Chapter 1: Introduction and Background

1.1 Introduction:

In the ever-evolving field of educational technologies and medical diagnostics, the integration of artificial intelligence and machine learning has opened new possibilities for enhancing diagnostic accuracy and efficiency. Cogniosis represents a significant step forward in this context, leveraging the power of ViT to transform the way brain strokes are detected from CT images.

Traditional methods of stroke diagnosis often rely on the expertise of highly trained radiologists and involve labor-intensive and time-consuming processes. These limitations can delay diagnosis and treatment, particularly in regions with limited access to specialized medical personnel. By contrast, Cogniosis offers a dynamic and automated solution, utilizing advanced machine learning techniques to provide real-time analysis and accurate diagnostic results.

The application is built on a robust technological foundation, utilizing Flutter for cross-platform development and Firebase for secure backend services. The integration of TensorFlow Lite enables on-device inference, allowing for rapid analysis of CT images without the need for constant internet connectivity. This ensures that Cogniosis can be used in a variety of settings, from urban hospitals to remote clinics, providing a valuable tool for healthcare providers worldwide.

The educational content included in the app further enhances its utility, offering users valuable information on brain stroke symptoms, prevention, and treatment. By raising awareness and empowering individuals with knowledge, Cogniosis aims to contribute to better public health outcomes and improve the overall quality of healthcare delivery.

1. 2 Background History:

The evolution of computer-aided diagnosis (CAD) systems marks a significant milestone in the realm of medical imaging. Over the decades, various algorithms and techniques have been employed to enhance the accuracy and efficiency of detecting abnormalities in medical images. Initially, these systems relied on basic image processing techniques and classical machine learning algorithms, which required extensive preprocessing and the expertise of radiologists to interpret the results.

With the advent of deep learning, a new paradigm in image recognition emerged, offering unprecedented accuracy and automation capabilities. Convolutional Neural Networks (CNNs) became the cornerstone of modern image analysis, enabling automatic feature extraction and classification from raw images. However, despite their success, CNNs had limitations in capturing long-range dependencies within images, prompting the exploration of more advanced models.

The ViT represents a breakthrough in this context, utilizing self-attention mechanisms to capture global context and perform exceptionally well on various image recognition tasks. By transforming the approach to medical image analysis, ViTs offer a robust solution for detecting brain strokes from CT scans. Cogniosis builds on these advancements, integrating a ViT model trained on a comprehensive dataset of brain stroke CT images from Kaggle. This approach ensures high accuracy and reliability, setting a new standard in the early detection of strokes.

1. 3 Statement of Problem Area:

Brain strokes represent one of the most severe and potentially fatal medical conditions, often resulting in profound disability or death if not promptly diagnosed and treated. The early and accurate detection of brain strokes through imaging techniques, particularly CT scans, is crucial for improving patient outcomes and providing timely medical intervention. However, traditional methods of analyzing CT scans are labor-intensive, time-consuming, and require the expertise of highly trained radiologists, which can be a significant barrier in regions with limited access to specialized medical personnel. This gap underscores the need for innovative solutions that can facilitate rapid and accurate diagnosis, thereby enhancing healthcare delivery.

Enter **Cogniosis**, a cutting-edge mobile application designed to harness the power of advanced machine learning algorithms to detect brain strokes from CT images efficiently. By leveraging a robust and sophisticated Vision Transformer (ViT) model, Cogniosis aims to democratize stroke diagnosis, providing an accessible, reliable, and user-friendly tool for healthcare providers across diverse settings. The app is meticulously crafted to offer real-time analysis and accurate results, bridging the gap in stroke diagnosis and enabling timely medical intervention, thus potentially saving lives and reducing the long-term impact of strokes on patients.

1. 4 Previous and Current Work:

The journey of leveraging machine learning for medical image analysis has seen numerous research projects and commercial applications, each contributing to the field's evolution. Early systems primarily utilized feature extraction and classical machine learning algorithms, which, despite their foundational role, required significant manual intervention and expert knowledge for accurate interpretation.

Recent advancements have ushered in the era of deep learning, where models like CNNs have shown remarkable promise in automating image recognition tasks. However, the inherent limitations of CNNs in capturing global dependencies prompted the exploration of more advanced models, leading to the development of ViTs. They excel in their ability to process the entire image as a sequence of patches, capturing long-range dependencies and providing superior accuracy.

Cogniosis stands on the shoulders of these giants, integrating a ViT model trained on an extensive Kaggle dataset consisting of approximately 2500 brain stroke CT images categorized into normal and stroke classes. This rigorous training process ensures that the model achieves an impressive accuracy rate of 96%, making it a reliable tool for stroke detection. By converting the trained model into TensorFlow Lite, we have seamlessly integrated it into the app, enabling real-time on-device inference and rapid analysis of CT images.

1. 5 Project Description:

Cogniosis is an innovative brain stroke detection application developed using Flutter, a powerful cross-platform framework that ensures consistent performance across various devices. The application leverages the ViT model, renowned for its exceptional accuracy in image recognition tasks, to detect brain strokes from CT images. Trained on a comprehensive dataset from Kaggle, the model can accurately classify images into normal and stroke categories, achieving an impressive accuracy rate of 96%.

The app's backend is powered by Firebase, providing a robust infrastructure for user authentication, real-time notifications, and secure data storage. Users can register and log in using their email or Google accounts, with plans to expand authentication options to include

Facebook and Apple ID in the future. The user interface is designed to be intuitive and user-friendly, featuring a splash screen, login screen, registration screen, homepage with educational content, and a contact us section. Users can easily upload or capture CT images for analysis, preview the selected images, and receive detailed results.

The homepage provides an overview of brain stroke symptoms, prevention, and treatment, along with a graph depicting patient statistics over the last six years. This educational content aims to raise awareness about brain strokes and empower users with knowledge to recognize early symptoms. Additionally, the app features a "Contact Us" section, allowing users to reach out for support or inquiries, further enhancing user engagement and support.

1. 6 Purpose:

The primary purpose of Cogniosis is to revolutionize the early detection of brain strokes by providing a reliable, accessible, and efficient diagnostic tool. By leveraging advanced machine learning techniques, Cogniosis aims to assist healthcare providers in making timely and accurate diagnoses, ultimately improving patient outcomes and reducing the long-term impact of strokes. The application is designed to be user-friendly, ensuring that medical practitioners can easily integrate it into their diagnostic workflow, regardless of their technical expertise.

In addition to its diagnostic capabilities, Cogniosis seeks to raise awareness about brain stroke symptoms, prevention, and treatment through its educational content. By providing users with valuable information and resources, the app aims to empower individuals to recognize early signs of stroke and seek prompt medical attention. This dual approach of diagnosis and education positions Cogniosis as a comprehensive tool for improving public health and enhancing the quality of healthcare delivery.

