protocols, utilizing the existing hospital network.

The deployment follows a distributed approach where MRI machines connect to the hospital PACS servers through internal networks. MRacle inference servers then connect to PACS through approved integration points, with application servers interfacing with inference servers through secure APIs. Radiologists and administrators access the application through secure web interfaces.

3.4 Persistent Data Management

The system employs MongoDB as its primary database. The database architecture includes horizontal partitioning based on patient identifiers to maintain performance with large datasets.

The collection design separates data into logical categories: patient information stores clinical history; MRI scan metadata references image storage locations; analysis results contain AI inference outputs, including tumor likelihood scores; and segmentation data maintains volumetric information. Additional collections manage system notifications and radiologist feedback. Data security measures include encryption for stored information, with particular attention to sensitive patient identifiers.

3.4 Scenarios

Scenario 1: Logging Into MRacle
Actor: Radiologist
Entry Condition:

- The radiologist is registered in the hospital's PACS system.
- The radiologist has valid credentials (e.g., username and password) synchronized with the hospital's PACS.

Exit Condition:

- The radiologist successfully logs into the MRacle system and gains access to their personalized dashboard.
- Authentication logs are updated to reflect the successful login.

Flow of Events:

1. The radiologist navigates to the MRacle login page from their workstation or integrated hospital system.
2. The radiologist inputs their username and password, which are verified against the hospital's PACS.
3. Upon successful verification, the radiologist is redirected to their personalized MRacle dashboard, displaying relevant tools and information.
4. If credentials are incorrect, the system displays an error message and prompts the radiologist to retry.
5. The system initiates a secure session for the radiologist, ensuring encrypted data transmission and session timeout settings for security.

Scenario 2: Automatic Processing and Prioritization of MRI Scans
Actor: MRacle AI System
Entry Condition:

- New MRI scans are uploaded to the MRacle system from the PACS.
- The scans are in supported formats (e.g., NIfTI [4], DICOM [5]) and pass initial validation checks.

Exit Condition:

- The MRacle Model analyzes uploaded MRI scans.
- Each scan is assigned a priority level based on tumor likelihood and stored in the MRacle database.
- Radiologists are notified of the newly prioritized scans.

Flow of Events:

1. The MRI scan (either the latest from the MRI machine or requested from the PACS by the radiologist) is uploaded to the MRacle system.
2. The system validates the format, quality, and completeness of each scan.
3. Invalid scans are flagged for review, and radiologists are notified of any issues.

4. The MRI scan is queued for analysis.
5. The scans are fed into the MRacle Model.
6. The model performs tumor detection and segmentation on each scan.
7. The Case Prioritization System evaluates the tumor likelihood scores generated by the AI model.
8. Scans are ranked from highest to lowest priority based on the assessed risk.
9. The AI-generated results, including segmentation overlays and priority scores, are stored in the MRacle database.
10. Radiologists and the doctor who requested the MRI scan receive notifications about new prioritized scans via the Notifications Page.
11. The prioritized list is updated in real time, ensuring radiologists can access the latest analysis results.

Scenario 3: Viewing AI-Generated Analysis Results
Actor: Radiologist
Entry Condition:
- MRI scans have been successfully analyzed using the MRacle Model.
- The radiologist is authenticated and has access to view analysis results.

Exit Condition:
- The radiologist successfully views and reviews the AI-generated tumor segmentation and likelihood scores.
- Radiologists connect to the PACS and perform the necessary procedures based on the results.

Flow of Events:
1. The radiologist logs into the MRacle system and navigates to the "Analysis Results" dashboard.
2. The system displays a list of analyzed MRI scans, highlighting those with high tumor likelihood.
3. The radiologist selects a specific MRI scan from the list to view detailed results.
4. The system presents the AI-generated tumor segmentation overlays on the MRI images.
5. The system displays the tumor likelihood score and any identified risk factors.
6. The radiologist examines the segmentation accuracy and reviews the highlighted areas.
7. For verification, the radiologist can zoom, pan, and compare the AI results with the original MRI images.
8. The radiologist can go to the PACS and perform the related procedures.

Scenario 4: Receiving Notifications About Prioritization
Actor: Radiologist, the doctor who requested the MRI scan.
Entry Condition:
- MRI scans have been processed and prioritized by the MRacle system.
- The radiologist is authenticated and has access to the notifications feature.

Exit Condition:
- The radiologist successfully receives and reviews notifications regarding newly prioritized scans.

Flow of Events:
1. The radiologist and the doctor who requested the MRI scan configure notification preferences, such as notification types (e.g., high-priority scans) and delivery methods (e.g., in-app, email).
2. As new MRI scans are processed and prioritized, the Case Prioritization System identifies scans that meet the user's notification criteria.
3. The system generates notifications for each relevant, prioritized scan, including key details like patient ID and priority level.
4. Notifications are delivered to the radiologist and the doctor who requested the MRI scan through the selected channels.
5. The radiologist accesses the "Notifications" page to view a list of recent alerts.
6. Upon receiving a notification, the radiologist can directly navigate to the corresponding MRI scan for immediate review and analysis.
7. Upon receiving a notification, the doctor can take immediate action.

Scenario 5: Monitoring Patient Past Records
Actor: Radiologist
Entry Condition:
- The radiologist is authenticated and logged into the MRacle system.
- The radiologist has access rights to view patient records within their scope of practice.

Exit Condition:
- The radiologist successfully views and navigates through a patient's past MRI scans and associated reports.

Flow of Events:
1. The radiologist selects the "Patient Records" option from the menu.
2. The radiologist enters the patient's identifier (e.g., patient ID, name, or date of birth) into the search bar.
3. The system displays a list of matching patients. The radiologist selects the appropriate patient profile from the results.
4. The system retrieves and displays the patient's past MRI scans, reports, and any AI-generated analyses stored in MRacle.
5. The radiologist can download the MRacle analysis and segmentation results.

6. The radiologist can also view the patient's old MRI images in the PACS and request MRacle analysis for them.
7. The radiologist utilizes filters (e.g., date range, scan type) and sorting options to navigate through the patient's historical data efficiently.
8. The system logs the access and viewing of patient records to ensure compliance with data protection regulations.

Scenario 6: Providing Feedback to Improve AI Performance
Actor: Radiologist
Entry Condition:
- The radiologist has reviewed AI-generated results and identified areas for improvement.
- The radiologist is authenticated and has permission to submit feedback.

Exit Condition:
- Feedback has been successfully submitted and logged into the AI Results Dataset Module.
- The system acknowledges receipt of the feedback for further model refinement.

Flow of Events:
1. After reviewing the AI analysis results, the radiologist identifies discrepancies or areas where the AI model's segmentation could be improved.
2. The radiologist selects the "Annotate Results" option on the MRI scan.
3. The radiologist makes corrections or confirms the AI-generated segmentation overlays.
4. The radiologist enters additional comments or suggestions to provide context for the feedback.
5. The radiologist submits the feedback by clicking the "Submit Feedback" button.
6. The system records the feedback in the MRacle backend and associates it with the corresponding MRI scan.
7. The system sends a confirmation message to the radiologist, acknowledging receipt of the feedback.
8. The feedback is utilized in future model training cycles to enhance AI accuracy and performance.

Scenario 7: Viewing Personal Statistics
Actor: Radiologist
Entry Condition:
- The radiologist is authenticated and logged into the MRacle system.
- The radiologist has accessed the personal statistics feature.

Exit Condition:
- The radiologist successfully views and interprets their personal performance and usage statistics.

Flow of Events:

1. The radiologist selects the “Personal Statistics” option from the main dashboard.
2. The system aggregates relevant data, including the number of scans reviewed, response times, and feedback provided.
3. The system displays the statistics intuitively, utilizing charts, graphs, and summary tables.

Scenario 8: Maintaining System and Monitoring System Logs

Actor: System Administrator

Entry Condition:

- The system administrator is authenticated and logged into the MRacle system with administrative privileges.

Exit Condition:

- The system administrator successfully completes maintenance tasks and reviews system logs.
- Any identified issues are addressed, and relevant actions are documented within the system.

Flow of Events:

1. The system administrator logs into the MRacle system and can only access the "Admin Dashboard."
2. The administrator selects "System Maintenance" and executes necessary tasks such as applying software updates and checking performance reports.
3. The administrator accesses the "System Logs" section to review logs related to user activities, system errors, and performance metrics.
4. Upon detecting any anomalies or errors in the logs, the administrator initiates troubleshooting procedures, including restarting services, resolving software conflicts, or enhancing security measures.
5. The administrator records all maintenance activities and resolutions within the system for audit purposes and generates reports summarizing system health and any actions taken.

