CS492: Senior Design Project II

Detailed Design 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

Detecting brain tumors early and accurately is crucial for giving patients the best chance of recovery. Delays or mistakes in diagnosing can cause serious problems, making treatments less effective and affecting quality of life. However, the imbalance between the number of radiologists and the total workload emphasizes the need for tools like **MRacle**.

MRacle is an innovative AI-based solution designed to address these challenges and change how brain tumors are detected and diagnosed. While considering ethical and professional issues, by analyzing images quickly and accurately, **MRacle** helps identify potential tumors and marks affected areas for more straightforward review using advanced neural networks. Furthermore, **MRacle** helps radiologists concentrate on the most critical patients by prioritizing those more likely to have tumors. **MRacle** aims to improve diagnostic accuracy, enhance patient outcomes, and help radiologists without replacing their role.

1.1 Purpose of the System

The MRacle system provides computational support for radiologists in brain tumor detection through MRI scan analysis. The application implements an AI-powered platform with a standardized interface designed for clinical environments. Its primary function is to assist radiologists in managing high volumes of MRI scans by providing preliminary analysis results, including tumor region identification and case prioritization based on detected abnormalities.

The system addresses workflow inefficiencies in radiology departments where radiologists may be required to analyze hundreds of scans daily. By performing initial detection and segmentation of potential tumor regions, MRacle enables radiologists to allocate their expertise more efficiently while maintaining their role in final diagnosis and decision-making.

1.2 Design Goals

1.2.1 Usability

- Implement an intuitive interface aligned with radiologists' existing workflows.
- Create a consistent and user-friendly experience throughout the system.
- Design tasks to minimize radiologists' efforts, allowing focus on core responsibilities.
- Ensure application responsiveness with appropriate user feedback for successful operations or errors.

1.2.2 Reliability

- Maintain consistent performance and accuracy in AI analyses.
- Ensure high system uptime with reliable error-handling mechanisms.
- Implement protocols to prevent and recover from failures.

- Conduct regular maintenance and updates to address emerging issues promptly.

1.2.3 Performance

- Deliver high-performance processing, completing AI analyses within a short interval per scan.
- Maintain user interface response times under one second for standard interactions.
- Optimize resource utilization for efficient operation.

1.2.4 Scalability

- Support simultaneous use by multiple users.
- Design the system to handle growing numbers of users and increasing data volumes.
- Accommodate increases in patients and data without sacrificing speed or functionality.
- Implement containerization to facilitate deployment across multiple servers.
- Ensure load distribution and service availability during demand fluctuations.

1.2.5 Privacy and Security

- Focus on the sensitivity of medical data with healthcare-specific compliance.
- Implement decisive measures to protect patient privacy and confidentiality.
- Secure data during transmission and storage to minimize unauthorized access risks.
- Deploy role-based access controls (RBAC) for authorization management.
- Require multi-factor authentication (MFA) for secure logins.
- Apply anonymization where possible to protect patient identities.

1.2.6 Marketability

- Design the system to address verified clinical needs in radiological workflows.
- Develop compatibility with standard hospital infrastructure and technical environments.
- Ensure compliance with regional medical device regulations and standards.
- Implement cost-effective deployment options for various healthcare facility sizes.

1.2.7 Aesthetics

- Implement a clean, professional interface appropriate for clinical environments.
- Utilize a consistent, restrained color palette conforming to medical visualization standards.
- Provide dark mode functionality to reduce eye strain during extended usage periods.

1.3 Definitions, Acronyms, and Abbreviations

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.
RBAC	Role-Based Access Control, an approach to restricting system access to authorized users based on roles.
MFA	Multi-Factor Authentication, a security process requiring users to provide two or more verification factors to gain access.
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 1: Definitions, Acronyms, and Abbreviations Table

1.4 Overview

MRacle is a web-based application for radiological analysis of brain MRI scans using artificial intelligence models. The system integrates with hospital PACS infrastructure to acquire and process MRI data, generating analysis results highlighting regions of interest and assigning priority levels to cases. The application addresses operational inefficiencies in radiological workflows, particularly in healthcare environments with high MRI scan volumes relative to available specialist resources. By providing computational support for initial image analysis and case organization, the system enables a more efficient allocation of radiologists while maintaining human intervention in diagnostic processes.