1. 7 Objectives:

The objectives of Cogniosis are multifaceted, reflecting the app's commitment to providing a comprehensive and user-centric solution for brain stroke detection. These objectives include:

Accuracy: Achieve high accuracy in detecting brain strokes from CT images using the Vision Transformer model, ensuring reliable diagnostic results.

- **Accessibility**: Provide an easy-to-use mobile application that can be accessed by healthcare providers in various settings, enhancing the reach and impact of the tool.
- **Efficiency**: Enable quick analysis of CT images, reducing the time required for diagnosis and facilitating timely medical intervention.
- **User Engagement**: Offer educational content on brain stroke symptoms, prevention, and treatment to increase user awareness and empower individuals with knowledge.
- **Security**: Ensure user data is securely stored and managed through Firebase's robust backend services, adhering to best practices in data security and privacy.
- **Scalability**: Design the application to support future enhancements and expansions, such as additional authentication methods and expanded educational content, ensuring long-term sustainability and relevance.

1. 8 Scope:

The scope of Cogniosis encompasses the development and deployment of a mobile application capable of detecting brain strokes from CT images with high accuracy and efficiency. The application includes several core functionalities:

- **User Authentication**: Secure login and registration using email or Google accounts, with future plans to include Facebook and Apple ID authentication.
- **Image Upload and Analysis**: Users can upload or capture CT images for analysis, with the ViT model providing real-time diagnostic results.
- **Result Display**: Detailed analysis results are displayed, including a classification of the image as normal or stroke, along with additional relevant information.
- **Educational Content**: The homepage features information on brain stroke symptoms, prevention, and treatment, as well as a graph of patient statistics over the last six years.
- **Contact Us Section**: Users can reach out for support or inquiries through the app's contact form.
- **User Interface**: Intuitive and user-friendly design, featuring a splash screen, login screen, registration screen, homepage, and contact us section.

1. 9 Tools and Technologies Used:

Table 1: Tools and Technologies Used

Tools and Technologies used	
1	Android studio
2	Flutter Language for App Development
3	Kaggle for Dataset and Model training
4	Tensorflow Lite
3	Firebase Database

1. 10 Deployment Platform:

Table 2: Deployment Platform

Operating System	**Architecture**
Android	Android 7.0 or later

1. 11 Project Scheduling:

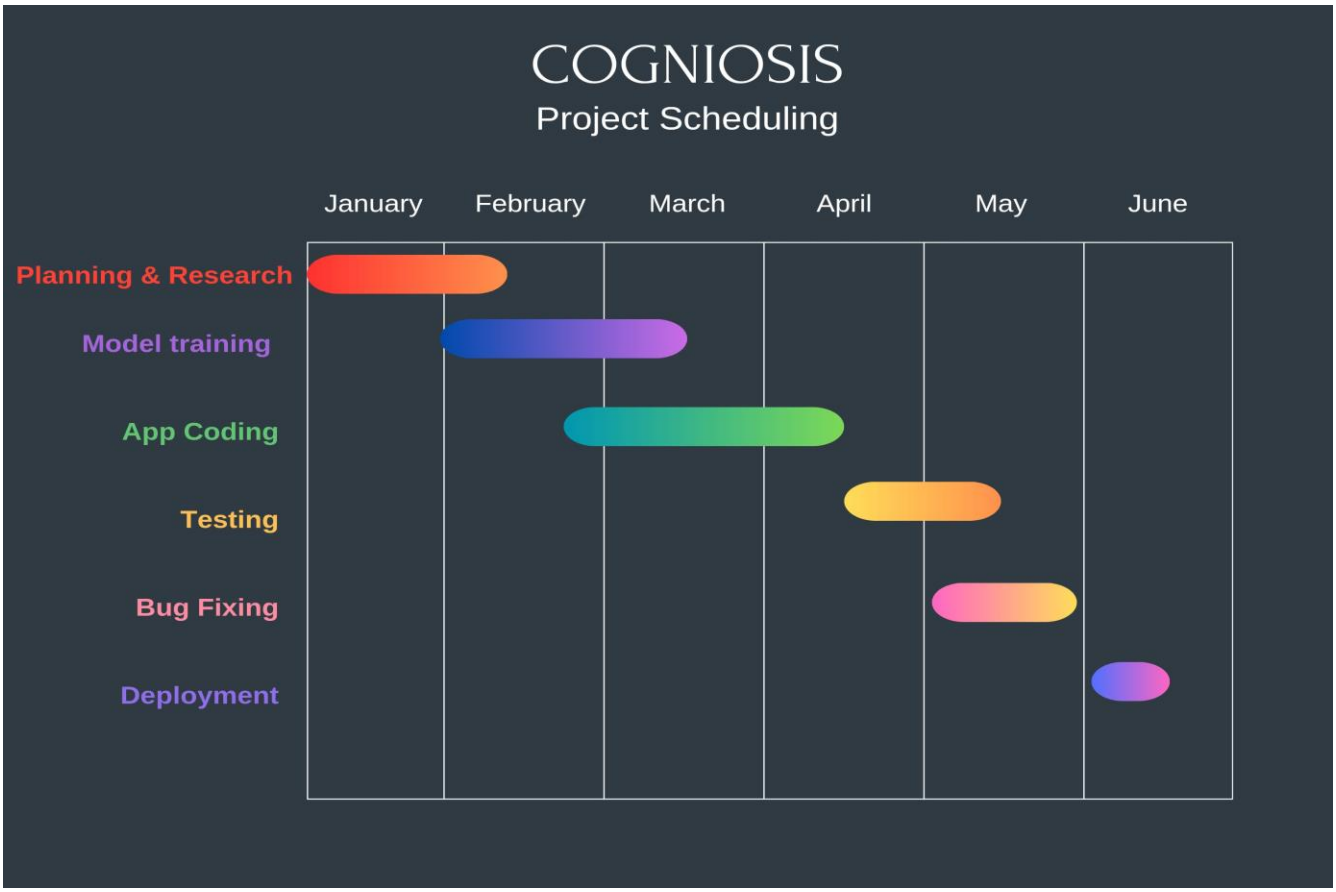

Figure 1Project Scheduling Diagram

Chapter 2: Software Requirements Specifications

2. 1 Functional Requirements:

The functional requirements for **Cogniosis** outline the core functionalities that ensure the application delivers a seamless and comprehensive user experience. These requirements are essential to meet the needs of healthcare providers and to ensure the app performs as expected. The functional requirements are as follows:

- **User Authentication and Management**:

  Users must be able to register and log in using their email addresses or Google accounts. The registration process should include fields for first name, last name, email, about me, address, mobile number, date of birth, and password.

- **Image Upload and Capture**:

  Users should be able to upload CT images from their device or capture new images using the device's camera. The app must support various image formats commonly used in medical imaging.