3.5 Use Case Model

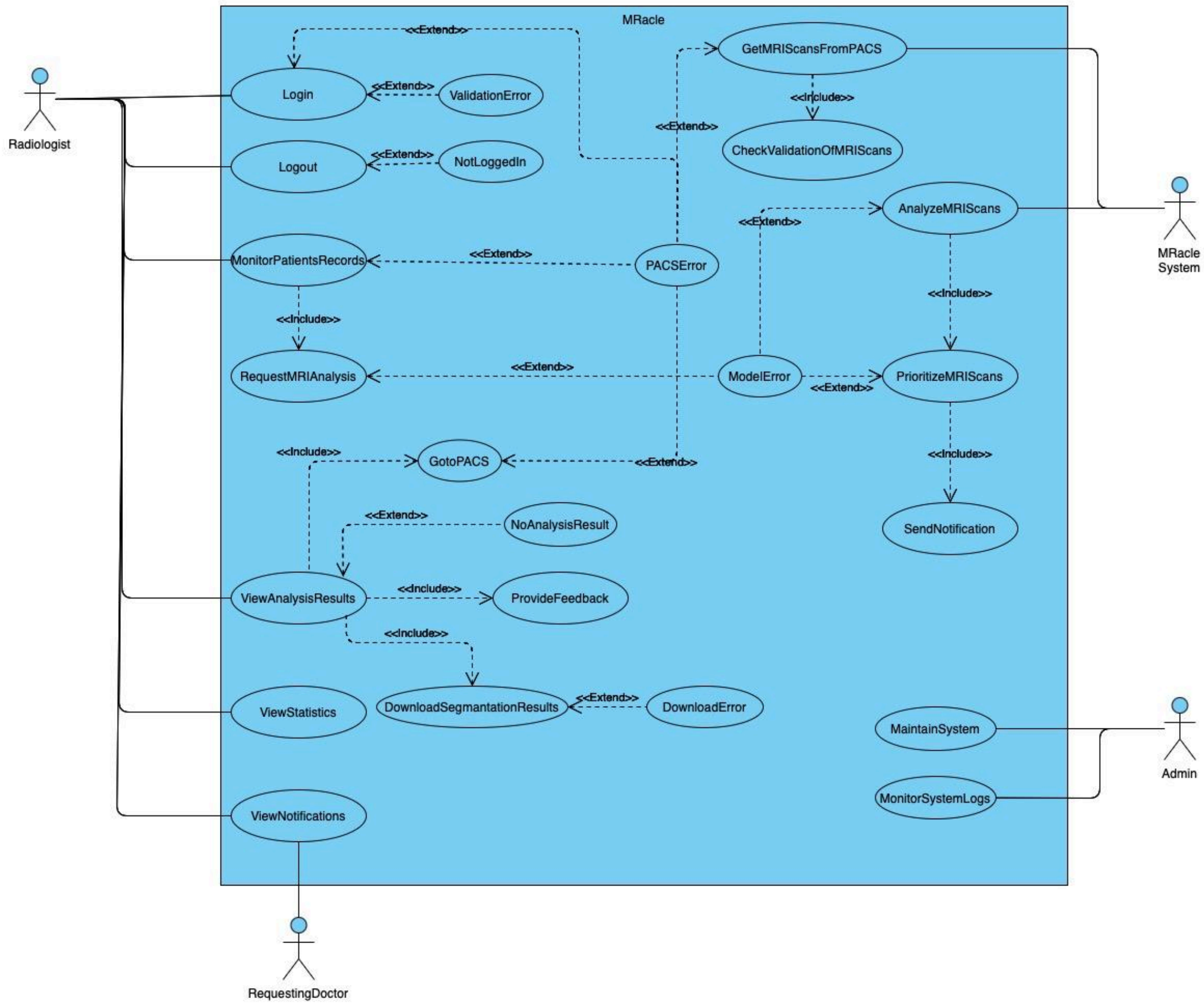

Figure 2: Use Case Diagram of the MRacle

3.6 Object and Class Model

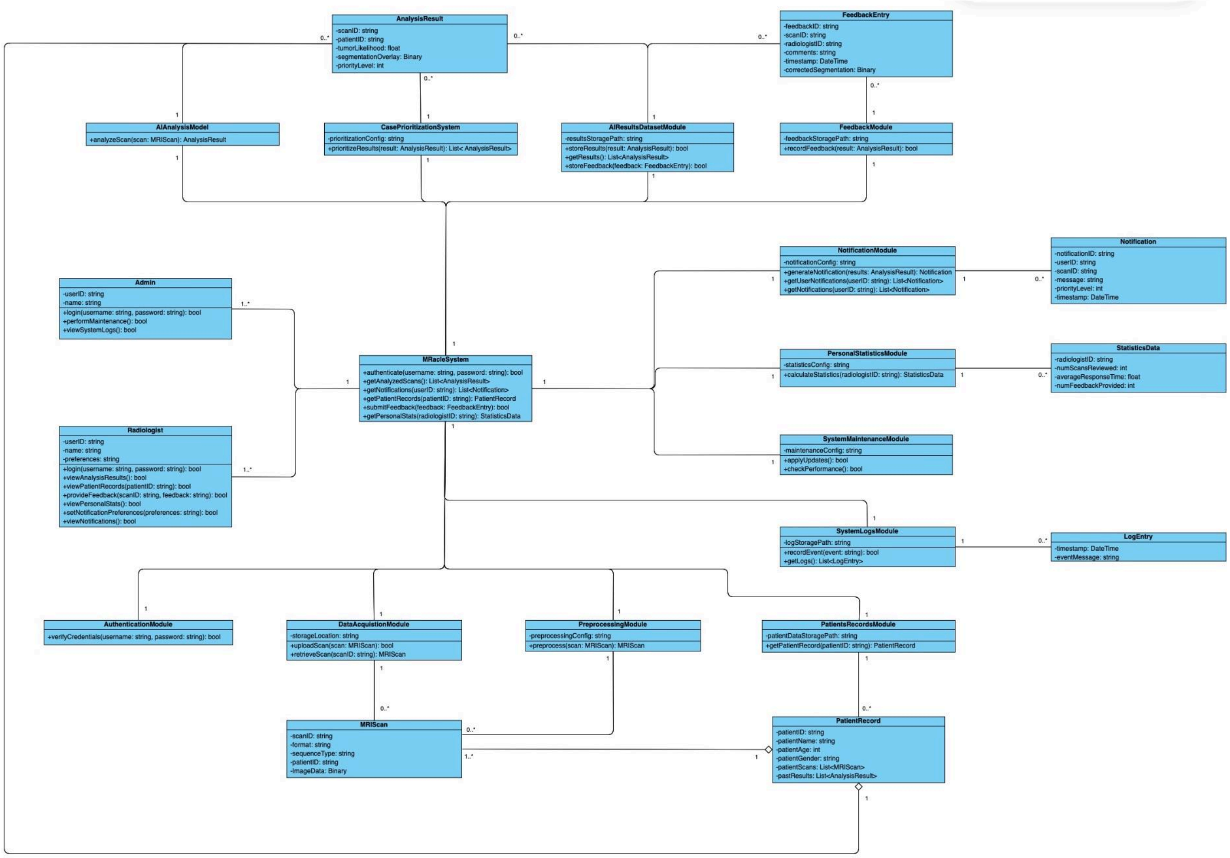

Figure 3: Class Diagram of the MRacle

3.7 Dynamic Models

3.7.1 Activity Diagrams

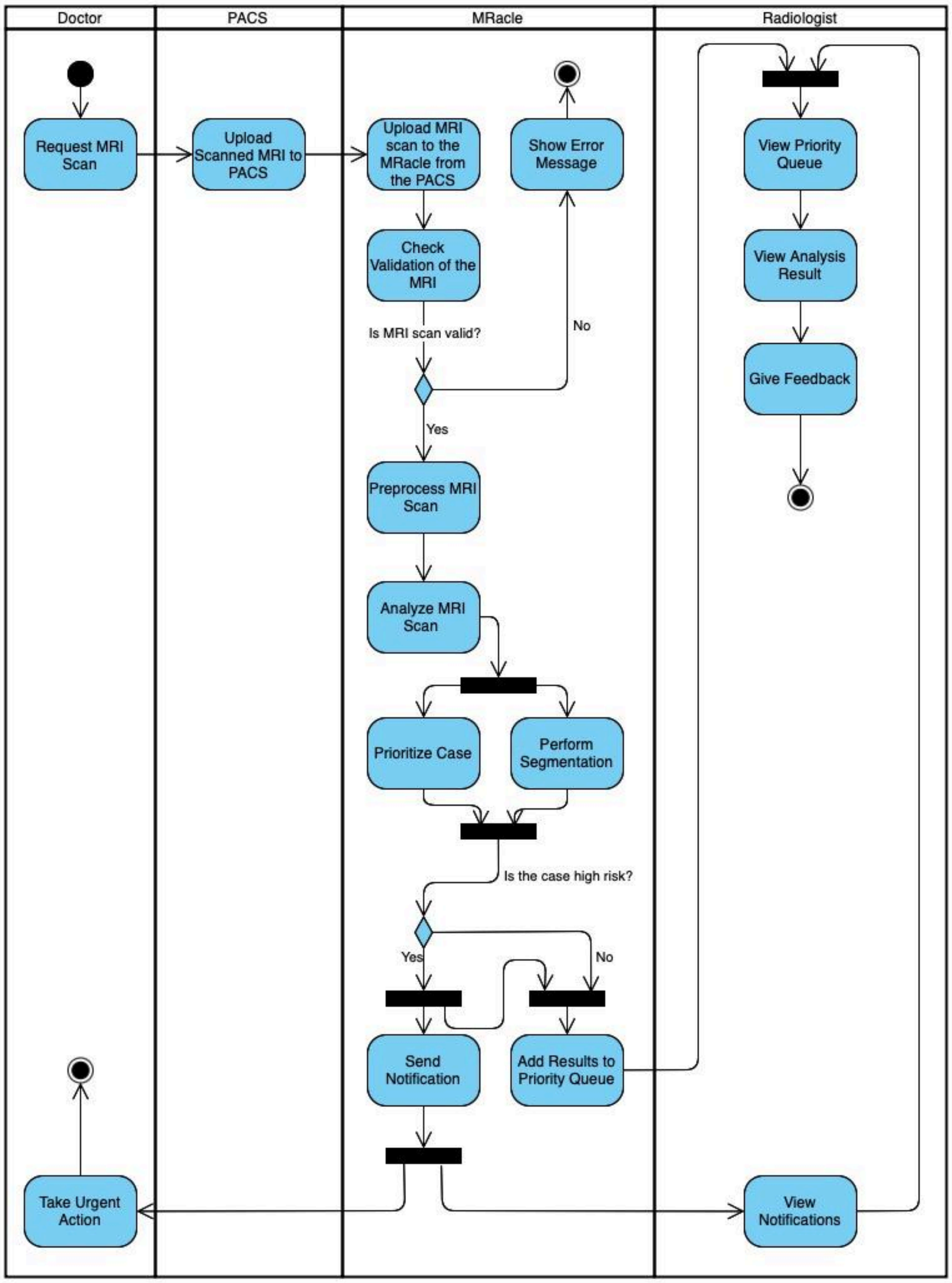

Figure 4: Activity Diagram for Automated MRacle Analysis

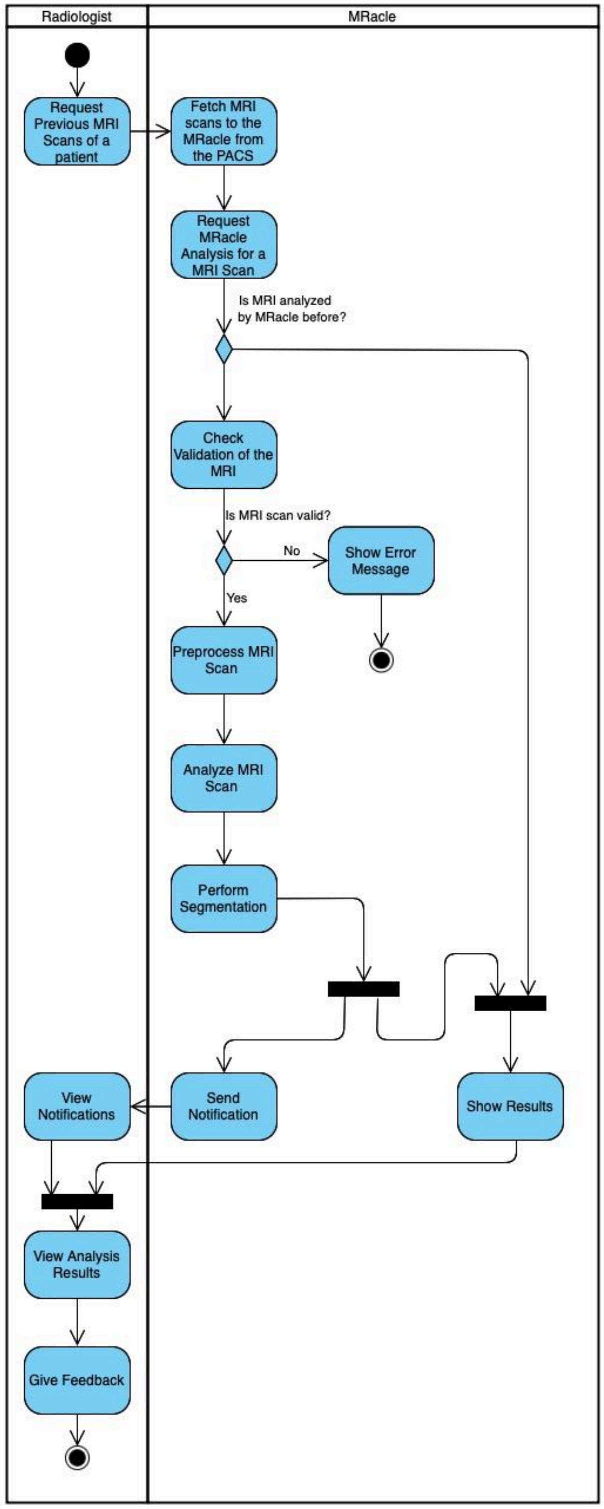

Figure 5: Activity Diagram for Subsequently Requested MRacle Analysis

3.7.2 State Diagrams

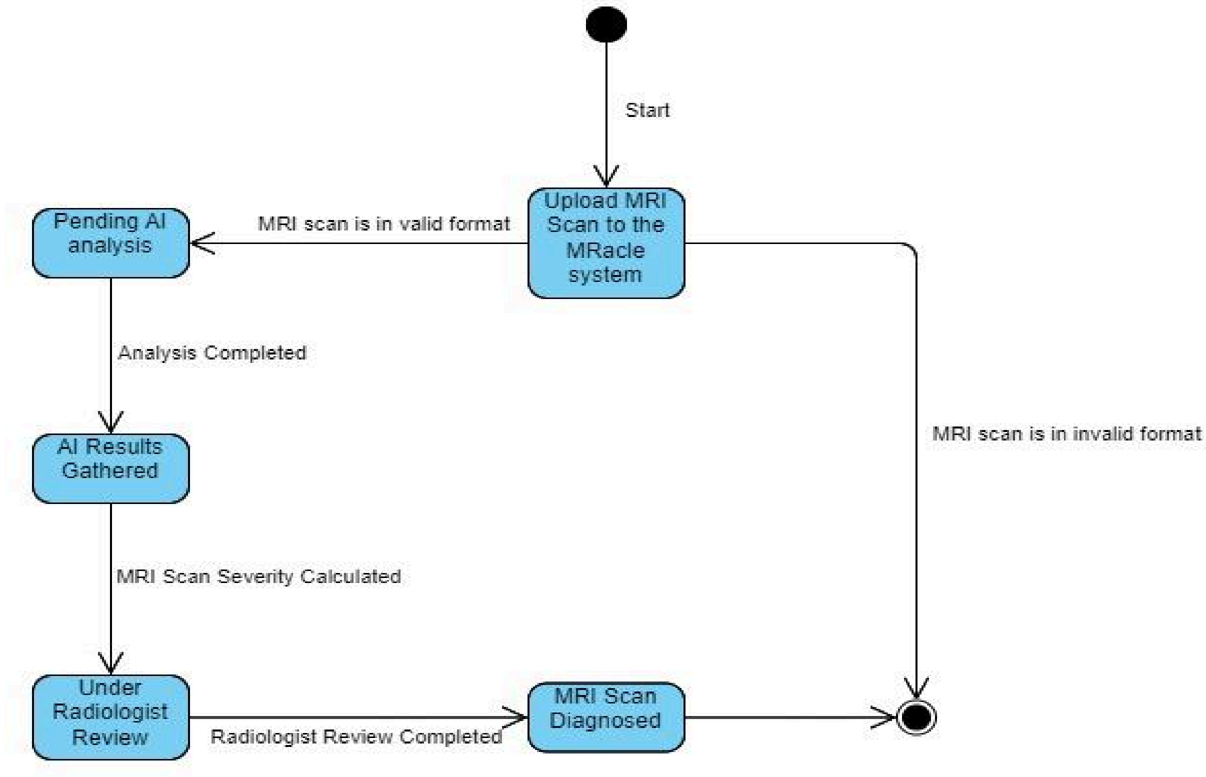

Figure 6: State Diagram of a MRI Scan

3.7.3 Sequence Diagrams

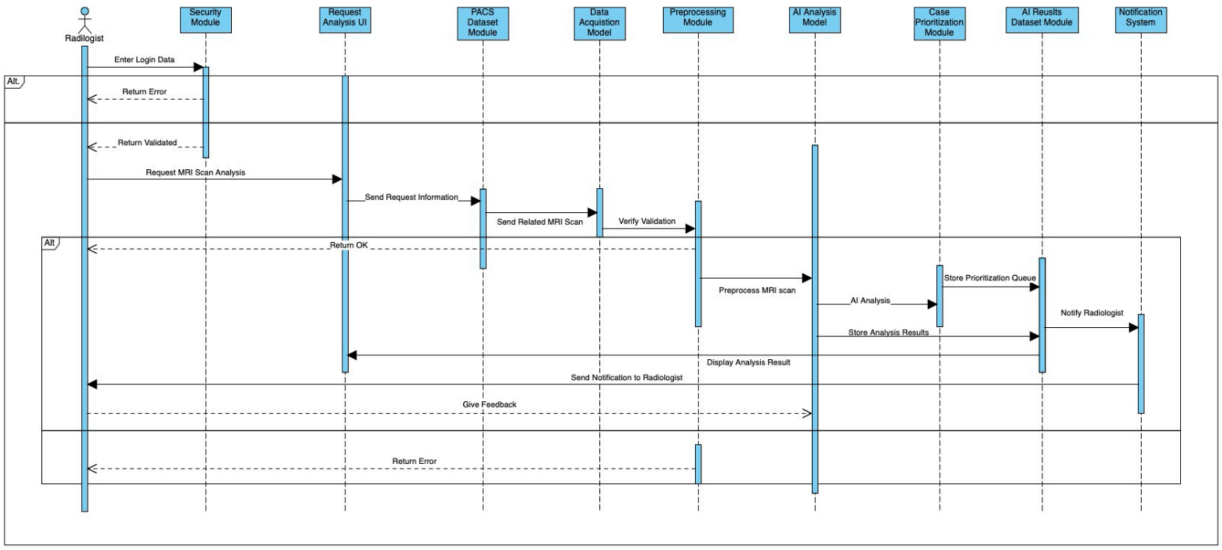

Figure 7: Sequence Diagram for AI Analysis Process

3.7.4 High-Level System Architecture

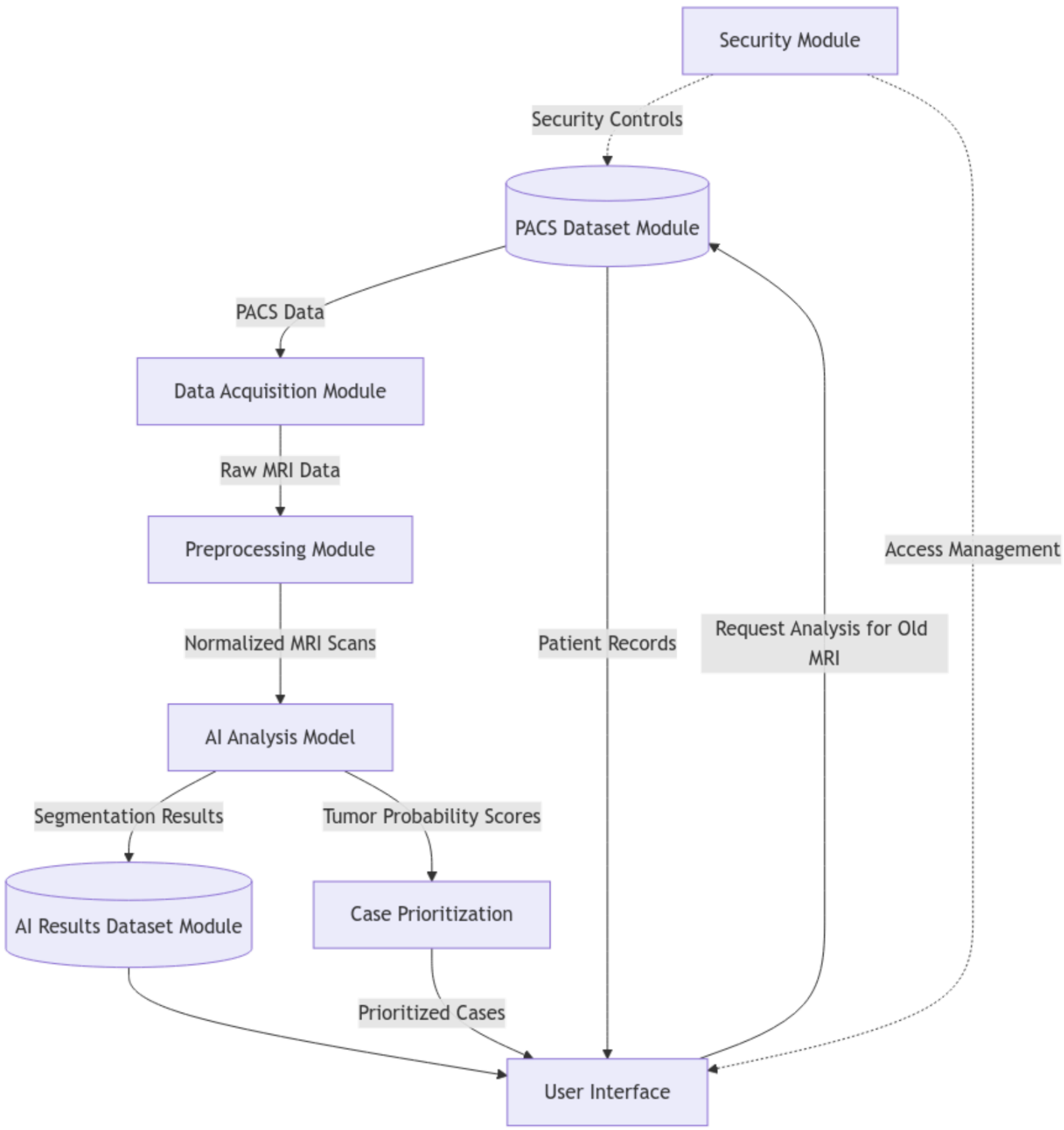

Figure 8: High-Level System Architecture Diagram

Components:

1. Security Module:

- Implements data encryption and access control mechanisms to ensure patient confidentiality.
- Monitors system activity to detect and prevent unauthorized access.

2. PACS Dataset Module:

- Interfaces with Picture Archiving and Communication Systems (PACS) to acquire and store MRI images.
- Integrates seamlessly with hospital radiology systems.

3. Data Acquisition Module:

- Manages acquisition of MRI scans in formats such as NIfTI [4] (.nii/.nii.gz) and DICOM [5] (.dcm).