2. Current Software Architecture

In Türkiye, the current healthcare system presents a need for MRacle to be met. While 52 MRI scans are performed per 1000 people per year in OECD countries, this number is 119 per 1000 people in Türkiye. However, the number of radiologists in Türkiye is significantly low, with 15 per 100,000 people in OECD countries. This number is only 5 per 100,000 people in Türkiye [1]. In other words, while our country performs more than twice as many MRIs as OECD countries, the number of radiologists is one-third of OECD countries. Therefore, in Türkiye, radiologists may have to report up to 300 daily examinations, and this causes them to have a limited time for examining MRs while also being consumed.

Given this problem's nature, efforts were also made to solve it. In 2020, a product named Türk-Beyin was developed as part of a project in collaboration with Gazi University and the Cumhurbaşkanlığı Dijital Dönüşüm Ofisi. The product integrates with the PACS system at Gazi University Hospital, analyzing all brain MRIs and notifying radiologists when abnormalities are detected [2]. However, the system is not currently being used despite its potential, likely due to licensing or certification challenges. There are no other known projects in Türkiye besides this project, and this lack of integration creates a significant opportunity for MRacle.

3. Proposed Software Architecture

3.1 Overview

MRacle implements a layered architecture pattern designed to process and analyze brain MRI scans while maintaining security standards for clinical data. The architecture comprises four layers with clearly defined responsibilities, enabling modular development, maintainability, and system scalability.

The Repository Layer is the data access abstraction, interfacing with MongoDB to provide persistent storage for patient information, analysis results, notifications, and system logs. This layer encapsulates data access logic. The Service Layer encapsulates the core logic, managing patient data, MRI image acquisition, AI-based analysis, case prioritization, notifications, and feedback processing. The Controller Layer represents RESTful endpoints, enabling client-server communication, handling authentication, patient data operations, analysis requests, and feedback collection. Controllers validate incoming requests, route them to appropriate services, and format responses according to API specifications. The UI Layer provides role-specific interfaces: radiologists can access diagnostic tools for MRI analysis, while administrators manage system configuration and user access controls.

External integration with hospital PACS systems enables access to MRI scans in standard medical formats, including DICOM and NIfTI, in the hospital system.

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.

