**NeuroScan: Advanced Brain Tumor Detection System**

**Software Requirements Specification**

Version 1.0



**Group Id: S2402CCFFE**

**Supervisor Name : Umair Ali**

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date (dd/mm/yyyy)** | **Version** | **Description** | **Author** |
| **02-05-2024** | **1.0** | **NeuroScan: Advanced Brain Tumor Detection System** | **Student Id:- BC220422733** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Table of Contents**

1. [Scope (of the project)](#scope)
2. [Functional Requirements & Non-Functional Requirements](#FRNFR)
3. [Use Case Diagram](#UCD)
4. [Usage Scenarios](#UCS)
5. [Adopted Methodology](#Adopted)
6. [Work Plan (Use MS Project to create Schedule/Work Plan)](#Gantt)

**Software Requirements Specification (SRS) Document**

**Project Title: - NeuroScan: Advanced Brain Tumor Detection System**

**1. Scope of Project: -**

NeuroScan aims to develop an advanced brain tumor detection system using cutting-edge technology in medical imaging and machine learning. The system will facilitate accurate and efficient diagnosis of brain tumors, allowing healthcare professionals to make informed treatment decisions. The scope of the project includes:

1. - Development of a user-friendly interface for uploading, processing, and analyzing medical images.
2. - Integration of machine learning algorithms for automated tumor detection and classification.
3. - Implementation of advanced visualization tools for displaying diagnostic results.
4. - Testing and validation of the system to ensure accuracy and reliability in clinical settings.
5. - Deployment of the system in healthcare facilities for use by radiologists and neurologists.

**2. Functional and non-Functional Requirements: -**

Here is the detail about the functional and non-functional requirements of our system in separate headings.

1. **Functional Requirements: -**
2. - **User Authentication:** The system should allow authorized users to log in securely.
3. - **Image Upload:** Users should be able to upload medical images (e.g., MRI scans) for analysis.
4. - **Image Processing:** The system should preprocess uploaded images to enhance quality and remove artifacts.
5. - **Tumor Detection:** Automated algorithms should detect and locate tumors within uploaded images.
6. - **Tumor Classification:** Detected tumors should be classified based on characteristics such as size, shape, and location.
7. - **Diagnostic Reporting:** The system should generate comprehensive reports summarizing diagnostic findings.
8. - **Visualization Tools:** Advanced visualization tools should enable users to view and analyze diagnostic results effectively.
9. - **Integration with PACS:** The system should integrate with Picture Archiving and Communication Systems (PACS) for seamless image management.
10. - **Security:** The system should adhere to strict security protocols to protect patient data and maintain confidentiality.
11. **Non-Functional Requirements:**
12. **- Performance:** The system should be capable of processing large medical images efficiently within reasonable time frames.
13. - **Accuracy:** Tumor detection algorithms should achieve high levels of accuracy and sensitivity to minimize false positives and negatives.
14. - **Usability:** The user interface should be intuitive and user-friendly, requiring minimal training for healthcare professionals.
15. - **Reliability:** The system should be robust and reliable, with minimal downtime and maximum uptime.
16. - **Scalability:** The system should be scalable to accommodate increasing numbers of users and image data.
17. **- Compliance:** The system should comply with relevant regulations and standards for medical devices and healthcare IT systems.
18. **- Interoperability:** The system should be interoperable with existing healthcare IT infrastructure and standards.
19. **- Data Security:** Patient data should be encrypted during transmission and storage to ensure data security and privacy.
20. **- Accessibility:** The system should be accessible to users with disabilities, conforming to accessibility standards.