- Supports incorporation of multiple MRI sequences (e.g., T1-CE, T2-FLAIR).
- Validates incoming data to ensure quality and format compatibility.

4. Preprocessing Module:

- Aligns MRI sequences, resamples images to a uniform voxel size, and performs skull-stripping.
- Ensures consistent data representation via normalization.

5. AI Analysis Model:

- Utilizes MRacle AI model for tumor detection and segmentation.

6. AI Results Dataset Module:

- Stores results generated by the AI analysis model, including tumor detection and segmentation outputs.

7. Case Prioritization System:

- Algorithms to prioritize cases based on tumor likelihood.
- Integration with radiologists' workflow for efficient case management.

8. User Interface (UI):

- An interface where the user can view all results from the model, such as segmentation results and tumor likelihood.
- A system where authorized users can upload MRI images externally to the system and request analysis.
- Radiologists can view patients' old MRIs and request MRacle analysis.

3.8 User Interface

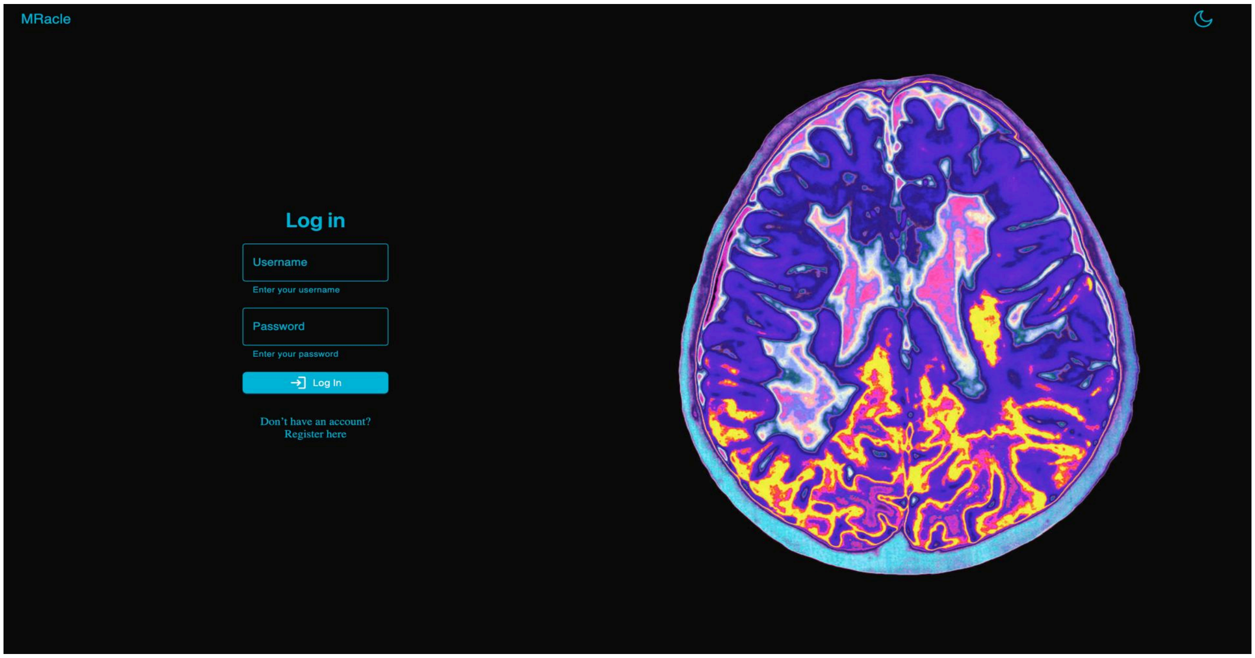

Figure 9: Login Screen for Radiologists

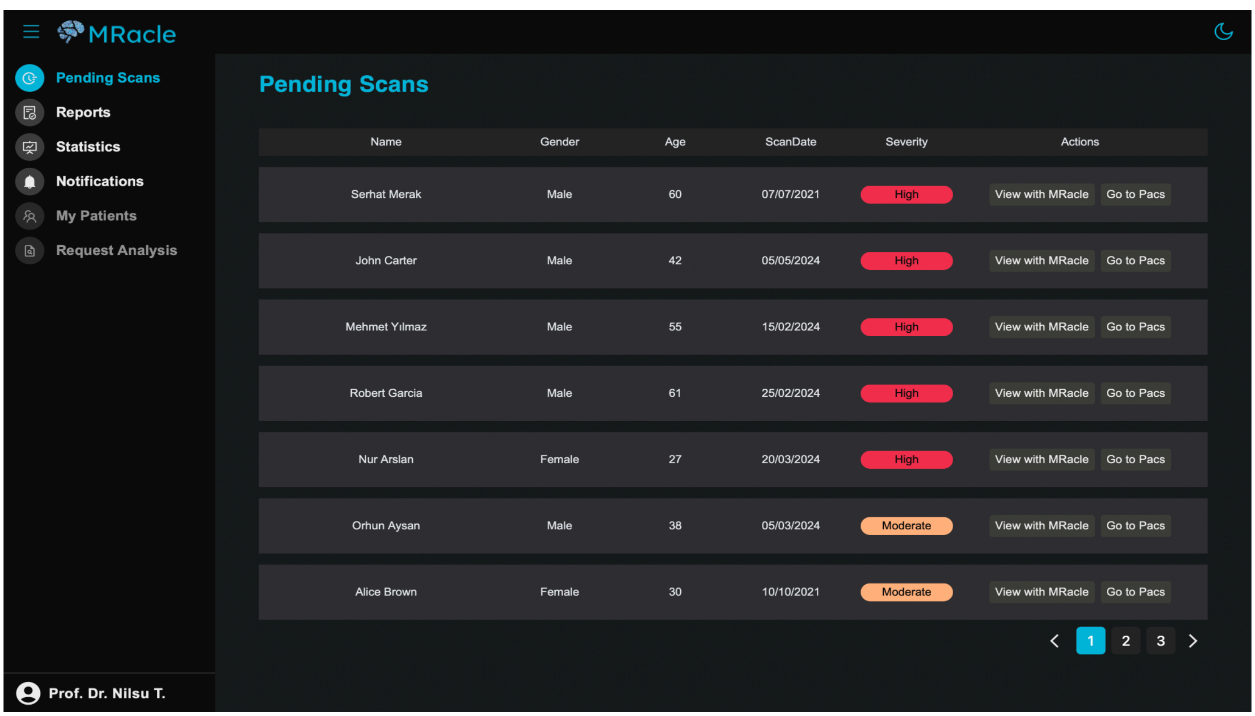

Figure 10: Screen of Analyzed and Prioritized Results Awaiting Radiologist Approval