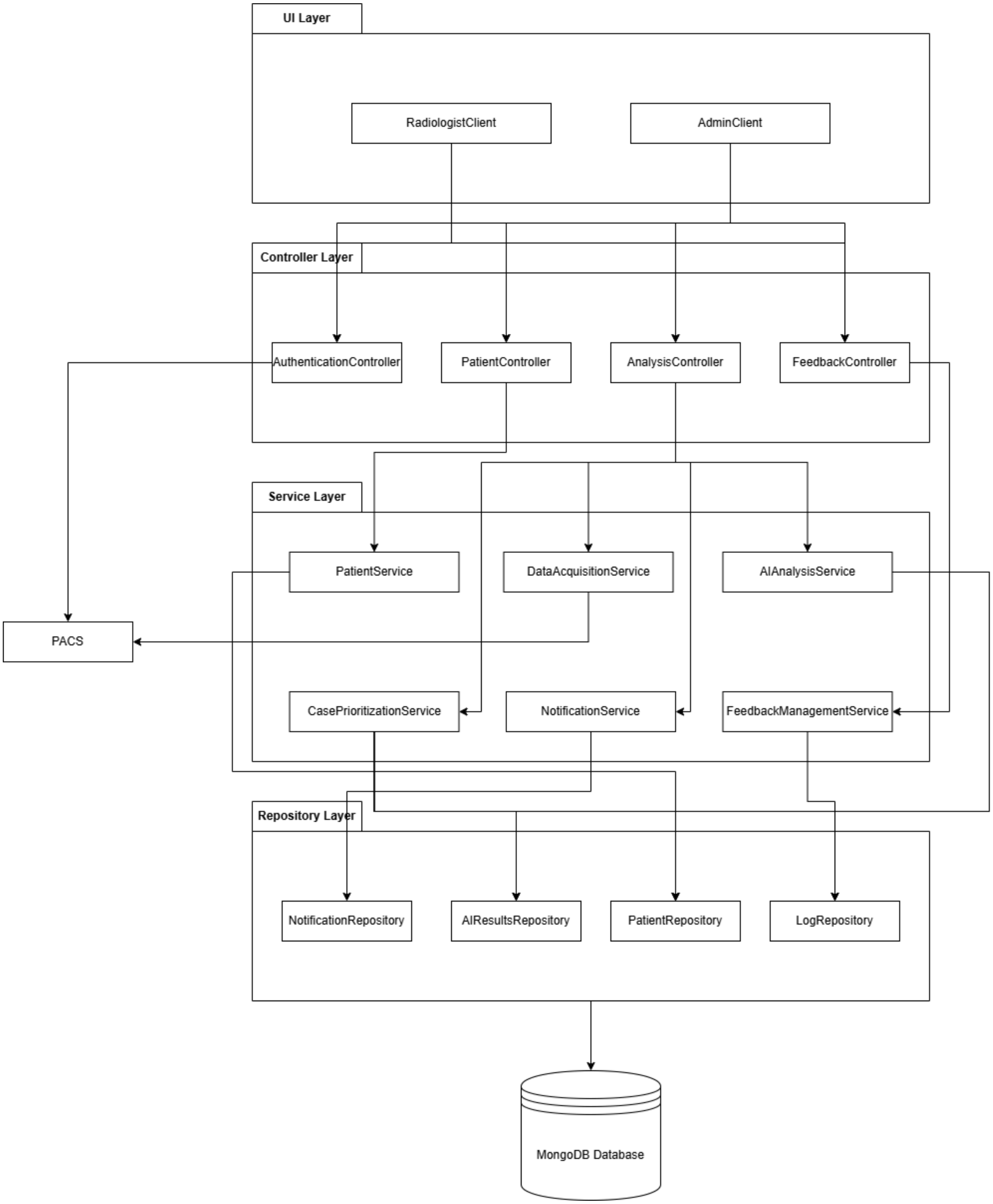

Figure 1: Subsystem Decomposition Diagram

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.

3.5 Access Control and Security

MRacle implements a security framework aligned with healthcare regulatory requirements. The authentication system leverages existing PACS authentication, allowing radiologists to use their established credentials. This integration eliminates redundant login procedures while maintaining security standards. The authorization framework uses role-based access control with permissions assigned based on clinical responsibilities. Radiologists can access assigned patient data and analysis tools while administrators manage system configuration and user accounts. Access limitations can be further defined based on department, specialization, and patient assignment to ensure appropriate data access.

Security measures include standard protocols for API authentication, secure data transmission, and activity tracking for all system operations. Session management includes appropriate timeouts and invalidation mechanisms to prevent unauthorized access.

4. Subsystem Services

4.1. UI Layer

- **RadiologistClient**: Implements the radiologist interface, visualizing patient data, MRI scans, AI-generated analysis results, prioritized cases, and feedback submission capabilities.
- **AdminClient**: Implements the administrator interface, providing system management capabilities, including user account administration, performance monitoring, and configuration management.

4.2. Controller Layer

- **AuthenticationController**: Manages user identity verification, token generation/validation, and session control. Processes login requests, validates credentials against PACS stored values, and enforces role-based access restrictions to maintain security.
- **PatientController**: Exposes endpoints for patient record operations, including medical history access and MRI scan metadata.
- **AnalysisController**: Process requests related to MRI scan processing, including analysis initiation, result retrieval, and segmentation data access. Coordinates with AIAnalysisService to execute analysis pipelines and deliver results to client applications.
- **FeedbackController**: Manages the collection of radiologist input regarding AI analysis accuracy, capturing corrections, confirmations, and improvement suggestions. Routes feedback data to appropriate services for storage and model refinement.