- **Real-Time Image Analysis**:

  The app should process the uploaded or captured CT images using the integrated ViT. It Provide real-time feedback on whether the image indicates a normal condition or a stroke.

- **Results Display**:

  After analysis, display detailed results to the user, including a clear indication of whether the image shows a normal brain or a stroke.

- **Educational Content**:

  The homepage should feature educational content on brain stroke symptoms, prevention, and treatment. Include a graph showing the statistics of patients affected by brain strokes over the past six years to provide context and raise awareness.

- **User Profile Management**:

  Allow users to view and update their profile information from the "My Profile" screen. Profile updates should include fields such as first name, last name, email, about me, address, mobile number, and date of birth.

- **Navigation and User Interface**:

  The app should have a user-friendly interface with a splash screen, login screen, registration screen, homepage, and contact us section. The homepage should include a slider at the top and an "Ask Us" card that navigates to the contact us screen. Include a header menu with options for "My Profile," "FAQs," "About Us," and "Logout."

- **Notifications**:

  Implement a notification system using Firebase to inform users about important updates, analysis results, and reminders.

- **Customer Support**:

  Provide a "Contact Us" section where users can reach out for support or inquiries. Include options for contacting support via email or through an in-app contact form.

- **FAQs and About Us**:

  Include an "FAQs" screen to address common questions and provide detailed explanations of the app's functionalities. The "About Us" screen should offer insights into the development team, the mission and vision of Cogniosis, and the importance of early stroke detection.

2. 2 Nonfunctional Requirements:

Nonfunctional requirements address the performance, quality attributes, and constraints of the application, ensuring that **Cogniosis** not only functions correctly but also provides a superior user experience. These requirements include:

- **Performance**:

  The app must exhibit fast loading times and smooth transitions between screens to ensure a responsive and engaging user experience. Real-time image analysis should be performed efficiently, with minimal latency.

- **Reliability**:

Ensure the app is stable and performs consistently across different devices and operating systems. Implement rigorous testing and error handling to prevent crashes and data loss.

- **Security**:

  Protect user data through encrypted communications and secure storage practices.

- **Usability**:

  Design the user interface to be intuitive and accessible, catering to users of all ages and technical abilities. Provide clear instructions and feedback to guide users through the app's functionalities.

- **Compatibility**:

  Ensure compatibility across various devices, particularly Android. Test the app on multiple hardware configurations to ensure consistent performance.

- **Maintainability**:

  Adopt a modular code structure to facilitate easy updates and feature enhancements. Maintain comprehensive documentation to support ongoing development and maintenance efforts.

- **Scalability**:

  Design the app to handle an increasing number of users and data without degradation in performance.Use scalable backend services to support growth and ensure reliability.

2. 3 Project/Product Feasibility Report:

The feasibility report for **Cogniosis** assesses the viability of the project across several dimensions, ensuring that it is technically, operationally, legally, economically, and schedule-wise feasible. This comprehensive assessment includes:

2. 3. 1 Technical Feasibility:

Technical feasibility evaluates the availability and suitability of the technologies required for the project. The analysis includes:

- **Flutter**:

  Selected for its ability to create high-performance cross-platform mobile applications.
- **Firebase**:

  Provides robust backend services, including authentication, real-time notifications, and secure data storage.
- **TensorFlow Lite**:

  Facilitates the deployment of the ViT model on mobile devices for real-time inference.
- **Dataset from Kaggle**:

  Ensures that the model is trained on a diverse and representative set of images, enhancing its accuracy and reliability.

2. 3. 2 Operational Feasibility:

Operational feasibility examines the app's ability to meet user needs and integrate seamlessly into their daily routines. This analysis includes:

- **User Needs and Preferences**: Gathering insights through surveys and feedback sessions to ensure the app meets the expectations of healthcare providers.
- **User Interface Design**: Ensuring an intuitive and user-friendly design that facilitates easy navigation and interaction.
- **Support Features**: Providing FAQs and contact forms to assist users with common issues and questions.
- The analysis confirms that **Cogniosis** is well-suited to meet user needs and function effectively within its intended environment.

2. 3. 3 Legal & Ethical Feasibility:

Legal and ethical feasibility assesses the app's compliance with relevant laws and ethical guidelines. This includes:

- **Data Protection and Privacy Laws**:

  Ensuring that user data is handled securely and ethically.
- **Intellectual Property Rights**:

Verifying that the app does not infringe on existing patents, trademarks, or copyrights.

- **Ethical Guidelines**:

  Ensuring that the app promotes inclusivity and accessibility without causing harm or discrimination.

2. 3. 4 Economic Feasibility:

Economic feasibility assesses the financial viability of developing and maintaining the app. This includes:

- **Cost-Benefit Analysis**:

  Estimating the expenses associated with development, marketing, and ongoing maintenance, as well as potential returns and benefits.

- **Revenue Streams**:

  Identifying potential revenue sources, such as app sales, subscriptions, and in-app purchases.

- **Funding Sources**:

  Exploring grants and investments to support the project's financial needs.

2. 3. 5 Schedule Feasibility:

Schedule feasibility assesses the project's timeline, ensuring that development can be completed within the stipulated period. This includes:

- **Project Scheduling**:

  Creating a detailed project schedule outlining each phase of development, from research and design to coding, testing, and deployment.

- **Milestones and Deadlines**:

  Setting specific milestones and deadlines for each task to ensure timely completion.

- **Risk Management**:

  Developing contingency plans to address potential risks and challenges that may impact the timeline.

This analysis confirms that **Cogniosis** can be developed and delivered on time, with a high likelihood of meeting all deadlines and milestones.

2. 3. 6 Motivational Feasibility:

Motivational feasibility assesses the team's commitment and drive to complete the project successfully. This includes:

- **Team Motivation**:

  Evaluating the team's passion for the project and their willingness to invest the necessary time and effort.

- **Incentives and Rewards**:

  Ensuring that team members are adequately motivated and recognized for their contributions.

2. 3. 7 Information Feasibility:

Information feasibility assesses the availability and quality of the data and resources required to develop and support the app. This includes:

- **Datasets**:

  Ensuring that the training datasets are comprehensive, diverse, and representative of real-world scenarios.

- **Technical Documentation**:

  Providing detailed documentation and tutorials for the chosen technologies.

- **User Feedback**:

  Incorporating user insights and feedback into the development process to ensure the app meets their needs.This analysis confirms that **Cogniosis** has access to the necessary data and resources to support successful development and ongoing improvement.

2. 3. 8 Specification Feasibility:

Specification feasibility assesses the clarity, completeness, and achievability of the project's requirements. This includes:

- **Requirement Review**:

  Ensuring that the functional and nonfunctional requirements are well-defined and aligned with the project's goals.