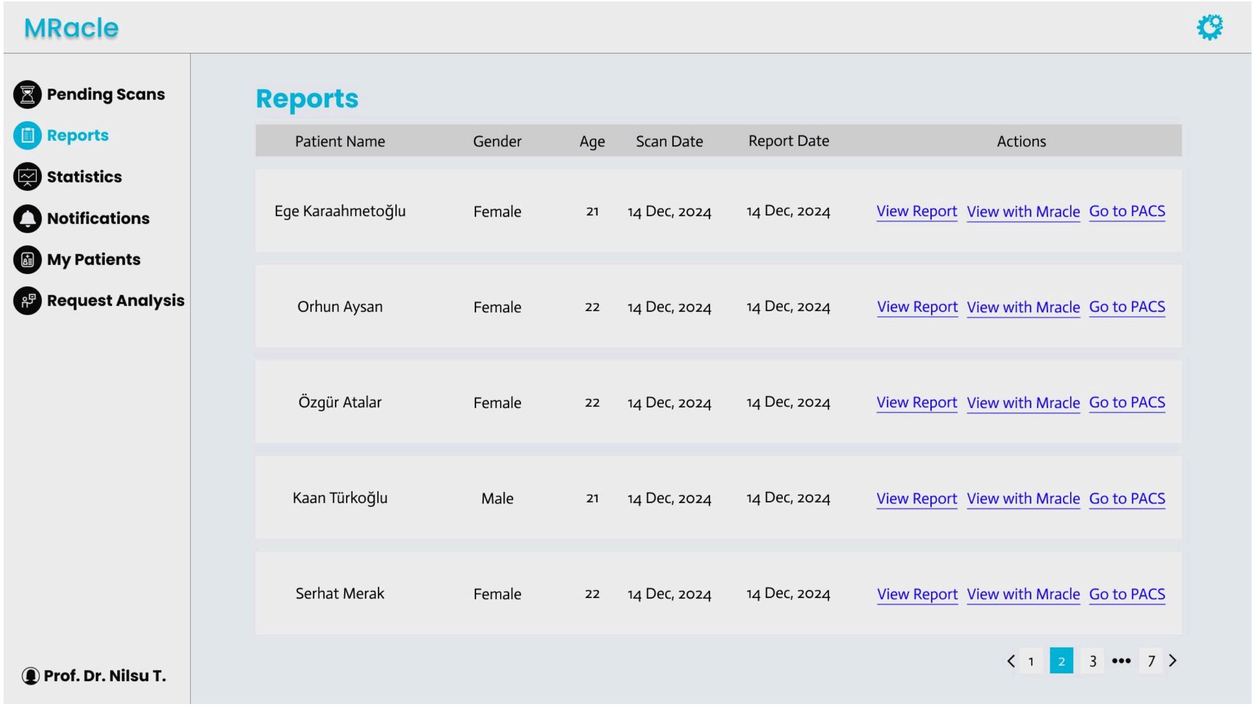

Figure 11: Screen of Radiologist's Latest Reports

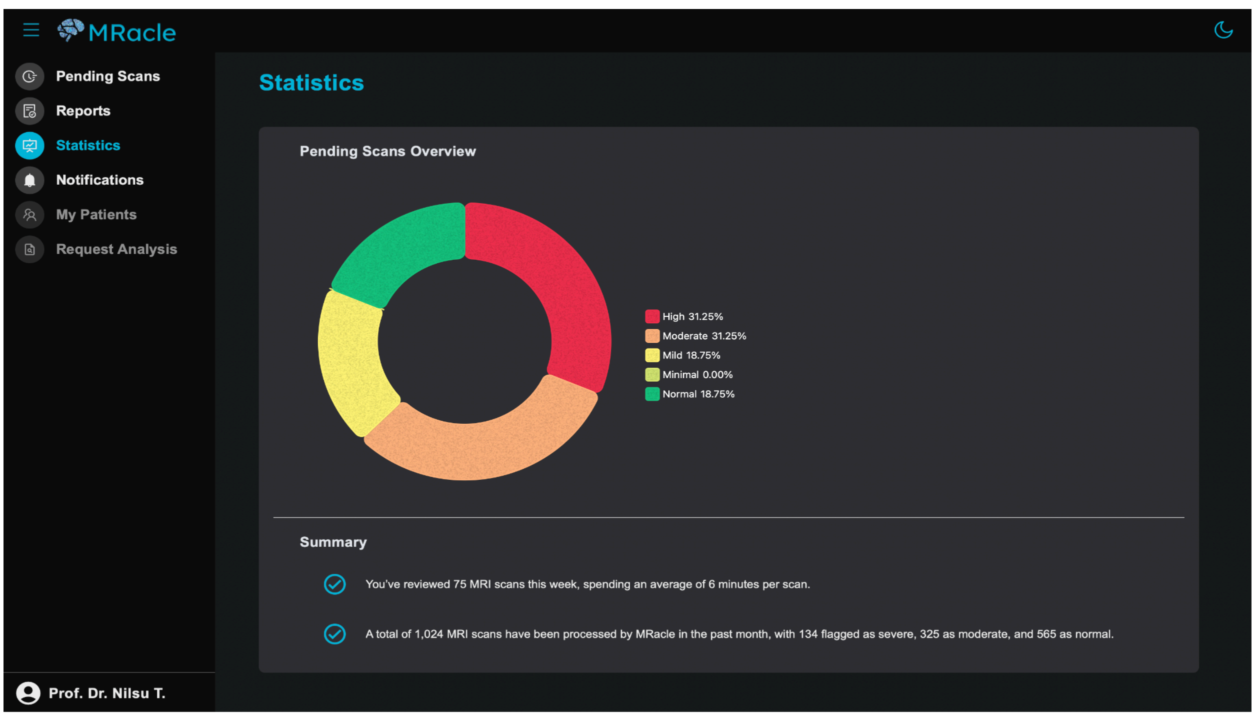

Figure 12: Screen of Radiologist's Statistics

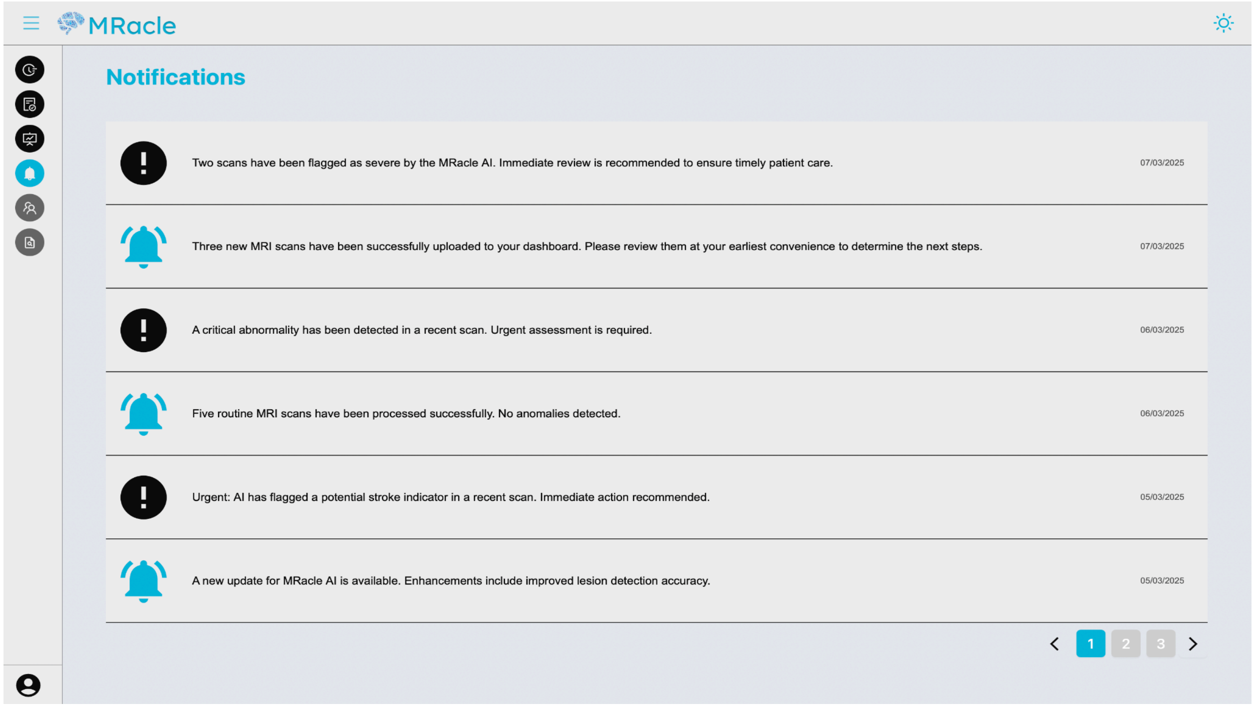

Figure 13: Screen of Notification

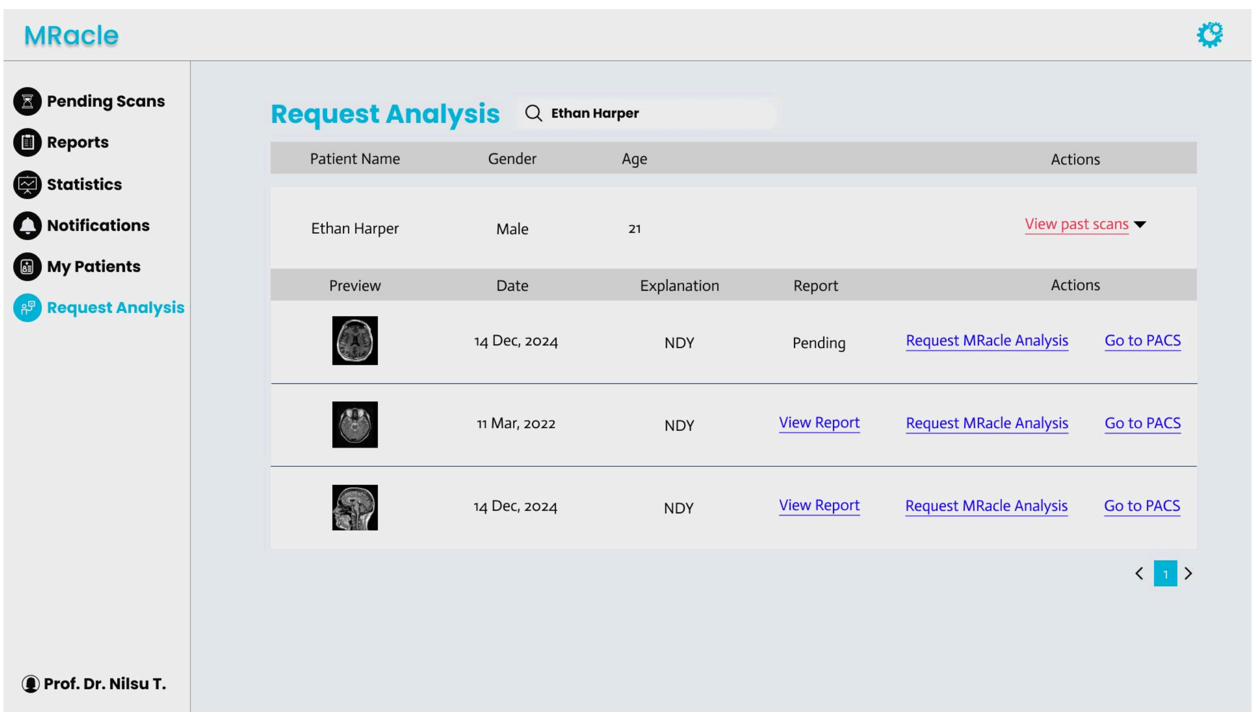

Figure 14: Screen of Radiologists Request MRacle Analysis for Patients' Old MRI Scans

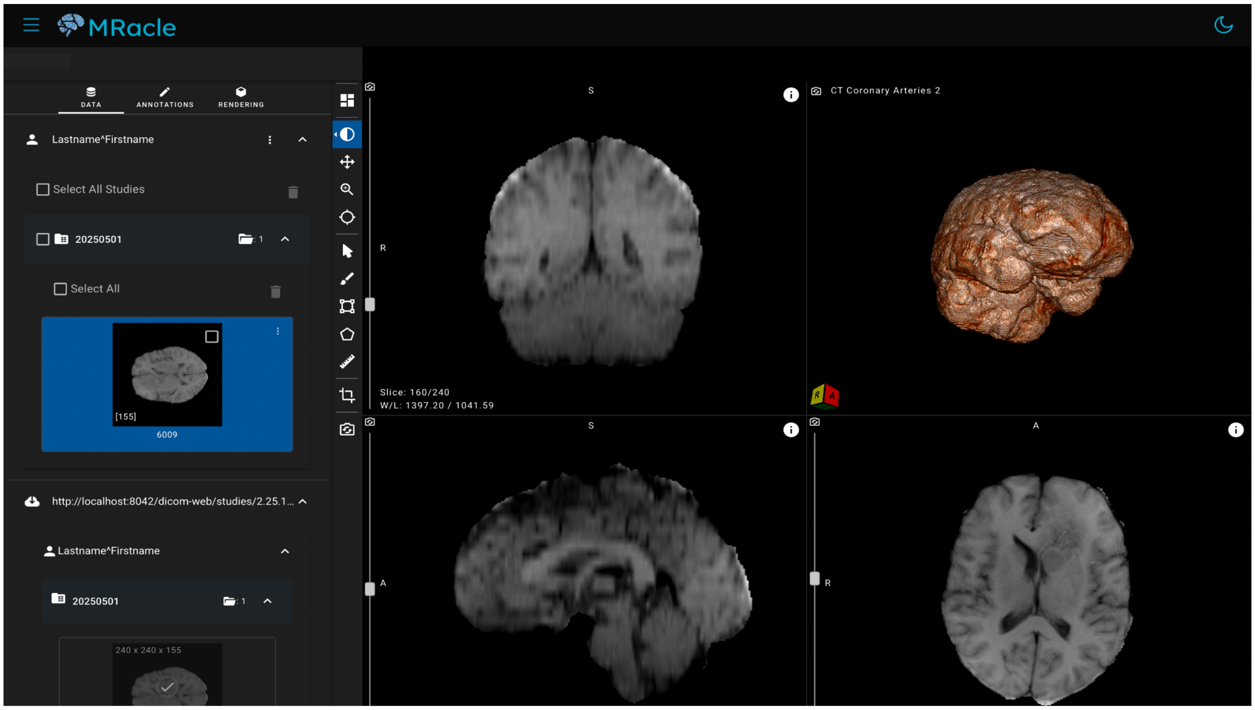

Figure 15: Screen of View Results and Segmentation

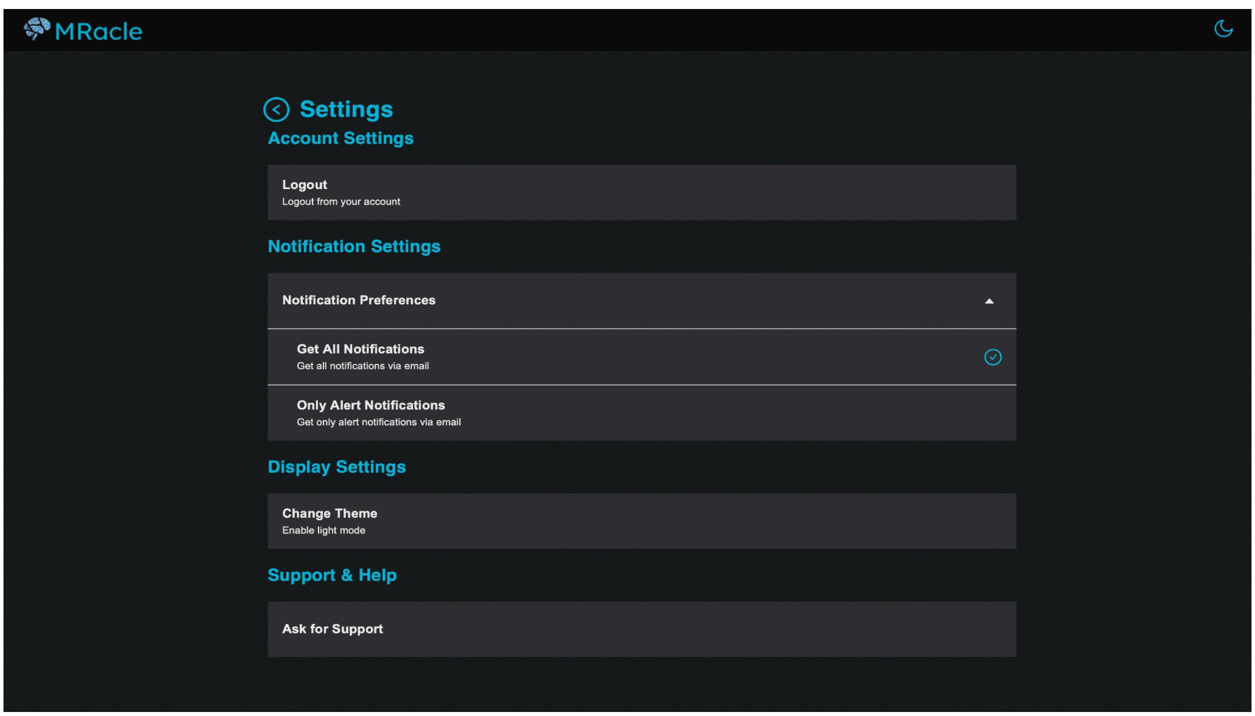

Figure 16: Screen of Options and Settings

4. Development and Implementation Details

4.1 Services

Our system relies on two core external services to emulate a hospital-like imaging environment and to present segmented MRI data to radiologists: the Orthanc PACS server and the OHIF Viewer [6].