4.3. Service Layer

- **PatientService**: Implements patient record management operations, including medical history tracking and MRI scan management.
- **DataAcquisitionService**: Facilitates MRI scan retrieval from PACS systems, performs format validation, and prepares imaging data for analysis. Supports multiple MRI sequence types and standard medical imaging formats, ensuring compatibility with hospital infrastructure.
- **AIAnalysisService**: Starts the pipeline for brain tumor analysis, including image preprocessing, AI model inference, segmentation, and probability calculations.
- **CasePrioritizationService**: Implements algorithmic prioritization of cases based on tumor probability scores and clinical urgency factors. Enables efficient radiologist workflow by identifying high-risk cases requiring immediate attention.
- **NotificationService**: Manages alert generation and delivery to radiologists regarding new scans, high-priority cases, and analysis completion. Supports preference management.

- **FeedbackManagementService**: Processes radiologist input on AI analysis accuracy, storing annotations, corrections, and quality assessments.

4.4. Repository Layer

- **NotificationRepository**: Implements operations for notification data, including notification information, content, delivery status, and acknowledgment tracking.
- **AIResultsRepository**: Manages analysis outputs, including detection results, segmentation data, probability metrics, and associated metadata. Implements efficient storage and retrieval mechanisms for volumetric imaging data.
- **PatientRepository**: Implements storage and retrieval operations for patient records, including clinical history and MRI scan history. Enforces data protection mechanisms and access controls for sensitive clinical information.
- **LogRepository**: Provides storage for system events, user actions, and error conditions to enhance troubleshooting.

4.5. External Systems and Storage

- **PACS Integration**: Establishes connectivity with hospital imaging infrastructure to access MRI scans via standard medical protocols. Implements adapters for DICOM communication and supports integration with diverse vendor systems like MR machines.
- **MongoDB Database**: Provides storage for system data, selected for its schema flexibility, scalability, and performance characteristics suited to medical imaging applications.

5. Test Cases

5.1 Functional Tests

Test ID	FT-01
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
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
Priority	Critical

Table 2: Functional Test Case 1

Test ID	FT-02
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 ● The connection remains stable during the session ● Appropriate error messages are displayed when authentication fails
Priority	Critical

Table 3: Functional Test Case 2

Test ID	FT-03
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
Priority	High

Table 4: Functional Test Case 3

Test ID	FT-04
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
Priority	High

Table 5: Functional Test Case 4

Test ID	FT-05
Test Type	Functional Testing (Integration Testing)
Title	Test Cases for AI Analysis Initiation
Procedure	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
Expected Results	● 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
Priority	Critical

Table 6: Functional Test Case 5

Test ID	FT-06
Test Type	Functional Testing (Integration Testing)
Title	Test Cases for AI Analysis Results Display
Procedure	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
Expected Results	● 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
Priority	High

Table 7: Functional Test Case 6

Test ID	FT-07
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
Priority	High

Table 8: Functional Test Case 7

Test ID	FT-08
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
Priority	High

Table 9: Functional Test Case 8

Test ID	FT-09
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
Priority	High

Table 10: Functional Test Case 9

Test ID	FT-10
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
Priority	High

Table 11: Functional Test Case 10

Test ID	FT-11
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
Priority	High

Table 12: Functional Test Case 11

Test ID	FT-12
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)
Priority	Minor

Table 13: Functional Test Case 12

Test ID	FT-13
Test Type	Functional Testing (Integration Testing)
Title	Test Cases for System Maintenance
Procedure	1. Check if administrators can access the system maintenance page 2. Check if administrators can view system logs 3. Check if administrators can apply software updates 4. Check if administrators can perform database maintenance 5. Check if maintenance activities are logged for audit purposes
Expected Results	● Administrators can access the system maintenance page ● System logs are viewable ● Software updates can be applied ● Database maintenance can be performed ● Maintenance activities are logged for audit purposes
Priority	High

Table 14: Functional Test Case 13

Test ID	FT-14
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 logs can be exported for external analysis 5. 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 ● Logs can be exported ● Critical errors are highlighted or flagged
Priority	High

Table 15: Functional Test Case 14

Test ID	FT-15
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
Priority	High

Table 16: Functional Test Case 15

Test ID	FT-16
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
Priority	High