- **Stakeholder Consultations**:

  Gathering input and feedback from stakeholders to ensure all requirements are accurately captured and prioritized.

- **Technical Feasibility**:

  Verifying that the chosen technologies can support the desired functionalities and performance attributes.

Chapter 3: System Performance Requirements

3. 1 Efficiency:

Efficiency is a cornerstone of **Cogniosis**, ensuring that the application operates seamlessly and responds promptly to user interactions. Achieving high efficiency involves several key aspects:

- **Fast Loading Times**: The application must load quickly, minimizing wait times for users. This is particularly crucial for healthcare providers who rely on swift access to diagnostic tools in time-sensitive situations.
- **Smooth Transitions**: Transitions between different screens and functionalities within the app should be smooth and fluid. This enhances user experience and reduces frustration caused by lag or delays.
- **Real-Time Processing**: The real-time image analysis feature, powered by TensorFlow Lite, must process CT images rapidly, providing users with immediate feedback. This capability is essential for maintaining engagement and ensuring that healthcare providers can make quick, informed decisions.
- **Resource Optimization**: The app should be optimized to use minimal system resources, ensuring that it performs well even on devices with lower specifications. This includes efficient memory management and minimal battery consumption.
- **Backend Efficiency**: Firebase backend services must be optimized for fast data retrieval and storage, supporting real-time notifications and user authentication without delays.

By focusing on these efficiency aspects, **Cogniosis** ensures a responsive and engaging user experience, critical for its success as a diagnostic tool.

3. 2 Reliability:

Reliability is paramount in **Cogniosis**, guaranteeing consistent performance and minimizing downtime to facilitate uninterrupted use in clinical settings. Key reliability requirements include:

- **Stability**: The app must perform reliably across various devices and operating systems. Rigorous testing should be conducted to identify and resolve any issues that may cause crashes or unexpected behavior.

- **Error Handling**: Implement robust error handling mechanisms to manage and recover from unexpected issues gracefully. This includes providing informative error messages and automatic recovery procedures.
- **Data Integrity**: Ensure that user data, including diagnostic results and profile information, is accurately stored and retrieved. Implement mechanisms to prevent data corruption and loss.
- **Redundancy**: Incorporate redundancy in backend services to ensure that the app remains functional even if some components fail. This may include multiple data centers and failover mechanisms for critical services.
- **Testing and Validation**: Conduct comprehensive testing, including unit tests, integration tests, and stress tests, to validate the app's reliability under various conditions and loads.

By ensuring high reliability, **Cogniosis** builds trust among users, making it a dependable tool for brain stroke detection.

3. 3 Security:

Security is foundational in **Cogniosis**, safeguarding user data and interactions against unauthorized access and malicious activities. Security measures include:

- **Data Encryption**: Encrypt all sensitive data, both in transit and at rest, to protect it from unauthorized access. Use strong encryption protocols to ensure data security.

3. 4 Maintainability:

Maintainability is integral to **Cogniosis**, facilitating efficient updates and enhancements to meet evolving user needs and technological advancements. Maintainability requirements include:

- **Modular Architecture**: Adopt a modular code structure that allows for easy updates and feature enhancements. Modular design improves code readability and simplifies maintenance tasks.
- **Documentation**: Maintain comprehensive documentation for all aspects of the application, including code, APIs, and user guides. This documentation supports ongoing development and helps new developers quickly understand the system.

- **Feedback Loops**: Incorporate user feedback into the development process to prioritize and address issues promptly. Regularly update the app based on user needs and emerging trends in medical diagnostics.

3. 5 Modification:

Modification capabilities in **Cogniosis** empower developers to adapt and extend the app's functionalities to accommodate new requirements and emerging technologies. Modification requirements include:

- **Extensibility**: Design the app to be easily extensible, allowing for the addition of new features and functionalities without significant rework. Use design patterns that support scalability and flexibility.
- **Customizability**: Enable customization options for users, such as configurable settings and preferences. This allows users to tailor the app to their specific needs and workflows.
- **Continuous Improvement**: Foster a culture of continuous improvement, where new ideas and enhancements are regularly explored and implemented. Encourage experimentation and innovation within the development team.

By prioritizing modification capabilities, **Cogniosis** remains adaptable and scalable, ensuring long-term relevance and success.

3. 6 Portability:

Portability in **Cogniosis** ensures seamless accessibility and functionality across diverse devices and operating systems, enhancing user reach and engagement. Portability requirements include:

- **Responsive Design**: Implement responsive design principles to adapt the user interface to different screen sizes and resolutions. Ensure that the app provides a consistent user experience on smartphones, tablets, and other devices.
- **Offline Functionality**: Enable offline functionality for critical features, allowing users to access and analyze CT images without an internet connection. Ensure that data synchronization occurs seamlessly when connectivity is restored.

- **Resource Optimization**: Optimize the app for efficient resource usage, ensuring smooth performance on devices with varying hardware capabilities. Minimize memory usage and battery consumption to enhance user experience.

By prioritizing portability, **Cogniosis** maximizes user accessibility and engagement, making it a valuable tool for healthcare providers worldwide.

Chapter 4: System Analysis & Design Overview

4. 1 Use Case Diagrams:

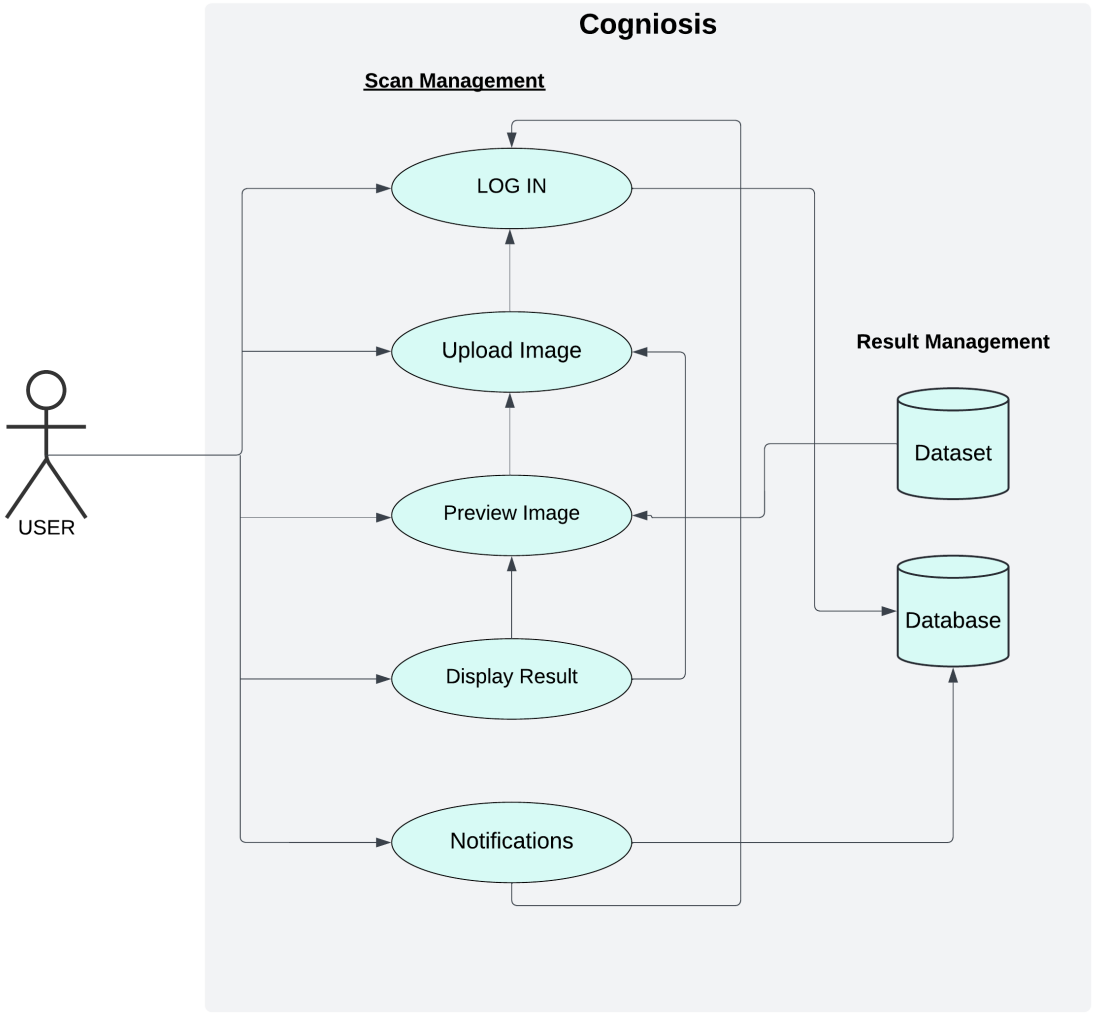

Figure 2 Use Case Diagram

Use case diagrams are essential for illustrating the interactions between users and the system. They provide a clear visual representation of the various functionalities and how users engage with the application. For **Cogniosis**, the primary actors include registered users (healthcare providers), guests, and administrators. Each use case represents a specific interaction or functionality within the app.

4. 2 Software Process Model:

Cogniosis follows an Agile software development methodology, promoting iterative development, continuous feedback, and flexibility to adapt to changing requirements. The Agile process model involves several sprints, each focusing on specific features and functionalities. Key phases in the Agile process for **Cogniosis** include:

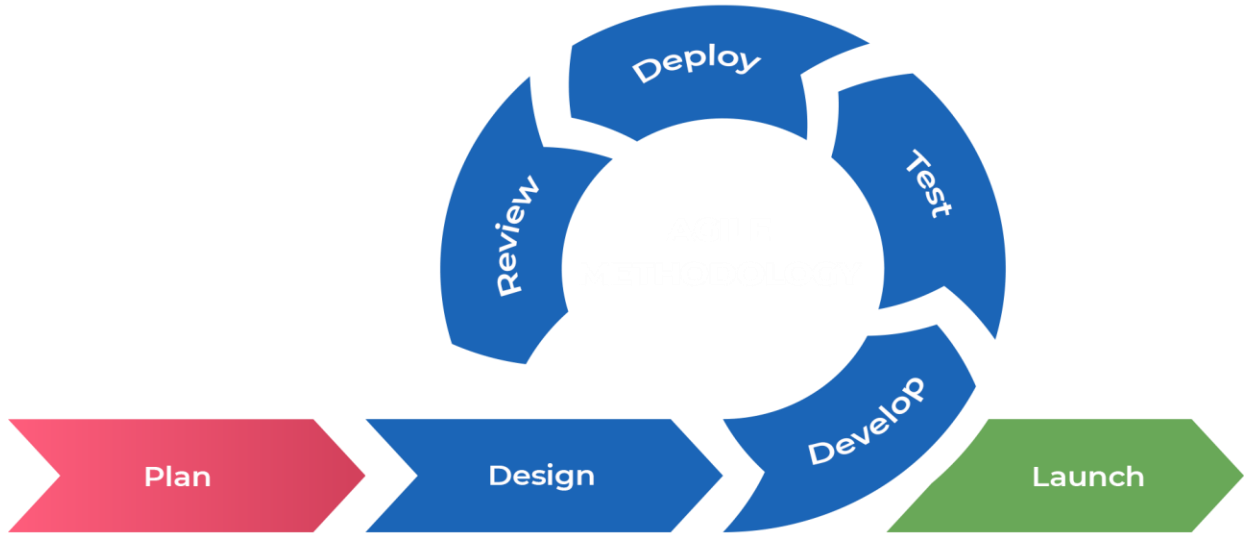

Figure 3 Agile Model

The Agile model is an iterative and incremental approach to software development that emphasizes flexibility, collaboration, and customer feedback. Instead of delivering a complete product at once, Agile breaks the project into smaller, manageable parts called sprints or iterations, each delivering a potentially shippable product increment. Teams work closely with stakeholders, continually refining and adjusting the product based on feedback and changing requirements.

4. 3 Data Model :

The data model for **Cogniosis** encompasses both the Entity-Relationship (ER) Model and the System Data Dictionary, providing a detailed blueprint of the data structure and relationships.