**Diagram(s):**

***Comments: Use relationships like extend or include between use-cases where required.***

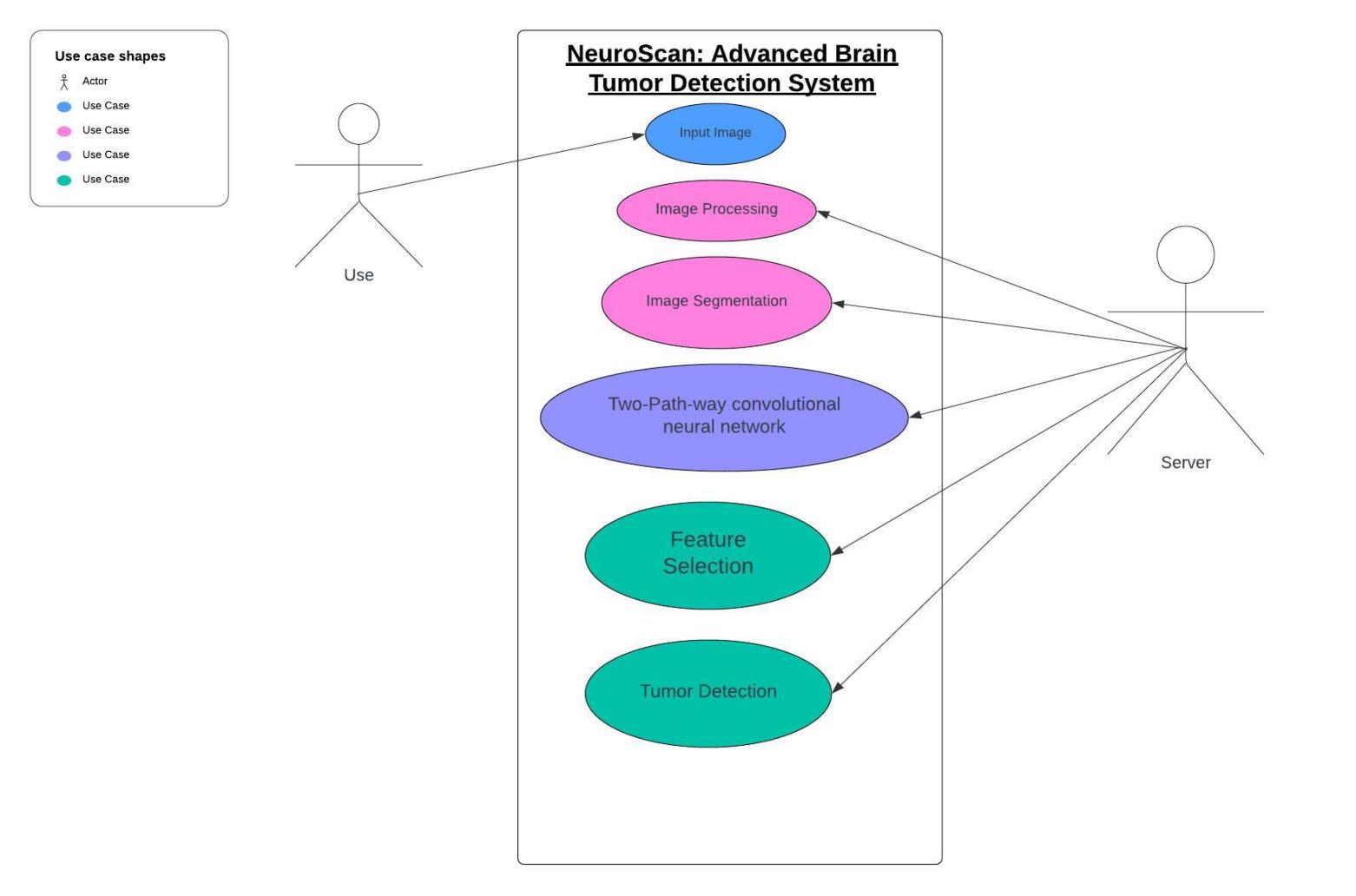
***Comments: Update use case diagram according to actor’s activity***