Orthanc serves as our local Picture Archiving and Communication System (PACS), simulating the infrastructure of a real hospital without exposing patient data to the Internet. Installed on an isolated network, Orthanc receives and stores all DICOM-formatted MRI sequences generated by the MRI devices. Radiologists browse these sequences directly within the Orthanc web interface, inspecting image frames and embedded patient metadata. However, our architecture does not surface Orthanc's storage directly to end users. Instead, our backend maintains a persistent DICOM connection with Orthanc: whenever a new MRI series arrives, Orthanc triggers a notification that our service layer intercepts. Through Orthanc's API, we retrieve the recently added series from Orthanc for processing. Once acquired, the DICOM series is passed into our segmentation pipeline.

In DICOM, a study represents a complete imaging examination for a patient (e.g., a single MRI appointment), while a series is a subset of that study containing MR images. Segmentation results, still in DICOM format, are then pushed back to Orthanc under a study or series identifier. This approach preserves the clinical audit log within the PACS while keeping our processing logic entirely separate from the viewer.

To visualize both raw MRI frames and MRacle-generated segmentations outside of PACS, we integrate the OHIF Viewer as a standalone web service. OHIF is installed on a separate application server and configured to pull data from Orthanc via its built-in DICOMweb endpoints. When the radiologist logs into our application and selects a patient study, our frontend directs OHIF to fetch the corresponding DICOM series and overlays the segmentation results as an additional segmented series. Users can adjust opacity, toggle overlays on and off, and scroll through slices synchronously. This ensures that radiologists and other clinicians benefit from a full-featured, interactive 3D MRI viewer without needing direct access to the underlying PACS storage.

By positioning our application "in the middle" between the radiologist's workstation and the local PACS, we isolate all MRacle processing from the clinical network. The entire chain (from sequence arrival in Orthanc, through our segmentation engine, back to Orthanc, and finally rendered via OHIF) operates within the same secured environment, ensuring compliance with healthcare regulations while delivering seamless, real-time tumor illustration to end users.

4.2 Frontend

The MRacle frontend is delivered as a desktop application built with Electron and React, bootstrapped using Vite to ensure fast startup times and efficient hot-reloading during development. Electron was chosen over a purely web-based interface to meet hospital IT policies that often restrict browser installations and external network access. Within React, we used both Material-UI and Ant Design component libraries: Material-UI provides a consistent, accessible design for form controls, dialogs, and layout grids, while Ant Design's components enable quick implementation of data views. Together, these libraries allow us to build an eye-catching, intuitive interface that remains performant. The application supports both dark and light themes. Typography and color palettes were selected to optimize contrast and reduce eye strain during extended review sessions.

Upon launch, users are greeted by the Login Screen. Once logged in, radiologists are shown the Pending Scans screen, which lists MRI series that have completed processing through our segmentation pipeline but have not yet been reviewed or reported by a radiologist. Each entry displays patient identifiers, scan date, and a color-coded severity metric (High, Moderate, Mild, or Normal) derived from volumetric and morphological analysis of the segmented lesion. The list is sorted in descending order of severity to draw attention to the most critical cases. Click-through on the "View in MRacle" opens an embedded viewer powered by our OHIF integration, allowing navigation of raw and segmented images without leaving the desktop application.

The Reports screen shows all MRI series for which a diagnostic report has been written. Each row includes links to "View in MRacle" for an in-app review, as well as "View in PACS" to launch the native Orthanc web interface if more information about the patient is needed. A "View Report" action displays the radiologist's report. Both screens implement pagination, column sorting, and keyword filtering to help radiologists efficiently locate specific studies by patient name or severity category.

4.3 Backend

The MRacle backend is implemented in Python using FastAPI and managed via the uv package manager with a pyproject.toml to lock dependencies and simplify virtual environment creation. All the code runs inside this environment, ensuring that library versions (from FastAPI itself through the DICOM tooling and machine-learning frameworks) remain consistent across developer workstations.

MRacle currently relies on its token system because of the limitations of Orthanc's built-in access control for authentication. Although the open-source Orthanc PACS does not expose a robust, enterprise-grade authentication API, we designed MRacle so that in a real hospital deployment, it could adopt the PACS's user directory, allowing radiologists to sign in with their existing credentials and permissions.

At its core, the backend exposes a single REST endpoint, /predict, which initiates the segmentation pipeline for a specified study or series. In parallel, a WebSocket connection is maintained with Orthanc via DICOMweb event subscriptions: when Orthanc has a new MRI series, it sends a notification that our WebSocket listener picks up immediately. Upon receipt, MRacle's listener calls the same internal segmentation logic as the /predict endpoint, ensuring that every incoming study is queued for model processing without manual intervention.

Within the segmentation pipeline, the incoming DICOM series is first converted to a NIfTI volume using dicom2nifti, which normalizes image orientation and voxel spacing to match our model's training data. The preprocessed volume is then passed to our model, which outputs three probability maps corresponding to "edema," "non-enhancing tumor," and "enhancing tumor." From these maps, MRacle computes a severity metric (a weighted score reflecting the relative volumes and intensities of each tissue class) to prioritize critical cases. Finally, both the DICOM-formatted segmentation overlay and the numerical severity value are written to MRacle's local database; this storage layer maintains metadata linking studies, series UIDs, segmentation files, and severity scores so that the frontend can query and display them on demand. By combining FastAPI's high-performance routing, DICOMweb event handling, and a segmentation pipeline, the MRacle backend delivers automated, real-time tumor segmentation while remaining ready for integration with hospital IT infrastructures.

4.4 Model

Our tumor segmentation model is built upon the nnU-Net framework, which automates much of the configuration and optimization of a 3D U-Net for medical image segmentation. We trained exclusively on the BraTS 2023 adult glioma dataset, comprising 1,470 patients, each with four co-registered MRI modalities: T1, post-contrast T1-weighted (T1Gd), T2, and FLAIR. We write a conversion script to make this dataset compatible with the nnU-Net framework. The model achieved a mean validation Dice score of 0.91.

The Dice coefficient is defined as:

$$Dice(A, B) = \frac{2|A \cap B|}{|A| + |B|}$$

where A is the set of predicted tumor voxels and B the set of ground-truth voxels. This metric ranges from 0 (no overlap) to 1 (perfect overlap), directly reflecting the accuracy of spatial segmentation.

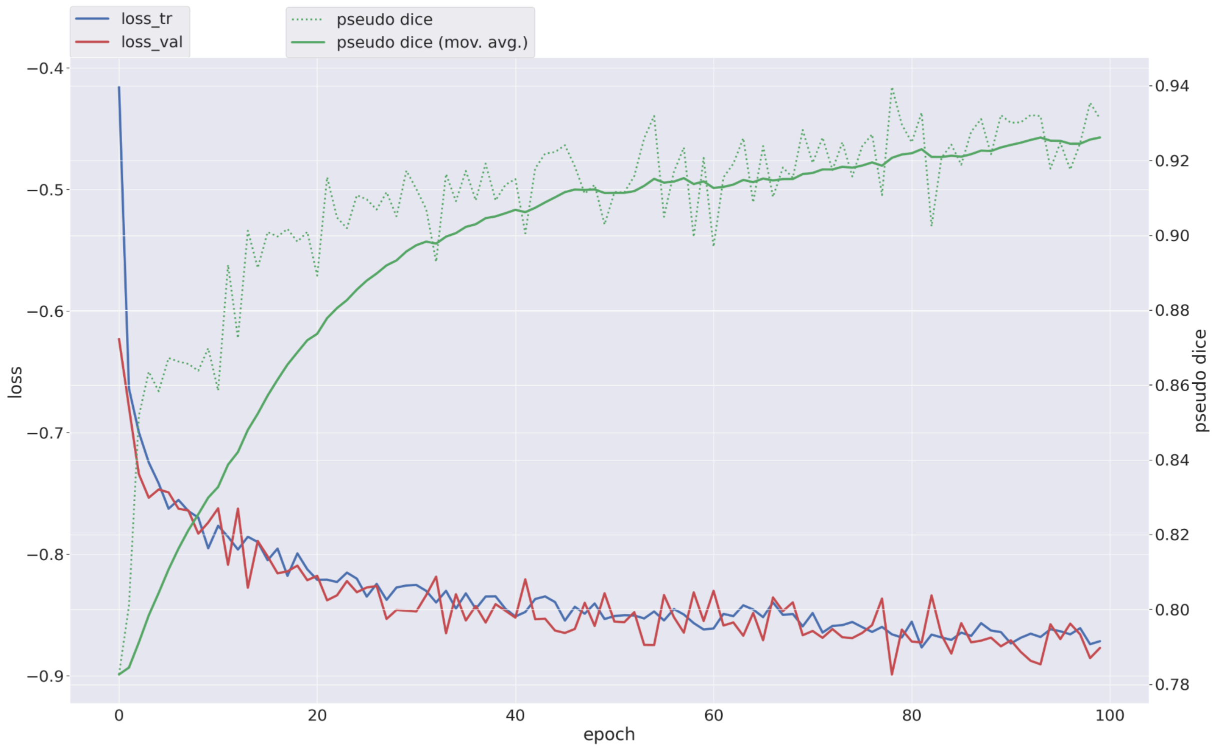

Figure 17: Model Training: Epoch vs Dice Loss

5. Test Cases and Results

5.1 Functional Tests

Test ID	FT-01	Priority	Critical
Test Type	Functional Testing (Integration Testing)		
Title	Test Cases for Radiologist Login		
Procedure	1. Check if the user can access the login page 2. Check if the user can enter their username and password 3. Check if the login button is clickable 4. Check if appropriate error messages are displayed for invalid credentials 5. Check if the user is redirected to the dashboard after successful login 6. Check if the session is maintained after login		
Expected Results	● Login page is accessible ● User can enter credentials in the respective fields ● The login button is clickable and functions properly ● Appropriate error messages are displayed for invalid credentials ● User is redirected to the dashboard after successful authentication ● User session is maintained throughout the system usage		
Result	Satisfactory - Tested on 01.05.2025		

Table 1: Functional Test Case 1

Test ID	FT-02	Priority	Critical
Test Type	Functional Testing (Integration Testing)		
Title	Test Cases for PACS Integration Authentication		
Procedure	1. Check if MRacle can connect to the hospital's PACS system 2. Check if MRacle can authenticate with the PACS system 3. Check if the authentication is maintained during the session 4. Check if error messages are displayed when PACS authentication fails		
Expected Results	● MRacle successfully connects to the hospital's PACS system ● Authentication with PACS is successful ● Appropriate error messages are displayed when authentication fails		
Result	Satisfactory - Tested on 01.05.2025		

Table 2: Functional Test Case 2

Test ID	FT-03	Priority	High
Test Type	Functional Testing (Integration Testing)		
Title	Test Cases for MRI Scan Upload		
Procedure	1. Check if the system can receive MRI scans from PACS 2. Check if the system can handle different MRI formats (NIfTI, DICOM) 3. Check if the system validates the uploaded scans correctly 4. Check if error messages are displayed for invalid scan formats 5. Check if the system stores the uploaded scans in the appropriate storage location		
Expected Results	● MRI scans are successfully received from PACS ● System accepts multiple MRI formats as specified in requirements ● Invalid scans are identified and appropriate error messages are displayed ● Successfully validated scans are stored in the correct location ● The upload process completes without errors or delays		
Result	Satisfactory - Tested on 01.05.2025		

Table 3: Functional Test Case 3

Test ID	FT-04	Priority	High
Test Type	Functional Testing (Integration Testing)		
Title	Test Cases for Multiple MRI Sequence Synchronization		
Procedure	1. Check if multiple MRI sequences (T1-CE, T2-FLAIR) can be uploaded for a single patient 2. Check if the sequences are properly linked to the same patient 3. Check if the system allows synchronized navigation between different sequences 4. Check if the synchronization maintains correct anatomical alignment		
Expected Results	● Multiple MRI sequences are successfully linked to the same patient ● Navigation between different sequences is synchronized ● Anatomical alignment is maintained during synchronized viewing ● Changes in zoom, pan, or scroll are reflected across all sequences		
Result	Satisfactory - Tested on 01.05.2025		

Table 4: Functional Test Case 4

<table>
<tr><td>Test ID</td><td>FT-05</td><td>Priority</td><td>Critical</td></tr>
<tr><td>Test Type</td><td colspan="3">Functional Testing (Integration Testing)</td></tr>
<tr><td>Title</td><td colspan="3">Test Cases for AI Analysis Initiation</td></tr>
<tr><td>Procedure</td><td colspan="3">1. Check if the radiologist can initiate AI analysis on selected MRI scans
2. Check if the preprocessing module correctly standardizes the scans
3. Check if the AI model receives the preprocessed data correctly
4. Check if the system displays a progress indicator during analysis
5. Check if the system notifies when analysis is complete</td></tr>
<tr><td>Expected Results</td><td colspan="3">● Radiologist can successfully initiate AI analysis
● Preprocessing module standardizes scans (alignment, resampling, skull-stripping)
● AI model receives preprocessed data without errors
● Progress indicator is displayed during analysis
● System notifies when analysis is complete</td></tr>
<tr><td>Result</td><td colspan="3">Satisfactory - Tested on 01.05.2025</td></tr>
</table>