4.3.1 ER Model**:**

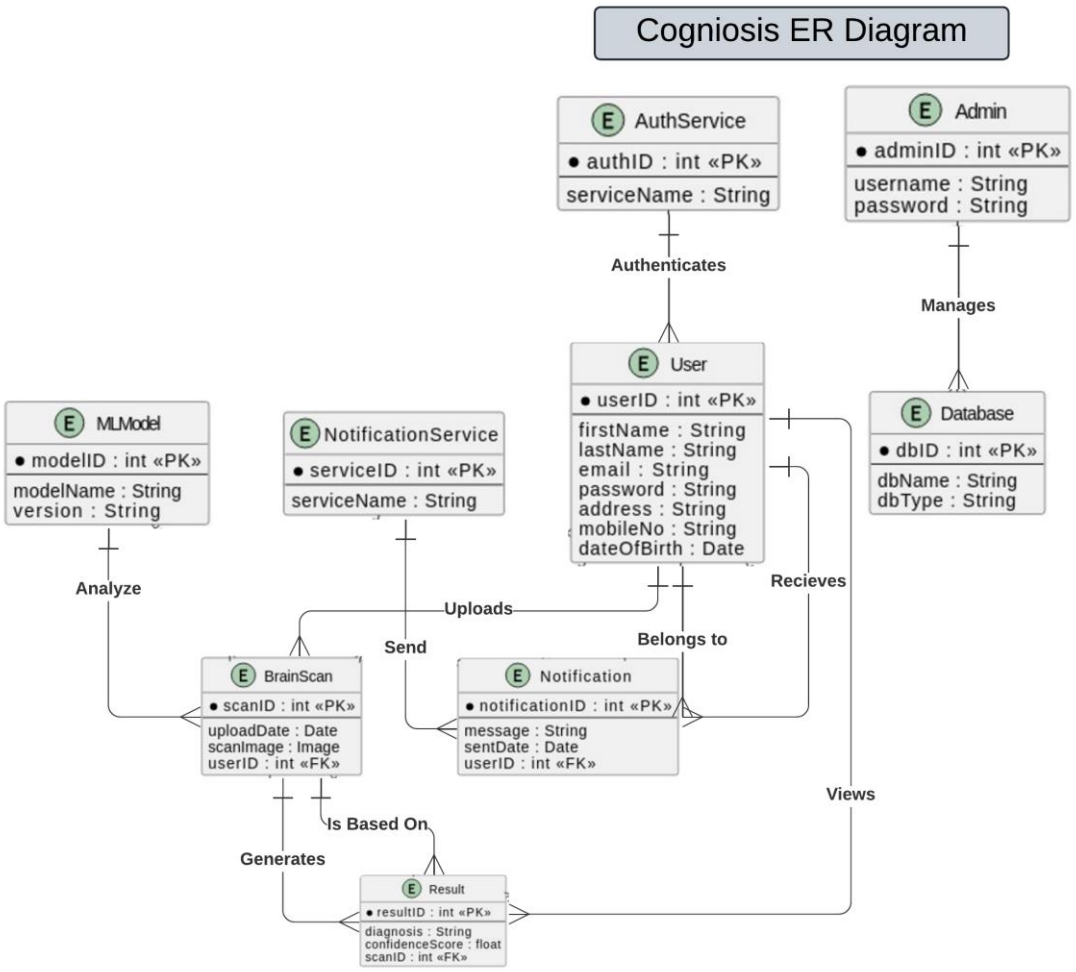

Figure 4 ER Diagram

An Entity-Relationship (ER) Diagram is a visual representation of a database's structure, showing how data is organized and related. It uses entities (represented by rectangles) to represent objects or concepts, and relationships (shown as diamonds or lines) to illustrate how entities are connected. ER Diagrams are essential for designing and understanding the logical structure of databases, helping to ensure data is stored and accessed efficiently.

4.3.2 System Data Dictionary:

The System Data Dictionary provides detailed descriptions of all data elements, ensuring consistent data handling and integrity. For instance:

4. 4 Behavioral Models:

Behavioral models describe the dynamic behavior of the system, capturing how it responds to various stimuli and interactions.

4.4.1 Data Flow Models:

Data Flow Models (DFMs) illustrate the flow of data within **Cogniosis**, identifying input, processing, storage, and output points.

4.4.1.1 Data Flow Diagram Level 0 :

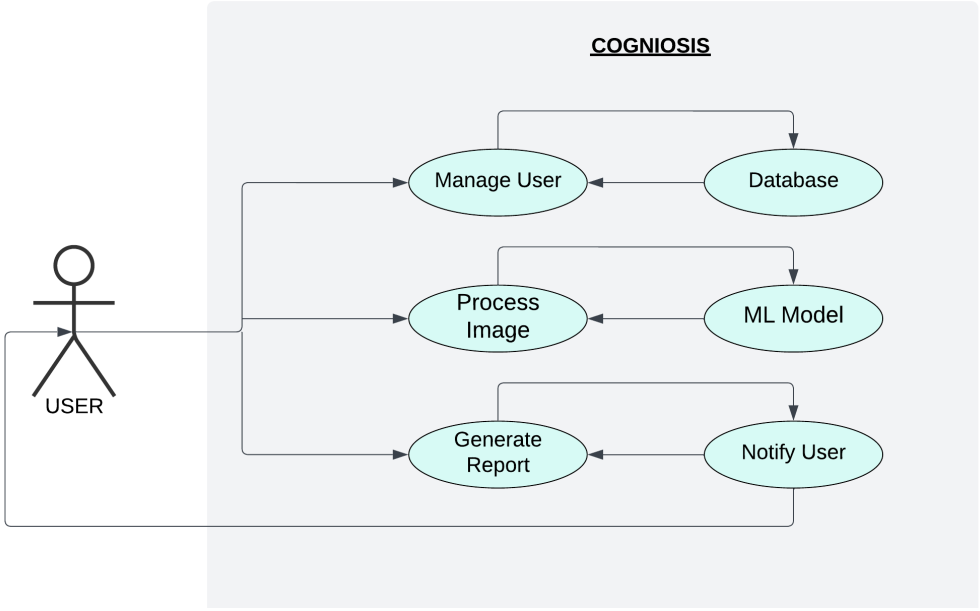

Figure 5 Data Flow Diagram Level 0

A Data Flow Diagram (DFD) Level 0, also known as a Context Diagram, provides a high-level overview of an entire system or process. It represents the system as a single process (usually with a circle or oval) and shows how it interacts with external entities, like users or other systems, through data flows.