Table 17: Functional Test Case 16

Test ID	FT-17
Test Type	Functional Testing (Integration Testing)
Title	Test Cases for MRI Preprocessing
Procedure	1. Check if image alignment works correctly 2. Check if resampling to uniform voxel size works correctly 3. Check if skull-stripping works correctly 4. Check if normalization works correctly 5. Check if preprocessing results are suitable for AI analysis
Expected Results	● Image alignment works correctly ● Resampling to uniform voxel size works correctly ● Skull-stripping works correctly ● Normalization works correctly ● Preprocessing results are suitable for AI analysis
Priority	Critical

Table 18: Functional Test Case 17

Test ID	FT-18
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
Priority	Critical

Table 19: Functional Test Case 18

Test ID	FT-19
Test Type	Functional Testing (Integration Testing)
Title	Test Cases for AI Segmentation Accuracy
Procedure	1. Check if the AI model correctly segments tumor regions 2. Check if the segmentation matches expert annotations (using Dice similarity coefficient) 3. Check if the model performs consistently across different tumor types 4. Check if the model performs consistently across different MRI qualities 5. Check if the model handles edge cases correctly
Expected Results	● AI model correctly segments tumor regions ● Segmentation accuracy meets defined thresholds ● Model performs consistently across tumor types ● Model performs consistently across MRI qualities ● Model handles edge cases correctly
Priority	Critical

Table 20: Functional Test Case 19

Test ID	FT-20
Test Type	Functional Testing (Integration Testing)
Title	Test Cases for Tumor Likelihood Scoring
Procedure	1. Check if the model 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	● Model 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
Priority	Critical

Table 21: Functional Test Case 20

Test ID	FT-21
Test Type	Functional Testing (Integration Testing)
Title	Test Cases for Report Generation
Procedure	1. Check if reports include all necessary information 2. Check if reports include segmentation images 3. Check if reports include tumor likelihood scores 4. Check if reports can be downloaded in appropriate formats 5. Check if reports are correctly associated with patient records
Expected Results	● Reports include all necessary information ● Reports include segmentation images ● Reports include tumor likelihood scores ● Reports can be downloaded in appropriate formats ● Reports are correctly associated with patient records
Priority	High

Table 22: Functional Test Case 21

Test ID	FT-22
Test Type	Functional Testing (Integration Testing)
Title	Test Cases for Report Editing
Procedure	1. Check if radiologists can edit report content 2. Check if radiologists can add annotations to images 3. Check if radiologists can save draft reports 4. Check if radiologists can finalize reports 5. Check if report versioning is maintained
Expected Results	● Radiologists can edit report content ● Radiologists can add annotations to images ● Radiologists can save draft reports ● Radiologists can finalize reports ● Report versioning is maintained
Priority	High

Table 23: Functional Test Case 22

Test ID	FT-23
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
Priority	High

Table 24: Functional Test Case 23

Test ID	FT-24
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
Priority	High

Table 25:Functional Test Case 24

Test ID	FT-25
Test Type	Functional Testing (Integration Testing)
Title	Test Cases for Tumor Change Detection
Procedure	1. Check if the system can compare current and previous scans 2. Check if tumor growth or shrinkage is detected 3. Check if changes are quantified and displayed 4. Check if the comparison results are stored 5. Check if significant changes trigger notifications
Expected Results	● System can compare current and previous scans ● Tumor growth or shrinkage is detected ● Changes are quantified and displayed ● Comparison results are stored ● Significant changes trigger notifications
Priority	High

Table 26: Functional Test Case 25

Test ID	FT-26
Test Type	Functional Testing (Integration Testing)
Title	Test Cases for Historical Scan Navigation
Procedure	1. Check if users 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 5. Check if users can compare multiple historical scans side by side
Expected Results	● Users can navigate between historical scans ● Scan dates are clearly displayed ● Analysis results for historical scans are accessible ● Scans are displayed in chronological order ● Users can compare multiple historical scans side by side
Priority	High

Table 27: Functional Test Case 26

Test ID	FT-27
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
Priority	High

Table 28: Functional Test Case 27

Test ID	FT-28
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
Priority	High

Table 29: Functional Test Case 28

Test ID	FT-29
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
Priority	High