Table 5: Functional Test Case 5

<table>
<tr><td>Test ID</td><td>FT-06</td><td>Priority</td><td>High</td></tr>
<tr><td>Test Type</td><td colspan="3">Functional Testing (Integration Testing)</td></tr>
<tr><td>Title</td><td colspan="3">Test Cases for AI Analysis Results Display</td></tr>
<tr><td>Procedure</td><td colspan="3">1. Check if segmentation overlays are correctly displayed on MRI images
2. Check if tumor likelihood scores are calculated and displayed
3. Check if different segmentation types are visually distinguishable
4. Check if the user can toggle the visibility of segmentation overlays
5. Check if the user can adjust the opacity of overlays</td></tr>
<tr><td>Expected Results</td><td colspan="3">● Segmentation overlays are accurately positioned on the MRI images
● Tumor likelihood scores are calculated and displayed correctly
● Different types of segmentation are visually distinguishable (e.g., by color)
● User can toggle the visibility of segmentation overlays
● User can adjust the opacity of overlays to better view underlying anatomy</td></tr>
<tr><td>Result</td><td colspan="3">Satisfactory - Tested on 01.05.2025</td></tr>
</table>

Table 6: Functional Test Case 6

Test ID	FT-07	Priority	High
Test Type	Functional Testing (Integration Testing)		
Title	Test Cases for Case Prioritization System		
Procedure	1. Check if cases are prioritized based on tumor likelihood scores 2. Check if high-risk cases are placed at the top of the worklist 3. Check if the prioritization updates in real-time as new scans are analyzed 4. Check if the radiologist can manually adjust the priority of cases 5. Check if the priority status is visually indicated in the worklist		
Expected Results	● Cases are automatically prioritized based on tumor likelihood ● High-risk cases appear at the top of the worklist ● Prioritization updates in real-time with new analysis results ● Radiologist can manually adjust priority if necessary ● Priority status is clearly indicated visually in the worklist		
Result	Satisfactory - Tested on 01.05.2025		

Table 7: Functional Test Case 7

Test ID	FT-08	Priority	High
Test Type	Functional Testing (Integration Testing)		
Title	Test Cases for Notification System		
Procedure	1. Check if notifications are sent to radiologists for high-priority cases 2. Check if notifications are sent to the requesting doctor 3. Check if notifications are delivered through configured channels (in-app, email) 4. Check if notification preferences can be configured 5. Check if notifications can be marked as read		
Expected Results	● Notifications are sent for high-priority cases ● Requesting doctors receive appropriate notifications ● Notifications are delivered through configured channels ● Users can configure notification preferences ● Notifications can be marked as read and are tracked properly		
Result	Planned to be added in the upcoming version.		

Table 8: Functional Test Case 8

Test ID	FT-09	Priority	High
Test Type	Functional Testing (Integration Testing)		
Title	Test Cases for Patient Record Viewing		
Procedure	1. Check if radiologists can search for patients by identifier 2. Check if radiologists can view a patient's past MRI scans and reports 3. Check if the system maintains data privacy by restricting access to authorized users 4. Check if the patient records are displayed in a chronological order 5. Check if the user can filter patient records by date or scan type		
Expected Results	● Radiologists can search for patients using identifiers ● Past MRI scans and reports are accessible ● Only authorized users can access patient data ● Records are displayed in chronological order ● Users can filter records by date or scan type		
Result	Satisfactory - Tested on 01.05.2025		

Table 9: Functional Test Case 9

Test ID	FT-10	Priority	High
Test Type	Functional Testing (Integration Testing)		
Title	Test Cases for Requesting Analysis on Previous Scans		
Procedure	1. Check if radiologists can select previous scans from patient records 2. Check if radiologists can request AI analysis on selected previous scans 3. Check if the system processes the request and performs the analysis 4. Check if the results are stored with appropriate timestamps 5. Check if the results are accessible in the patient's record		
Expected Results	● Radiologists can select previous scans from patient records ● AI analysis can be requested on previous scans ● System processes the request and performs analysis ● Results are stored with appropriate timestamps ● Results are accessible in the patient's record		
Result	Satisfactory - Tested on 01.05.2025		

Table 10: Functional Test Case 10

Test ID	FT-11	Priority	High
Test Type	Functional Testing (Integration Testing)		
Title	Test Cases for AI Analysis Feedback		
Procedure	1. Check if radiologists can annotate AI-generated results 2. Check if radiologists can mark false positives/negatives 3. Check if radiologists can provide additional comments 4. Check if the feedback is properly stored in the AI Results Dataset 5. Check if confirmation is displayed after feedback submission		
Expected Results	● Radiologists can annotate AI-generated results ● False positives/negatives can be marked ● Additional comments can be provided ● Feedback is stored in the AI Results Dataset ● Confirmation is displayed after feedback submission		
Result	Satisfactory - Tested on 01.05.2025		

Table 11: Functional Test Case 11

Test ID	FT-12	Priority	Minor
Test Type	Functional Testing (Integration Testing)		
Title	Test Cases for Radiologist Statistics Page		
Procedure	1. Check if personal statistics are calculated correctly 2. Check if statistics include number of scans reviewed 3. Check if statistics include response times 4. Check if statistics include feedback provided 5. Check if statistics are displayed in an intuitive format		
Expected Results	● Personal statistics are calculated correctly ● Number of scans reviewed is displayed ● Response times are displayed ● Feedback provided is displayed ● Statistics are displayed in an intuitive format (charts, graphs)		
Result	Satisfactory - Tested on 01.05.2025		

Table 12: Functional Test Case 12

Test ID	FT-13	Priority	High
Test Type	Functional Testing (Integration Testing)		
Title	Test Cases for System Maintenance		
Procedure	1. Check if administrators can view system logs 2. Check if administrators can apply software updates 3. Check if administrators can perform database maintenance 4. Check if maintenance activities are logged for audit purposes		
Expected Results	● System logs are viewable ● Software updates can be applied ● Database maintenance can be performed ● Maintenance activities are logged for audit purposes		
Result	Satisfactory - Tested on 01.05.2025		

Table 13: Functional Test Case 13

Test ID	FT-14	Priority	High
Test Type	Functional Testing (Integration Testing)		
Title	Test Cases for System Logs Monitoring		
Procedure	1. Check if administrators can filter logs by type (user activities, system errors) 2. Check if administrators can filter logs by date range 3. Check if logs contain essential information (user, action, timestamp) 4. Check if critical errors are highlighted or flagged		
Expected Results	● Logs can be filtered by type ● Logs can be filtered by date range ● Logs contain essential information ● Critical errors are highlighted or flagged		
Result	Satisfactory - Tested on 01.05.2025		

Table 14: Functional Test Case 14

Test ID	FT-15	Priority	High
Test Type	Functional Testing (Integration Testing)		
Title	Test Cases for Dashboard Layout		
Procedure	1. Check if the dashboard displays all required elements 2. Check if the layout is intuitive and follows design guidelines 3. Check if the most important information is clearly visible 4. Check if the dashboard adapts to different screen sizes 5. Check if navigation to other sections is easily accessible		
Expected Results	• Dashboard displays all required elements • Layout is intuitive and follows design guidelines • Important information is clearly visible • Dashboard adapts to different screen sizes • Navigation to other sections is easily accessible		
Result	Satisfactory - Tested on 01.05.2025		

Table 15: Functional Test Case 15

Test ID	FT-16	Priority	High
Test Type	Functional Testing (Integration Testing)		
Title	Test Cases for MRI Viewer Controls		
Procedure	1. Check if zoom controls work correctly 2. Check if pan controls work correctly 3. Check if brightness/contrast adjustment works correctly 4. Check if measurement tools work correctly 5. Check if keyboard shortcuts function as expected		
Expected Results	• Zoom controls work correctly • Pan controls work correctly • Brightness/contrast adjustment works correctly • Measurement tools work correctly • Keyboard shortcuts function as expected		
Result	Satisfactory - Tested on 01.05.2025		

Table 16: Functional Test Case 16

Test ID	FT-17	Priority	Critical
Test Type	Functional Testing (Integration Testing)		
Title	Test Cases for NIfTI to DICOM Conversion: Brain MRI		
Procedure	1. Check if the conversion module accepts a standard brain MRI in NIfTI format. 2. Check if the module invokes the conversion routine without errors. 3. Check if the output DICOM series contains the correct NIfTI data. 4. Check if the image orientation and voxel spacing in the DICOM headers match the source NIfTI. 5. Check if a DICOM viewer displays anatomically correct brain slices.		
Expected Results	● Conversion completes successfully with no exceptions. ● All DICOM header fields reflect the source NIfTI header. ● Image orientation matrices and voxel dimensions match the original NIfTI. ● The DICOM viewer displays anatomically correct brain slices.		
Result	Satisfactory - Tested on 01.05.2025		

Table 17: Functional Test Case 17

Test ID	FT-18	Priority	Critical
Test Type	Functional Testing (Integration Testing)		
Title	Test Cases for Handling Different MRI Sequences		
Procedure	1. Check if the system correctly identifies T1-weighted sequences 2. Check if the system correctly identifies T1-weighted contrast-enhanced sequences 3. Check if the system correctly identifies T2-weighted sequences 4. Check if the system correctly identifies T2-FLAIR sequences 5. Check if the system correctly processes each sequence type		
Expected Results	● System correctly identifies T1-weighted sequences ● System correctly identifies T1-weighted contrast-enhanced sequences ● System correctly identifies T2-weighted sequences ● System correctly identifies T2-FLAIR sequences ● System correctly processes each sequence type		
Result	Satisfactory - Tested on 01.05.2025		

Table 18: Functional Test Case 18

Test ID	FT-19	Priority	Critical
Test Type	Functional Testing (Integration Testing)		
Title	Test Cases for NIfTI to DICOM Conversion: Segmentation Mask		
Procedure	1. Check if the system accepts a binary segmentation mask in NIfTI. 2. Check if conversion to a DICOM Segmentation Object completes without errors. 3. Check if the segmentation object's header references the correct source series (StudyInstanceUID/SeriesInstanceUID). 4. Check if the pixel data of the segmentation align spatially. 5. Check if overlaying the segmentation in a DICOM viewer aligns.		
Expected Results	● A valid DICOM segmentation object is produced. ● References in the segmentation header correctly point to the anatomical series. ● Segmentation overlays align precisely with anatomy in the viewer.		
Result	Planned to be added in the upcoming version.		

Table 19: Functional Test Case 19

Test ID	FT-20	Priority	Critical
Test Type	Functional Testing (Integration Testing)		
Title	Test Cases for Tumor Likelihood Scoring		
Procedure	1. Check if the backend calculates tumor likelihood scores 2. Check if the scores correlate with actual tumor presence 3. Check if the scores are normalized and consistent 4. Check if the scores are displayed correctly 5. Check if the scores are used correctly for prioritization		
Expected Results	● Backend calculates tumor likelihood scores ● Scores correlate with actual tumor presence ● Scores are normalized and consistent ● Scores are displayed correctly ● Scores are used correctly for prioritization		
Result	Satisfactory - Tested on 01.05.2025		

Table 20: Functional Test Case 20

Test ID	FT-21	**Priority**	High
Test Type	Functional Testing (Integration Testing)		
Title	Test Cases for DICOM to NIfTI Conversion: Brain MRI		
Procedure	1. Check if the conversion tool accepts a series of brain MRI DICOM. 2. Check if invoking the DICOM-to-NIfTI routine completes without errors. 3. Check if the output NIfTI volume includes the complete 3D dataset. 4. Check if the affine transformation encodes spacing, orientation, and origin correctly. 5. Check if an NIfTI viewer (e.g., ITK-SNAP) renders a continuous, correctly ordered 3D brain volume.		
Expected Results	● The NIfTI file is generated without errors. ● Spatial metadata matches the DICOM tags (ImageOrientationPatient, ImagePositionPatient, PixelSpacing). ● Visualization shows a continuous, correctly ordered 3D brain volume.		
Result	Satisfactory - Tested on 01.05.2025		

Table 21: Functional Test Case 21

Test ID	FT-22	**Priority**	High
Test Type	Functional Testing (Integration Testing)		
Title	Test Cases for DICOM to NIfTI Conversion: Segmentation Mask		
Procedure	1. Check if the system accepts a DICOM Segmentation Object. 2. Check if conversion to one or more NIfTI mask files completes successfully. 3. Check if each mask NIfTI corresponds to its segmented structure.. 4. Check if overlaying masks on the anatomical NIfTI in a viewer aligns correctly.		
Expected Results	● One or more NIfTI mask files are generated for each segmentation label. ● Affine transforms ensure voxel-perfect alignment. ● Viewer correctly renders mask overlays on anatomy.		
Result	Satisfactory - Tested on 01.05.2025		

Table 22: Functional Test Case 22

Test ID	FT-23	Priority	High
Test Type	Functional Testing (Integration Testing)		
Title	Test Cases for Data Storage		
Procedure	1. Check if MRI scans are stored correctly 2. Check if AI analysis results are stored correctly 3. Check if patient information is stored correctly 4. Check if reports are stored correctly 5. Check if storage capacity is monitored		
Expected Results	● MRI scans are stored correctly ● AI analysis results are stored correctly ● Patient information is stored correctly ● Reports are stored correctly ● Storage capacity is monitored		
Result	Satisfactory - Tested on 01.05.2025		

Table 23: Functional Test Case 23

Test ID	FT-24	Priority	High
Test Type	Functional Testing (Integration Testing)		
Title	Test Cases for Data Retrieval		
Procedure	1. Check if MRI scans can be retrieved quickly 2. Check if AI analysis results can be retrieved quickly 3. Check if patient information can be retrieved quickly 4. Check if reports can be retrieved quickly 5. Check if data retrieval remains performant with large datasets		
Expected Results	● MRI scans are stored correctly ● AI analysis results are stored correctly ● Patient information is stored correctly ● Reports are stored correctly ● Storage capacity is monitored		
Result	Satisfactory - Tested on 01.05.2025		