4.4.1.2 Data Flow Diagram Level 1:

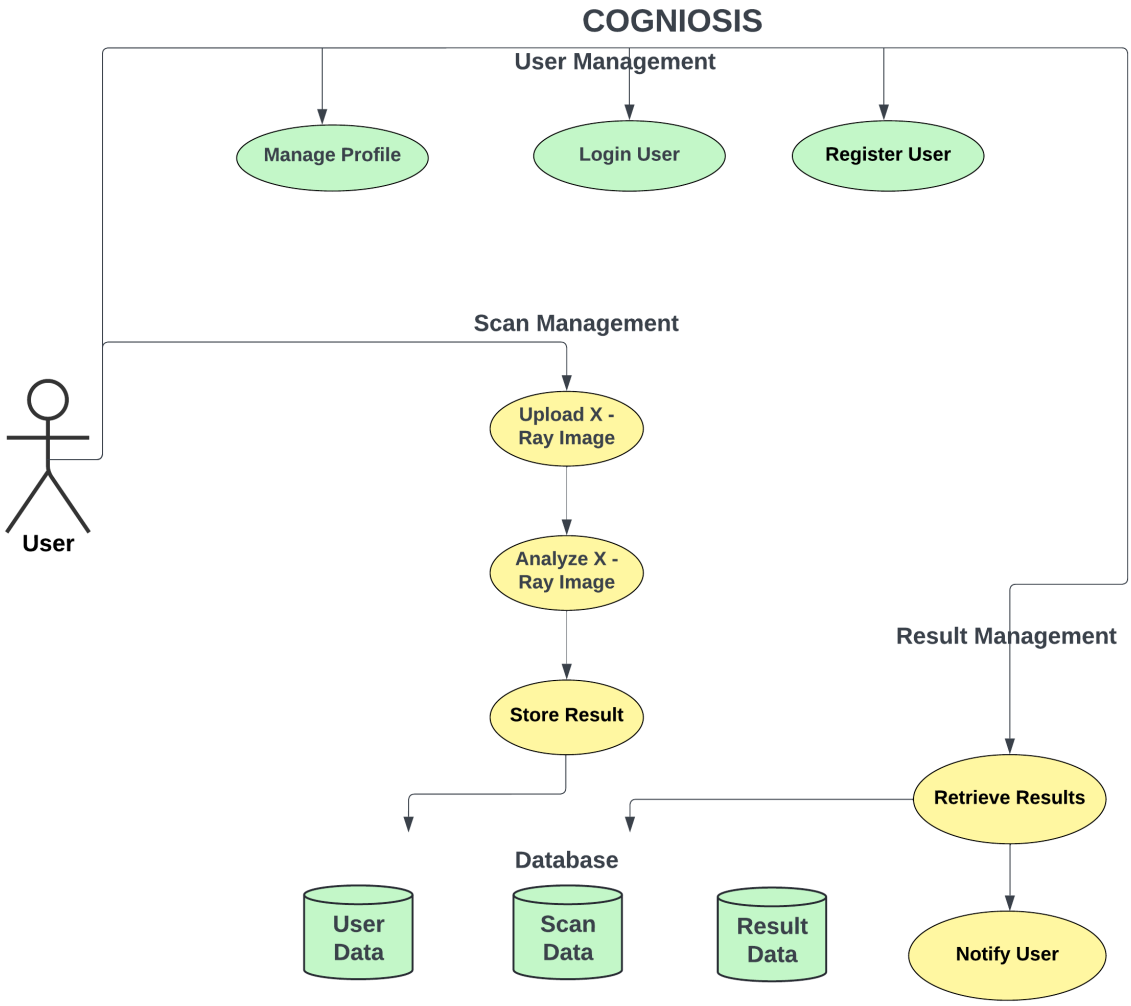

Figure 6 Data Flow Diagram Level 1

A Data Flow Diagram (DFD) Level 1 breaks down the high-level process from the Level 0 diagram into more detailed sub-processes. It provides a deeper look into the main process by splitting it into multiple sub-processes, each represented by a circle or oval. These sub-processes are interconnected by data flows, and they interact with external entities as well as each other. Level 1 DFD shows the internal structure of the system, highlighting how data moves between sub-processes and what data stores or external entities are involved. This level

provides more detailed insight into the system's functionality while still maintaining an overview.

4.4.1.3 Data Flow Diagram Level 2:

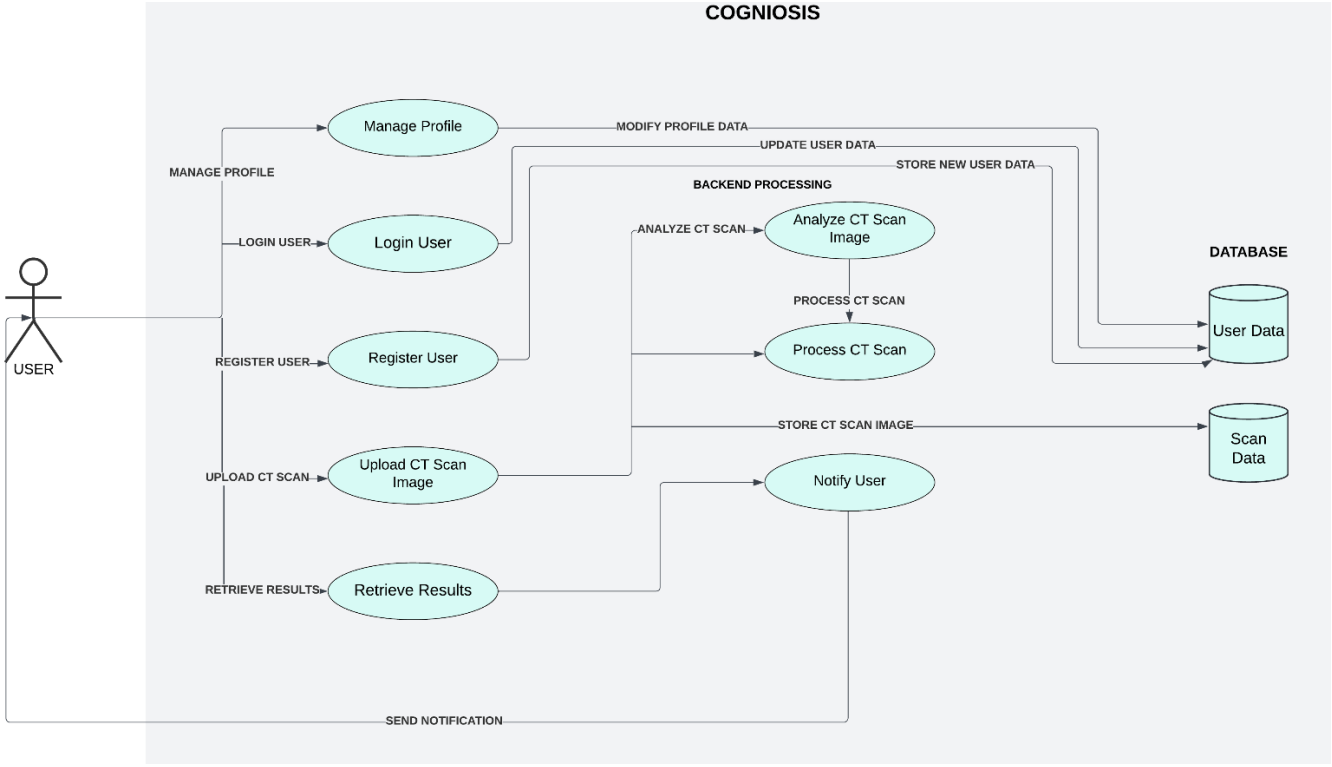

Figure 7 Data Flow Diagram Level 2

A Data Flow Diagram (DFD) Level 2 further decomposes the sub-processes found in a Level 1 DFD into more detailed and specific processes. This level of detail illustrates how each sub-process from Level 1 can be broken down into smaller, more granular processes, showing the exact flow of data between them. The Level 2 DFD captures specific data stores, data flows, and interactions between processes within the system. It is used to provide a comprehensive understanding of complex systems, where each process is broken down until it cannot be

simplified further. This level helps in identifying the smallest functional components of the system and their data interactions.