Table 30: Functional Test Case 29

Test ID	FT-30
Test Type	Functional Testing (Component Testing)
Title	Test Cases for User Preference Configuration
Procedure	1. Check if users can configure their dashboard layout 2. Check if users can configure notification preferences 3. Check if users can configure display preferences 4. Check if user preferences are saved correctly 5. Check if user 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
Priority	Medium

Table 31: FunctionalTest Case 30

Test ID	FT-31
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 MRacle can send analysis results to the hospital information system 3. Check if data synchronization works correctly 4. Check if integration handles data format differences correctly 5. Check if integration errors are handled appropriately
Expected Results	● MRacle can receive patient information from the hospital information system ● MRacle can send analysis results to the hospital information system ● Data synchronization works correctly ● Integration handles data format differences correctly ● Integration errors are handled appropriately
Priority	High

Table 32: Functional Test Case 31

Test ID	FT-32
Test Type	Functional Testing (Integration Testing)
Title	Test Cases for Email Notification Integration
Procedure	1. Check if the system can send email notifications 2. Check if email notifications contain correct information 3. Check if email notifications are sent on appropriate triggers 4. Check if email notification failures are handled appropriately 5. Check if email notification settings can be configured
Expected Results	● System can send email notifications ● Email notifications contain correct information ● Email notifications are sent on appropriate triggers ● Email notification failures are handled appropriately ● Email notification settings can be configured
Priority	Medium

Table 33: Functional Test Case 32

Test ID	FT-33
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
Priority	High

Table 34: Functional Test Case 33

Test ID	FT-34
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
Priority	High

Table 35: Functional Test Case 34

Test ID	FT-35
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
Priority	High

Table 36: Functional Test Case 35

Test ID	FT-36
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 modify existing user permissions 3. Check if administrators can deactivate user accounts 4. Check if the system enforces password policies 5. Check if the system provides a secure password reset mechanism
Expected Results	● Administrators can create new user accounts with appropriate roles ● User permissions can be modified and applied immediately ● User accounts can be deactivated without losing associated data ● Password policies are enforced (complexity, expiration, history) ● Password reset mechanism is secure and functional
Priority	Critical

Table 37: Functional Test Case 36

Test ID	FT-37
Test Type	Functional Testing (Component Testing)
Title	Test Cases for Data Filtering
Procedure	1. Check if users can filter data lists using single criteria 2. Check if users 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
Priority	Medium

Table 38: Functional Test Case 37

Test ID	FT-38
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
Priority	Medium

Table 39: Functional Test Case 38

Test ID	FT-39
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
Priority	Medium

Table 40: Functional Test Case 39

Test ID	FT-40
Test Type	Functional Testing (Integration Testing)
Title	Test Cases for Session Management
Procedure	1. Check if user sessions timeout after the configured idle period 2. Check if users are warned before session timeout 3. Check if sessions can be manually extended 4. Check if multiple concurrent sessions for the same user are handled appropriately 5. Check if session data is properly cleaned up after logout
Expected Results	● Sessions timeout after configured idle period ● Timeout warnings are displayed before session expiration ● Users can extend sessions when prompted ● Multiple concurrent sessions are managed according to policy ● Session data is properly cleared after logout
Priority	High

Table 41: Functional Test Case 40

5.2 Non-Functional Tests

Test ID	NFT-01
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
Priority	High

Table 42: Non-Functional Test Case 1

Test ID	NFT-02
Test Type	Non-Functional Testing (Performance Testing)
Title	Test Cases for UI Response Time
Procedure	1. Check if UI responses occur within one second as specified in requirements 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
Priority	High

Table 43: Non-Functional Test Case 2

Test ID	NFT-03
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)
Priority	High

Table 44: Non-Functional Test Case 3

Test ID	NFT-04
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
Priority	Critical

Table 45: Non-Functional Test Case 4

Test ID	NFT-05
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
Priority	High

Table 46: Non-Functional Test Case 5

Test ID	NFT-06
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
Priority	High

Table 47: Non-Functional Test Case 6

Test ID	NFT-07
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
Priority	High

Table 48: Non-Functional Test Case 7

Test ID	NFT-08
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
Priority	High

Table 49: Non-Functional Test Case 8

Test ID	NFT-09
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
Priority	High

Table 50: Non-Functional Test Case 9

Test ID	NFT-10
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
Priority	Medium