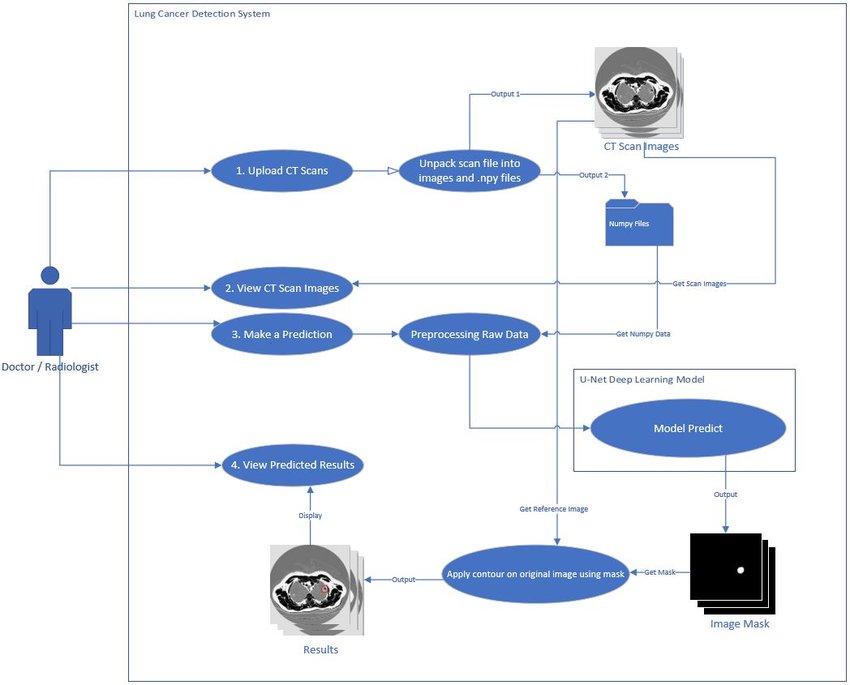
***Comments: Use UML notations.***

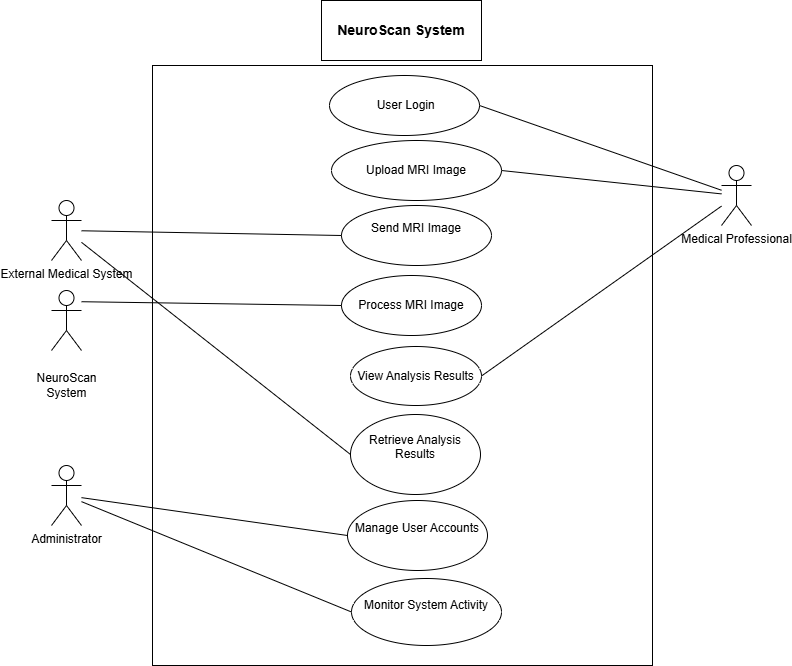
***Comments: Add missing actors.***

***Comments: Remove duplicate use cases.***

***=============================================***





#

**Usage Scenarios:**

***Comments: Update usage scenario of each use case according to your updated use case diagram.***

***Comments: Provide usage scenario of each use case given in use case diagram.***

***=============================================***

**--**

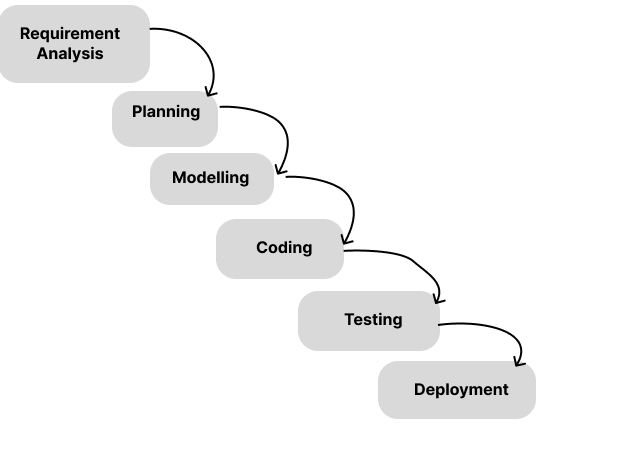
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Use Case Title** | **Use Case ID** | **Actions** | **Description** | **Alternative Paths** | **Pre-Conditions** | **Author** | **Exceptions** |
| User Authentication | UC001 | 1. Enter username and password   2. Click “Login” | Authenticate user credentials against database. | Invalid credentials --> Display error message | User is not logged in | System Analyst | User account is locked |
| Image Upload | UC002 | 1. Select medical image file   2. Click “Upload” | Upload medical image for analysis | Invalid file format --> Display error message | User is logged in | System Analyst | Image upload fails |
| Image Processing | UC003 | Pre-process uploaded image | Enhance image quality and remove artifacts | N/A | Image is uploaded successfully | System Analyst | Image processing fails |
| Tumor Detection | UC004 | Analyze processed image | Detect and locate tumors within the image | No tumors detected --> Display message | Tumors are detected successfully | System Analyst | Tumor detection algorithms fails |