Table 24:Functional Test Case 24

Test ID	FT-25	Priority	High
Test Type	Functional Testing (Integration Testing)		
Title	Test Cases for Handling Corrupted Conversion Inputs		
Procedure	1. Check if the system detects a corrupted or truncated NIfTI file during conversion. 2. Check if the system detects a DICOM series missing critical tags. 3. Check if meaningful error messages are returned. 4. Check if no partial or malformed output files are written in either scenario.		
Expected Results	● Conversion routines detect corruption or missing data and abort. ● Clear, actionable error messages are provided without crashing. ● No invalid output files are produced.		
Result	Satisfactory - Tested on 01.05.2025		

Table 25: Functional Test Case 25

Test ID	FT-26	Priority	High
Test Type	Functional Testing (Integration Testing)		
Title	Test Cases for Historical Scan Navigation		
Procedure	1. Check if radiologists can navigate between historical scans 2. Check if scan dates are clearly displayed 3. Check if analysis results for historical scans are accessible 4. Check if scans are displayed in chronological order		
Expected Results	● Radiologists can navigate between historical scans ● Scan dates are clearly displayed ● Analysis results for historical scans are accessible ● Scans are displayed in chronological order		
Result	Satisfactory - Tested on 01.05.2025		

Table 26: Functional Test Case 26

Test ID	FT-27	Priority	High
Test Type	Functional Testing (Integration Testing)		
Title	Test Cases for Error Handling During Scan Upload		
Procedure	1. Check if the system handles corrupted files appropriately 2. Check if the system handles unsupported file formats appropriately 3. Check if the system handles incomplete uploads appropriately 4. Check if error messages are clear and helpful 5. Check if the system allows retry after error		
Expected Results	● System handles corrupted files appropriately ● System handles unsupported file formats appropriately ● System handles incomplete uploads appropriately ● Error messages are clear and helpful ● System allows retry after error		
Result	Satisfactory - Tested on 01.05.2025		

Table 27: Functional Test Case 27

Test ID	FT-28	Priority	High
Test Type	Functional Testing (Integration Testing)		
Title	Test Cases for Error Handling During AI Analysis		
Procedure	1. Check if the system handles preprocessing failures appropriately 2. Check if the system handles model execution failures appropriately 3. Check if the system handles resource limitations appropriately 4. Check if error messages are logged 5. Check if users are notified of analysis failures		
Expected Results	● System handles preprocessing failures appropriately ● System handles model execution failures appropriately ● System handles resource limitations appropriately ● Error messages are logged ● Users are notified of analysis failures		
Result	Planned to be added in the upcoming version.		

Table 28: Functional Test Case 28

Test ID	FT-29	**Priority**	High
Test Type	Functional Testing (Component Testing)		
Title	Test Cases for System Configuration		
Procedure	1. Check if administrators can configure system settings 2. Check if administrators can configure AI model parameters 3. Check if administrators can configure notification settings 4. Check if configuration changes are applied correctly 5. Check if configuration changes are logged		
Expected Results	● Administrators can configure system settings ● Administrators can configure AI model parameters ● Administrators can configure notification settings ● Configuration changes are applied correctly ● Configuration changes are logged		
Result	Satisfactory - Tested on 01.05.2025		

Table 29: Functional Test Case 29

Test ID	FT-30	**Priority**	Medium
Test Type	Functional Testing (Component Testing)		
Title	Test Cases for User Preference Configuration		
Procedure	1. Check if radiologists can configure their dashboard layout 2. Check if radiologists can configure notification preferences 3. Check if radiologists can configure display preferences 4. Check if radiologists preferences are saved correctly 5. Check if radiologists preferences are applied correctly after login		
Expected Results	● Users can configure their dashboard layout ● Users can configure notification preferences ● Users can configure display preferences ● User preferences are saved correctly ● User preferences are applied correctly after login		
Result	Satisfactory - Tested on 01.05.2025		

Table 30: FunctionalTest Case 30

Test ID	FT-31	Priority	High
Test Type	Functional Testing (Integration Testing)		
Title	Test Cases for Hospital Information System Integration		
Procedure	1. Check if MRacle can receive patient information from the hospital information system 2. Check if data synchronization works correctly 3. Check if integration handles data format differences correctly 4. Check if integration errors are handled appropriately		
Expected Results	● MRacle can receive patient information from the hospital information system ● Data synchronization works correctly ● Integration handles data format differences correctly ● Integration errors are handled appropriately		
Result	Satisfactory - Tested on 01.05.2025		

Table 31: Functional Test Case 31

Test ID	FT-32	Priority	Medium
Test Type	Functional Testing (Integration Testing)		
Title	Test Cases for Window/Level Adjustment in Viewer		
Procedure	1. Check if the level adjustment controls are available in the viewer UI. 2. Check if the user can modify the window width via slider or numeric input. 3. Check if the user can modify the window level via slider or numeric input. 4. Check if changes to level are applied in real time to the displayed image.		
Expected Results	● Window/level controls (sliders or inputs) are present and enabled. ● Adjusting window width changes image contrast immediately. ● Adjusting window level changes image brightness immediately.		
Result	Satisfactory - Tested on 01.05.2025		

Table 32: Functional Test Case 32

Test ID	FT-33	Priority	High
Test Type	Functional Testing (Installation Testing)		
Title	Test Cases for System Installation		
Procedure	1. Check if installation requirements are clearly documented 2. Check if installation process is straightforward 3. Check if installation validates prerequisites 4. Check if installation completes successfully 5. Check if system functions correctly after installation		
Expected Results	● Installation requirements are clearly documented ● Installation process is straightforward ● Installation validates prerequisites ● Installation completes successfully ● System functions correctly after installation		
Result	Satisfactory - Tested on 01.05.2025		

Table 33: Functional Test Case 33

Test ID	FT-34	Priority	High
Test Type	Functional Testing (Installation Testing)		
Title	Test Cases for System Update		
Procedure	6. Check if update requirements are clearly documented 7. Check if update process preserves existing data 8. Check if update process is reversible (rollback) 9. Check if update completes successfully 10. Check if system functions correctly after update		
Expected Results	● Update requirements are clearly documented ● Update process preserves existing data ● Update process is reversible (rollback) ● Update completes successfully ● System functions correctly after update		
Result	Satisfactory - Tested on 01.05.2025		

Table 34: Functional Test Case 34

Test ID	FT-35	Priority	High
Test Type	Functional Testing (Integration Testing)		
Title	Test Cases for Data Export Functionality		
Procedure	1. Check if segmentation results can be exported in standard formats (NIfTI, DICOM) 2. Check if statistical analysis of tumor measurements can be exported 3. Check if exported data includes proper metadata and annotations 4. Check if export process completes without errors 5. Check if exported data can be imported back or used in other systems		
Expected Results	● Segmentation results can be exported in standard formats ● Statistical analysis can be exported in common formats (CSV, PDF) ● Exported data contains all necessary metadata and annotations ● Export process completes successfully ● Exported data is compatible with other medical imaging systems		
Result	Satisfactory - Tested on 01.05.2025		

Table 35: Functional Test Case 35

Test ID	FT-36	Priority	Critical
Test Type	Functional Testing (Component Testing)		
Title	Test Cases for User Account Management		
Procedure	1. Check if administrators can create new user accounts 2. Check if administrators can deactivate user accounts 3. Check if the system enforces password policies 4. Check if the system provides a secure password reset mechanism		
Expected Results	● Administrators can create new user accounts with appropriate roles ● User accounts can be deactivated without losing associated data ● Password policies are enforced (complexity, expiration, history) ● Password reset mechanism is secure and functional		
Result	Satisfactory - Tested on 01.05.2025		

Table 36: Functional Test Case 36

Test ID	FT-37	Priority	Medium
Test Type	Functional Testing (Component Testing)		
Title	Test Cases for Data Filtering		
Procedure	1. Check if radiologists can filter data lists using single criteria 2. Check if radiologists can filter data lists using multiple criteria 3. Check if filter operations execute within acceptable time limits 4. Check if filter criteria can be saved for future use 5. Check if filtering maintains data integrity		
Expected Results	● Single criterion filtering produces correct results ● Multiple criteria filtering produces correct results ● Filter operations complete within acceptable time ● Filter settings can be saved as presets ● Filtered data maintains integrity and consistency		
Result	Satisfactory - Tested on 01.05.2025		

Table 37: Functional Test Case 37

Test ID	FT-38	Priority	Medium
Test Type	Functional Testing (Component Testing)		
Title	Test Cases for Help System		
Procedure	1. Check if context-sensitive help is available throughout the application 2. Check if help documentation covers all major features 3. Check if help content is searchable 4. Check if help system includes visual aids (screenshots, diagrams) 5. Check if users can access help without losing their current context		
Expected Results	● Context-sensitive help is available and relevant ● Help documentation covers all system features ● Help content can be searched effectively ● Visual aids enhance understanding ● Help system can be accessed without disrupting workflow		
Result	Satisfactory - Tested on 01.05.2025		

Table 38: Functional Test Case 38

Test ID	FT-39	Priority	Medium
Test Type	Functional Testing (Component Testing)		
Title	Test Cases for Data Sorting		
Procedure	1. Check if data lists can be sorted by relevant columns 2. Check if sorting works in both ascending and descending order 3. Check if multiple-level sorting is supported 4. Check if sort order is maintained when navigating away and returning 5. Check if sorting large data sets performs within acceptable limits		
Expected Results	● Data can be sorted by any relevant column ● Both ascending and descending sorts work correctly ● Multiple-level sorting produces expected results ● Sort order persists during navigation ● Sorting large data sets completes efficiently		
Result	Satisfactory - Tested on 01.05.2025		

Table 39: Functional Test Case 39

Test ID	FT-40	Priority	High
Test Type	Functional Testing (Integration Testing)		
Title	Test Cases for Viewer Launch and Patient Context		
Procedure	1. Check if the "Open Viewer" button opens the viewer in a new tab. 2. Check if the URL contains the correct patient ID and study UID. 3. Check if the viewer header shows the correct patient name, ID, and study date. 4. Check if images load within five seconds, with a loading indicator visible until complete. 5. Check if the initial series displayed matches the requested study.		
Expected Results	● Viewer opens in a new tab. ● URL parameters match the selected patient and study. ● Patient details in the header are accurate. ● Images render within five seconds and the indicator disappears. ● The correct series is displayed on launch.		
Result	Satisfactory - Tested on 01.05.2025		

Table 40: Functional Test Case 40

5.2 Non-Functional Test

Test ID	NFT-01	**Priority**	High
Test Type	Non-Functional Testing (Performance Testing)		
Title	Test Cases for AI Analysis Performance		
Procedure	1. Check if AI analysis completes within the specified time limit 2. Check if the system can handle multiple simultaneous analysis requests 3. Check if performance remains stable with increasing database size 4. Check if the system maintains performance during peak usage periods 5. Check if system resources (CPU, memory) are optimally utilized		
Expected Results	● AI analysis completes within specified time limits ● Multiple simultaneous analyses run successfully ● Performance remains stable as database grows ● System performs well during peak usage ● System resources are optimally utilized		
Result	Satisfactory - Tested on 01.05.2025		

Table 41: Non-Functional Test Case 1

Test ID	NFT-02	**Priority**	High
Test Type	Non-Functional Testing (Performance Testing)		
Title	Test Cases for UI Response Time		
Procedure	1. Check if UI responses occur within one second 2. Check if image loading and rendering occurs efficiently 3. Check if navigation between pages is responsive 4. Check if the system provides feedback during long operations 5. Check if the UI remains responsive during background processing		
Expected Results	● UI responses occur within one second ● Images load and render efficiently ● Navigation between pages is responsive ● System provides feedback during long operations ● UI remains responsive during background processing		
Result	Satisfactory - Tested on 01.05.2025		

Table 42: Non-Functional Test Case 2

Test ID	NFT-03	Priority	High
Test Type	Non-Functional Testing (Security Testing)		
Title	Test Cases for Data Encryption		
Procedure	1. Check if patient data is encrypted during transmission 2. Check if patient data is encrypted at rest in storage 3. Check if encryption keys are properly managed 4. Check if encrypted data can only be decrypted by authorized users 5. Check if encryption meets healthcare standards		
Expected Results	● Patient data is encrypted during transmission ● Patient data is encrypted at rest ● Encryption keys are properly managed ● Only authorized users can decrypt data ● Encryption meets healthcare standards (HIPAA, GDPR)		
Result	Planned to be added in the upcoming version.		

Table 43: Non-Functional Test Case 3

Test ID	NFT-04	Priority	Critical
Test Type	Non-Functional Testing (Security Testing)		
Title	Test Cases for Role-Based Access Control		
Procedure	1. Check if users with different roles have appropriate access restrictions 2. Check if unauthorized users cannot access restricted functions 3. Check if system prevents privilege escalation 4. Check if authentication tokens are properly managed 5. Check if session timeouts are implemented for security		
Expected Results	● Different roles have appropriate access restrictions ● Unauthorized users cannot access restricted functions ● System prevents privilege escalation ● Authentication tokens are properly managed ● Session timeouts are implemented		
Result	Satisfactory - Tested on 01.05.2025		

Table 44: Non-Functional Test Case 4

Test ID	NFT-05	Priority	High
Test Type	Non-Functional Testing (Compatibility Testing)		
Title	Test Cases for PACS Compatibility		
Procedure	1. Check if MRacle integrates with different PACS vendors 2. Check if data formats are correctly interpreted from different PACS systems 3. Check if metadata from different PACS systems is correctly mapped 4. Check if integration remains stable after PACS updates 5. Check if error handling is appropriate for PACS integration issues		
Expected Results	● MRacle integrates with different PACS vendors ● Data formats are correctly interpreted ● Metadata is correctly mapped ● Integration remains stable after PACS updates ● Error handling is appropriate for integration issues		
Result	Satisfactory - Tested on 01.05.2025		

Table 45: Non-Functional Test Case 5

Test ID	NFT-06	Priority	High
Test Type	Non-Functional Testing (Reliability Testing)		
Title	Test Cases for Data Backup		
Procedure	1. Check if data backups are performed regularly 2. Check if backups include all necessary data 3. Check if backups are stored securely 4. Check if backup process does not impact system performance 5. Check if backup logs are maintained		
Expected Results	● Data backups are performed regularly ● Backups include all necessary data ● Backups are stored securely ● Backup process does not impact system performance ● Backup logs are maintained		
Result	Planned to be added in the upcoming version.		