Table 51: Non-Functional Test Case 10

Test ID	NFT-11
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
Priority	Medium

Table 52: Non-Functional Test Case 11

6. Consideration of Various Factors in Engineering Design

6.1 Considerations

6.1.1 Public Health Considerations

MRacle is an application that aims to facilitate current health practices in analyzing MR scans of patients and detecting probable tumors through this process. Therefore, MRacle considers health as a sensitive and critical point, as any wrong diagnosis and treatment might have severe effects on the patients. While the radiologists conclude the results, the algorithms and models developed in the application will be tested in detail, ensuring their reliability in diverse scenarios. To increase the accuracy of the process, MRacle will put a strong emphasis on utilizing the most up-to-date and diverse data in the algorithms to be compatible.

6.1.2 Public Safety Considerations

Ensuring the safety of patient data is crucial. MRacle will not permit the sharing or publicization of personal data and personal information with third parties. Internally, the data will be stored and encrypted to prevent unauthorized access. Since the diagnoses of the patients and their MR scans will be managed in the application, it is sensitive that we protect those data to guarantee public safety and confidence in the users of the MRacle platform. Additionally, the accuracy and reliability of AI analyses are continuously validated to prevent misdiagnoses that could affect patient health.

6.1.3 Global Considerations

While MRacle initially targets the Turkish healthcare market, We are designing it with the idea of it being used in different regions. Therefore, the system will comply with international data protection standards, with the help of various experts globally.

6.1.4 Social Considerations

As MRacle will handle sensitive patient data, social and ethical considerations can be raised. So to safeguard privacy, the platform will use encrypted and secure storage solutions.

Additionally, MRacle will not utilize existing patient data for training its machine-learning models, and it will certainly not have any hidden data collection mechanisms within the platform. The medical data required for model training and developing a knowledge base will be collected from institutions and experts with proper permits.

6.1.5 Environmental Considerations

MRacle contributes to environmental sustainability by reducing the time and resources needed for MR scans within healthcare facilities (both machines and radiologists). By helping with the diagnostic process, MRacle reduces the need for repetitive MR scans and minimizes the associated consumption and waste.

6.1.6 Economic Considerations

A maintenance-based system is planned during the integration of MRacle with the healthcare industry. The radiologists working in the medical centers will not pay to use the platform. However, medical centers must pay a predetermined annual fee to use the system.

6.2 Constraints

6.2.1 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.

6.2.2 Privacy and Security Constraints

- Role-Based Access Controls (RBAC) guarantee that only authorized healthcare professionals can access patient data.
- All interactions and data collected from the hospitals and medical centers will be handled with the data protection law General Data Protection Regulation (GDPR) [3].

6.2.3 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.

6.3 Engineering Standards

6.3.1 Digital Imaging and Communications in Medicine (DICOM)

DICOM is a widely adopted standard for handling, storing, and transmitting medical imaging information [6]. We plan to implement DICOM in our project to ensure seamless integration across different devices and systems.

6.3.2 Neuroimaging Informatics Technology Initiative (NIfTI)

NIfTI is a standard format designed explicitly for neuroimaging data [5]. Our project aims to utilize NIfTI to maintain consistency and compatibility in the processing and analysis of neuroimaging datasets.

6.3.3 IEEE 830-1998: Software Requirements Specifications

This standard provides guidelines for preparing clear and structured software requirements specifications [7]. In our project, we adhere to IEEE 830-1998 to document functional and non-functional requirements.

6.3.4 FDA Regulations for Software as a Medical Device (SaMD)

These regulations outline the requirements for developing and maintaining software classified as a medical device [8]. Our project aims to ensure compliance with FDA regulations to guarantee the safety and effectiveness of the software components designated as medical devices.

6.3.5 Health Insurance Portability and Accountability Act (HIPAA)

HIPAA mandates safeguards for the protection of sensitive patient data [4]. In our project, we plan to implement HIPAA-compliant measures to ensure patient information's confidentiality, integrity, and availability.

6.3.6 General Data Protection Regulation (GDPR)

GDPR provides a regulatory framework for data protection and privacy within the European Union [3]. Our project plans to incorporate GDPR-compliant protocols to securely process and store personal health information, ensuring compliance with EU data protection laws.