| Tumor Classification | UC005 | Classify detected tumors | Classify tumors based on characteristics | N/A | Tumors are classified successfully | System Analyst | Tumor classification fails |
| Diagnostic Reporting | UC006 | Generate diagnostic report | Summarize diagnostic findings and recommendations | N/A | Diagnostic report is generated successfully | System Analyst | Report generation fails |
| Visualization Tools | UC007 | View diagnostic results | Use advanced visualization tools to analyze | N/A | Results are visualized successfully | System Analyst | Visualization tools fail to load |

**Adopted Methodology**

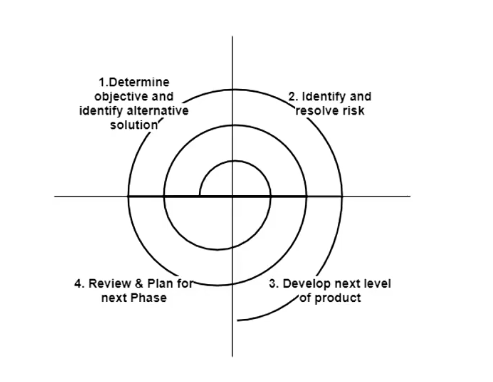
The project will follow the VU Process Model, which combines elements of the Waterfall and Spiral models. The Waterfall model provides a structured approach to development, with distinct phases for planning, analysis, design, implementation, testing, and maintenance. The Spiral model incorporates iterative cycles of development, allowing for feedback and refinement throughout the project lifecycle. This hybrid approach ensures both thorough planning and flexibility to adapt to changing requirements and feedback.

**Waterfall Methodology:-**

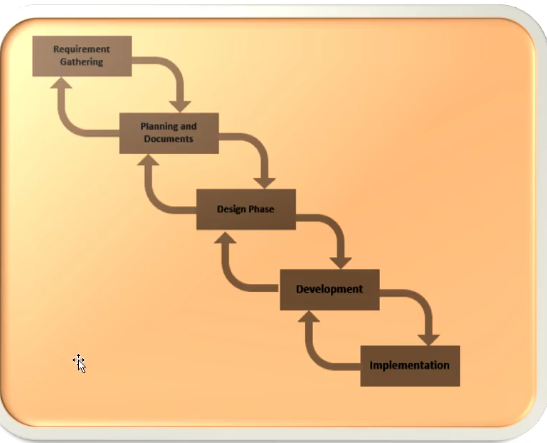
Waterfall methodology consists of anaysis, design, coding and testing. The system requirement concerns with system users and classified as software and hardware. The system give s complete construction of the system.



**Spiral Methodology:-**



***Diagram of VU Process model***



**Work Plan (Use MS Project to create Schedule/Work Plan)**