Table 46: Non-Functional Test Case 6

Test ID	NFT-07	Priority	High
Test Type	Non-Functional Testing (Reliability Testing)		
Title	Test Cases for Data Recovery		
Procedure	1. Check if data can be recovered from backups 2. Check if recovery process preserves data integrity 3. Check if recovery can be performed for specific components 4. Check if recovery process is documented 5. Check if recovery process is tested regularly		
Expected Results	● Data can be recovered from backups ● Recovery process preserves data integrity ● Recovery can be performed for specific components ● Recovery process is documented ● Recovery process is tested regularly		
Result	Planned to be added in the upcoming version.		

Table 47: Non-Functional Test Case 7

Test ID	NFT-08	Priority	High
Test Type	Non-Functional Testing (Performance Testing)		
Title	Test Cases for System Load Testing		
Procedure	1. Check if the system maintains performance under normal load 2. Check if the system maintains performance under peak load 3. Check if the system handles gradually increasing load appropriately 4. Check if system resources are monitored during load testing 5. Check if performance bottlenecks are identified and documented		
Expected Results	● System maintains performance under normal load ● System maintains performance under peak load ● System handles gradually increasing load appropriately ● System resources are monitored during load testing ● Performance bottlenecks are identified and documented		
Result	Satisfactory - Tested on 01.05.2025		

Table 48: Non-Functional Test Case 8

Test ID	NFT-09	Priority	High
Test Type	Non-Functional Testing (Performance Testing)		
Title	Test Cases for Concurrent User Access		
Procedure	1. Check if multiple users can access the system simultaneously 2. Check if user actions are isolated and do not interfere 3. Check if system performance degrades gracefully under high concurrency 4. Check if database locks are handled correctly 5. Check if session management works correctly with multiple users		
Expected Results	● Multiple users can access the system simultaneously ● User actions are isolated and do not interfere ● System performance degrades gracefully under high concurrency ● Database locks are handled correctly ● Session management works correctly with multiple users		
Result	Satisfactory - Tested on 01.05.2025		

Table 49: Non-Functional Test Case 9

Test ID	NFT-10	Priority	Medium
Test Type	Non-Functional Testing (Accessibility Testing)		
Title	Test Cases for UI Accessibility		
Procedure	1. Check if the UI is keyboard navigable 2. Check if the UI has sufficient color contrast 3. Check if the UI includes appropriate alt text for images 4. Check if the UI is compatible with screen readers 5. Check if the UI complies with accessibility standards		
Expected Results	● UI is keyboard navigable ● UI has sufficient color contrast ● UI includes appropriate alt text for images ● UI is compatible with screen readers ● UI complies with accessibility standards		
Result	Satisfactory - Tested on 01.05.2025		

Table 50: Non-Functional Test Case 10

Test ID	NFT-11	**Priority**	Medium
Test Type	Non-Functional Testing (Accessibility Testing)		
Title	Test Cases for Error Message Accessibility		
Procedure	1. Check if error messages are clearly visible 2. Check if error messages are descriptive 3. Check if error messages suggest corrective actions 4. Check if error messages are non-technical for user errors 5. Check if error messages are accessible to screen readers		
Expected Results	● Error messages are clearly visible ● Error messages are descriptive ● Error messages suggest corrective actions ● Error messages are non-technical for user errors ● Error messages are accessible to screen readers		
Result	Satisfactory - Tested on 01.05.2025		

Table 51: Non-Functional Test Case 11

6. Maintenance Plan and Details

Following deployment, the long-term success of MRacle depends on a comprehensive and sustainable maintenance strategy that addresses the technical, operational, and clinical demands of real-world environments. The system will be actively supported in clinical settings through a combination of scheduled updates, bug resolution, and compatibility maintenance to ensure it continues to function effectively as hospital IT infrastructure evolves. These updates will include feature improvements, critical security patches, and enhancements based on radiologist feedback, all delivered in a controlled and versioned manner to minimize disruption.

The ongoing delivery of services to hospitals is central to our maintenance model. MRacle will provide consistent support through remote assistance channels, enabling hospital IT personnel and radiologists to receive technical help without delays. System documentation and training materials will be regularly updated to reflect the latest system capabilities. Feedback submitted by radiologists through the built-in annotation and reporting tools will be collected, reviewed, and used to inform future development cycles, ensuring the platform continues to align with clinical expectations and workflows.

Maintaining the performance of the AI model is another critical component of the maintenance strategy. MRacle's segmentation model will be closely monitored. As new anonymized imaging data becomes available, especially from diverse institutions, retraining procedures will be conducted to improve generalizability and adapt to changing clinical practices. This process will include incorporating user feedback and updated medical standards to refine both the accuracy and reliability of tumor detection and segmentation. Prior to the release of new model versions, thorough validation will be performed to confirm compliance with clinical safety benchmarks.

MRacle is built to be both scalable and integration-ready to support future adoption at a larger scale. The modular, container-based architecture enables deployment across various hospital environments with minimal infrastructure adjustments.

The system also incorporates robust health monitoring and error-logging mechanisms. MRacle will continuously track the status of core components, such as API availability, segmentation engine performance, and data storage health. Any critical issues, including failures in analysis or system communication, will be logged and flagged for immediate resolution. Logs will be stored centrally and used not only for debugging but also to anticipate potential system bottlenecks and prevent downtime before it affects users.

Through this comprehensive approach to maintenance, MRacle aims to deliver a reliable, secure, and adaptable solution that can continue to serve the needs of radiologists and patients over time.

7. Other Project Elements

7.1 Consideration of Various Factors in Engineering Design

7.1.1 Constraints

7.1.1.1 Economic Constraints

As developers, all the frameworks and libraries we used were free, some of which we accessed by verifying our status as current students. However, we had to personally cover the cost of purchasing the web domain name.

7.1.1.2 User Experience and Usability Constraints

Given the nature of our product, users are expected to be informed of technical terminology related to MR scans and medical data. However, since our users are radiologists, this is not an issue we expect.

The application must be intuitive, allowing users without extensive technical knowledge to utilize its features effectively.

7.1.2 Standards

During the development of MRacle, we followed several important standards to guarantee the system's reliability and safety. These included compliance with HIPAA (Health Insurance Portability and Accountability Act) to protect the privacy and security of patient information. Additionally, we adhered to technical standards for AI in healthcare, as defined by organizations such as the IEEE (Institute of Electrical and Electronics Engineers) and ISO (International Organization for Standardization), which establish guidelines for the safe, effective, and ethical use of artificial intelligence in medical applications.

7.2 Ethics and Professional Responsibilities

The ethical considerations surrounding AI in healthcare are significant. In developing MRacle, we placed a strong emphasis on ethical principles such as ensuring transparency in AI decision-making processes and upholding the right to human oversight, particularly in cases where the AI's diagnosis may differ from that of medical professionals. A core objective throughout the project was to support and enhance the diagnostic process without undermining the critical role of human judgment in medical practice. Additionally, by encrypting all user data, we safeguard patient privacy and confidentiality.

7.3 Teamwork Details

7.3.1 Contributing and Functioning Effectively on the Team

We organized ourselves into three subgroups to ensure effective collaboration and optimal use of everyone's skills: the backend development team, the frontend development team, and the AI model development team. Our tasks were divided into these subgroups, which allowed us to work efficiently. We established weekly Zoom meetings where each subgroup reported their progress and discussed the next steps. These meetings ensured everyone remained informed about the development.

7.3.2 Helping Create a Collaborative and Inclusive Environment

To create a collaborative and inclusive environment, we encourage open communication and active participation from all team members. We used platforms like Slack and GitHub to share updates, ask questions, and provide feedback. Team members were encouraged to express their ideas and concerns freely, ensuring that everyone's input was valued. By promoting mutual respect and support, we created a workspace where everyone felt comfortable contributing and working towards our common goal.

7.3.3 Taking Lead Role and Sharing Leadership on the Team

Leadership responsibilities were distributed across the team to ensure balanced workload management and decision-making. Each subgroup had a designated lead who coordinated tasks, facilitated discussions, and addressed challenges. However, leadership was flexible, and team members took initiative when needed, whether by solving technical issues, proposing new ideas, or assisting others. This approach ensured that leadership was a shared responsibility rather than being limited to a single person. By working together and stepping up when necessary, we strengthened teamwork and maintained steady project progress.

7.3.4 Meeting Objectives

As mentioned above, throughout the development of the project, we conducted regular weekly team meetings. These meetings were essential for maintaining effective communication, aligning our efforts with project objectives, and fostering collaborative problem-solving. They served as key checkpoints to assess progress, address emerging challenges, and develop strategic solutions together. By setting clear agendas and goals for each meeting, we ensured efficient use of time while keeping all team members informed and actively involved. These sessions played a crucial role in promoting transparency, accountability, and team cohesion, ultimately helping us reach our project milestones and deliverables.

7.4 New Knowledge Acquired and Applied

Throughout the development of MRacle, our team acquired and applied a wide range of knowledge spanning artificial intelligence, medical imaging, and healthcare systems integration.

From a technical standpoint, we gain an understanding of medical data formats such as DICOM and NIfTI, learning how to parse, convert, and visualize them effectively in clinical workflows. We also gained experience with PACS systems and DICOMweb protocols, which are essential for building scalable medical software solutions.

In the field of AI, we became proficient in medical image segmentation by working extensively with the nnU-Net framework and the BraTS 2023 adult glioma dataset. We learned how to prepare data for training, interpret model outputs, and assess segmentation accuracy using domain-specific metrics like the Dice metric.

We implemented full-stack development practices on the software engineering side, including RESTful API design with FastAPI, WebSocket integration, secure authentication, and real-time event handling. We also practiced docker containerization and deployment using tools compatible with hospital IT infrastructure.

Finally, collaborating with radiologists and potential stakeholders taught us how to translate clinical requirements into user-friendly software features. We recognized the importance of explainability, performance, and trust in AI-assisted diagnosis tools, which informed many design decisions.

8. Conclusion and Future Work

In conclusion, MRacle provides practical solutions to challenges in the current healthcare system in the domain of brain tumor diagnosis. It addresses high costs, time inefficiencies, and diagnostic inconsistencies by integrating AI models into the clinical workflow. By enhancing the coordination between radiologists and AI, MRacle reduces the burden on medical professionals while maintaining high diagnostic accuracy. For reliable deployment in Türkiye, additional tuning using Türkiye-specific medical imaging data is recommended.

In terms of machine learning advancements, gathering a Turkish dataset is crucial to improve model accuracy and facilitate its use in Türkiye. This dataset should exclude any irrelevant components, such as visible parts of screening machines in the images. Moreover, anonymization techniques such as defacing and skull-stripping will be implemented to increase safety and privacy. To expand the dataset while maintaining privacy, federated learning techniques can be implemented in the future.

From a marketing perspective, we started discussions with Medicana Hospital and are considering a potential collaboration with them. This will allow us to gather feedback and improve the development of MRacle through the insights of radiologists and experts in the medical imaging field. Moreover, discussions with doctors and industry experts revealed a strong demand for medical second opinion services on brain tumor risk analysis. By focusing on MRacle, we aim to expand our customer base gradually.

9. Glossary

Term	Definition
MRI	Magnetic Resonance Imaging, a medical imaging technique used to form pictures of the anatomy and physiological processes of the body.
AI	Artificial Intelligence, the simulation of human intelligence processes by computer systems.
PACS	Picture Archiving and Communication System, a medical imaging technology for storing, retrieving, and distributing medical images.
DICOM	Digital Imaging and Communications in Medicine, a standard for handling, storing, and transmitting medical imaging information.
NIfTI	Neuroimaging Informatics Technology Initiative, a file format designed for storing neuroimaging data.
UI	User Interface, the means by which users interact with the system.
REST	Representational State Transfer, an architectural style for designing networked applications.
API	Application Programming Interface, a set of rules that allow different software applications to communicate with each other.
HTTP	Hypertext Transfer Protocol, the foundation of data communication on the World Wide Web.
FLAIR	Fluid Attenuated Inversion Recovery, a type of MRI sequence that suppresses fluid signals.
T1-CE	T1-weighted Contrast-Enhanced, an MRI sequence that uses contrast agents to highlight specific tissues.
GDPR	General Data Protection Regulation, a regulation on data protection and privacy in the European Union.
HIPAA	Health Insurance Portability and Accountability Act, US legislation providing data privacy and security provisions for safeguarding medical information.
FDA	Food and Drug Administration, a federal agency responsible for protecting public health by ensuring safety and efficacy of medical products.
SaMD	Software as a Medical Device, software intended to be used for medical purposes without being part of a hardware medical device.
MongoDB	A cross-platform document-oriented NoSQL database program that uses JSON-like documents with optional schemas.

Table 52: Definitions, Acronyms, and Abbreviations Table

10. References

[1] V. N. V. L. S. Swathi, K. Sinduja, V. Ravi Kumar, A. Mahendar, G. V. Prasad, and B. Samya, "Deep learning-based brain tumor detection: An MRI segmentation approach," *MATEC Web Conf.*, vol. 392, p. 01157, 2024.

[2] Yashu and V. Kukreja, "Explainable AI in medical imaging: A hybrid CNN and random forest framework for brain tumor detection," in *2025 International Conference on Automation and Computation (AUTOCOM)*, 2025, pp. 1662–1667.

[3] L. Tang, J. Li, and S. Fantus, "Medical artificial intelligence ethics: A systematic review of empirical studies," *Digit. Health*, vol. 9, p. 20552076231186064, 2023.

[4] "NIfTI: — neuroimaging informatics technology initiative," *Nih.gov*, 21-Jan-2005. [Online]. Available: https://nifti.nimh.nih.gov. [Accessed: 01-May-2025].

[5] "Current edition," *DICOM*. [Online]. Available: https://www.dicomstandard.org/current. [Accessed: 01-May-2025].

[6] "Connect your modality to Orthanc — Orthanc Book documentation," *Uclouvain.be*. [Online]. Available: https://orthanc.uclouvain.be/book/integrations/modality.html. [Accessed: 01-May-2025].