6.3.7 Unified Modeling Language (UML) 2.5.1

UML 2.5.1 is a standardized modeling language utilized for system design and documentation [9]. In our project, we employed UML to create various diagrams, including class, activity, state, sequence, and system architecture diagrams, ensuring a consistent representation of system functionalities.

Factor	Effect Degree	Comment
Public Health	9	MRacle must adopt the latest best practices and utilize the most current data throughout the algorithm development.
Public Safety	10	As the application is integrated into the medical industry, it must take various factors into account during its design to ensure public safety and prevent the publicization or sharing of personal and medical data.
Global Factors	5	MRacle will be designed for scalability and adaptability for global deployment.
Social Factors	9	MRacle must prioritize patient privacy in its design and will avoid hidden data collection mechanisms as well as training on patient data
Environmental Factors	4	MRacle will focus on the accuracy of the processes to reduce wasteful consumption of healthcare resources.
Economic Factors	7	MRacle should emphasize economic considerations and arrange monetization to provide inclusiveness.

Table 2: Engineering Considerations of Factors

7. Teamwork Details

7.1 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 organization into these subgroups 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.

7.2 Helping Creating a Collaborative and Inclusive Environment

To foster a collaborative and inclusive environment, we encouraged 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 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.

8. Glossary

Term	Definition
AI	Artificial Intelligence
MRI	Magnetic Resonance Imaging
DICOM	Digital Imaging and Communications in Medicine
NIfTI	Neuroimaging Informatics Technology Initiative
FDA	Food and Drug Administration
OECD	Organisation for Economic Co-operation and Development
SaMD	Software as a Medical Device
PACS	Picture Archiving and Communication System
GDPR	General Data Protection Regulation
HIPAA	Health Insurance Portability and Accountability
RBAC	Role-Based Access Controls
MFA	Multi-Factor Authentication
UML	Unified Modeling Language
T2-FLAIR	T2 Fluid Attenuated Inversion Recovery
T1-CE	T1 Contrast-Enhanced

9. References

[1] M. Erşan, “Radyolog isyanı: 1 dakikada MR görüntüleri izlenip rapor yazılamaz... Hataya itiyorlar,” Hürriyet.com.tr, 14-Nov-2018. [Online]. Available: https://www.hurriyet.com.tr/gundem/radyolog-isyani-1-dakikada-mr-goruntuleri-izlenip-rapor-yazilamaz-hataya-itiyorlar-41018425. [Accessed: 21-Nov-2024].

[2] “Türk Beyin Projesi (TBP),” Türkiye Cumhuriyeti Cumhurbaşkanlığı Dijital Dönüşüm Ofisi. [Online]. Available: https://cbddo.gov.tr/projeler/tbp/. [Accessed: 21-Nov-2024].

[3] “General Data Protection Regulation (GDPR) – legal text,” General Data Protection Regulation (GDPR). [Online]. Available: https://gdpr-info.eu/. [Accessed: 21-Nov-2024].

[4] “The HIPAA Privacy Rule,” HHS.gov. [Online]. Available: https://www.hhs.gov/hipaa/for-professionals/privacy/index.html. [Accessed: 21-Nov-2024].

[5] “NIfTI: — Neuroimaging Informatics Technology Initiative,” NIH.gov, 21-Jan-2005. [Online]. Available: https://nifti.nimh.nih.gov/. [Accessed: 21-Nov-2024].

[6] “Current edition,” DICOM. [Online]. Available: https://www.dicomstandard.org/current. [Accessed: 21-Nov-2024].

[7] “About the Unified Modeling Language Specification Version 2.5.1,” OMG.org. [Online]. Available: https://www.omg.org/spec/UML/2.5.1/About-UML. [Accessed: 21-Nov-2024].

[8] “Software as a Medical Device (SaMD),” FDA.gov. [Online]. Available: https://www.fda.gov/medical-devices/digital-health-center-excellence/software-medical-device-samd. [Accessed: 21-Nov-2024].

[9] “IEEE recommended practice for software requirements specifications,” IEEE, Piscataway, NJ, USA, 2008.