4.4.2 System Sequence Models:

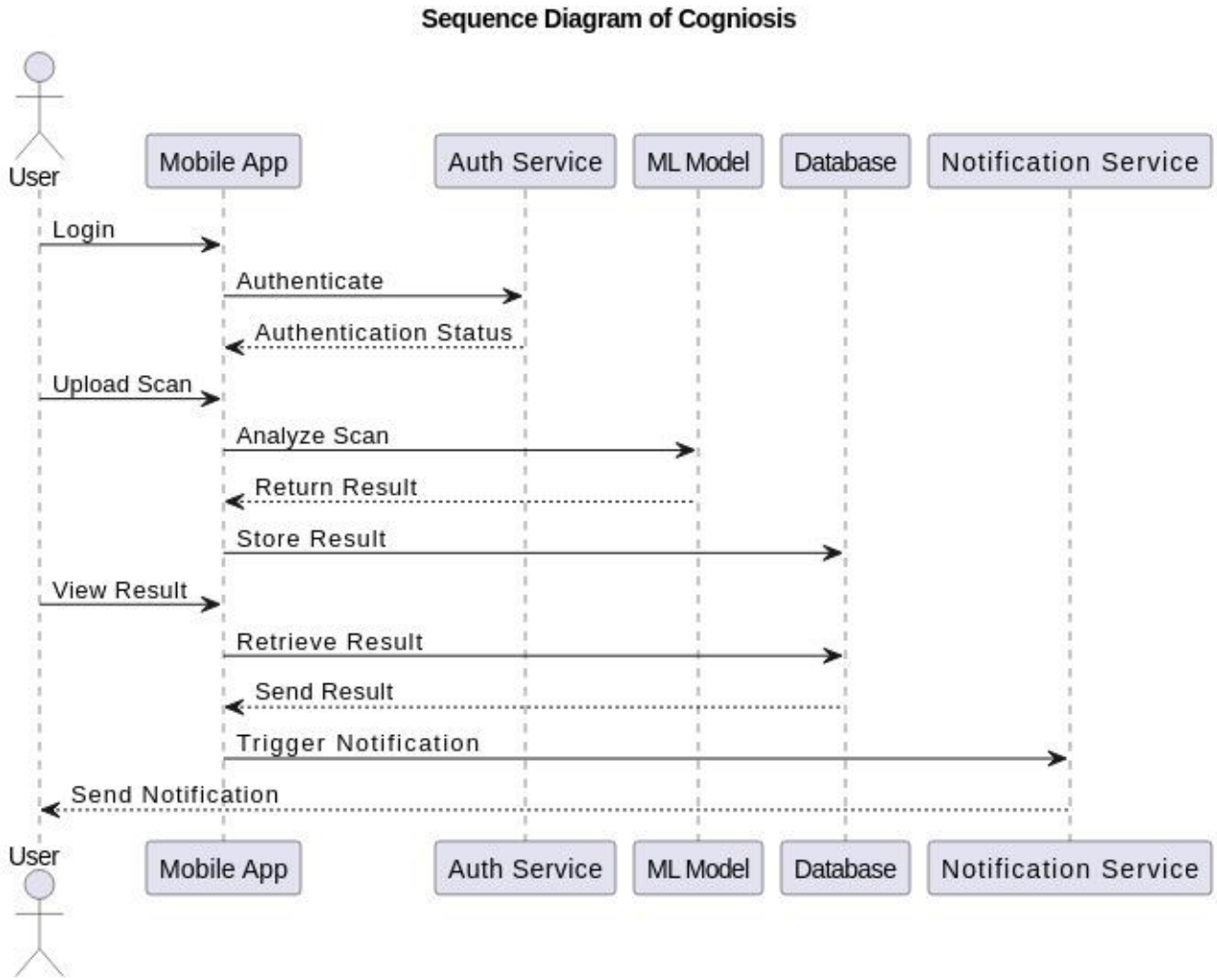

Figure 8 System Sequence Model

A Sequence Diagram is a UML tool that depicts how objects interact in a system over time. It shows objects as horizontal boxes and their lifelines as vertical dashed lines. Messages between objects are represented by horizontal arrows, indicating the flow and sequence of

communication. Sequence Diagrams help visualize the dynamic behavior of systems, making them useful for design and analysis.

4. 5 Object Models:

Object models define the structure and behavior of objects within the system, focusing on their attributes and methods.

4.5.1 State Models:

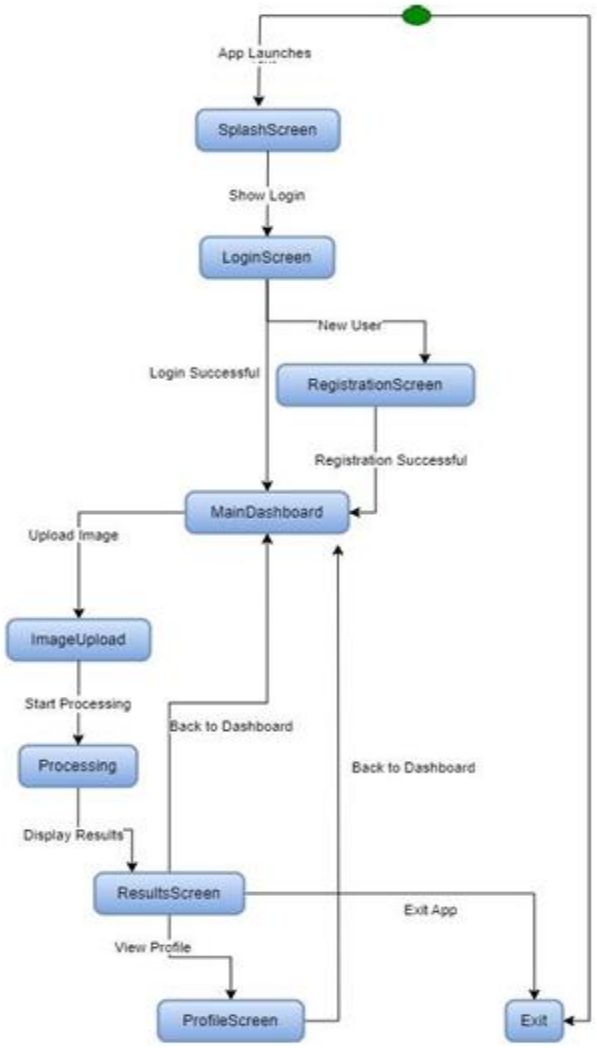

Figure 9 State Diagram

A state diagram, also known as a state machine diagram, visually represents the different states an object or system can be in and the transitions between those states. It shows how an object

responds to events or actions, moving from one state to another. This diagram is useful for modeling the behavior of systems and is commonly used in software engineering and systems design to describe how a system reacts to various inputs and changes over time.

4.5.2 Class Inheritance Model:

The Class Inheritance Model outlines the hierarchical relationships between classes, promoting reusability and modularity.

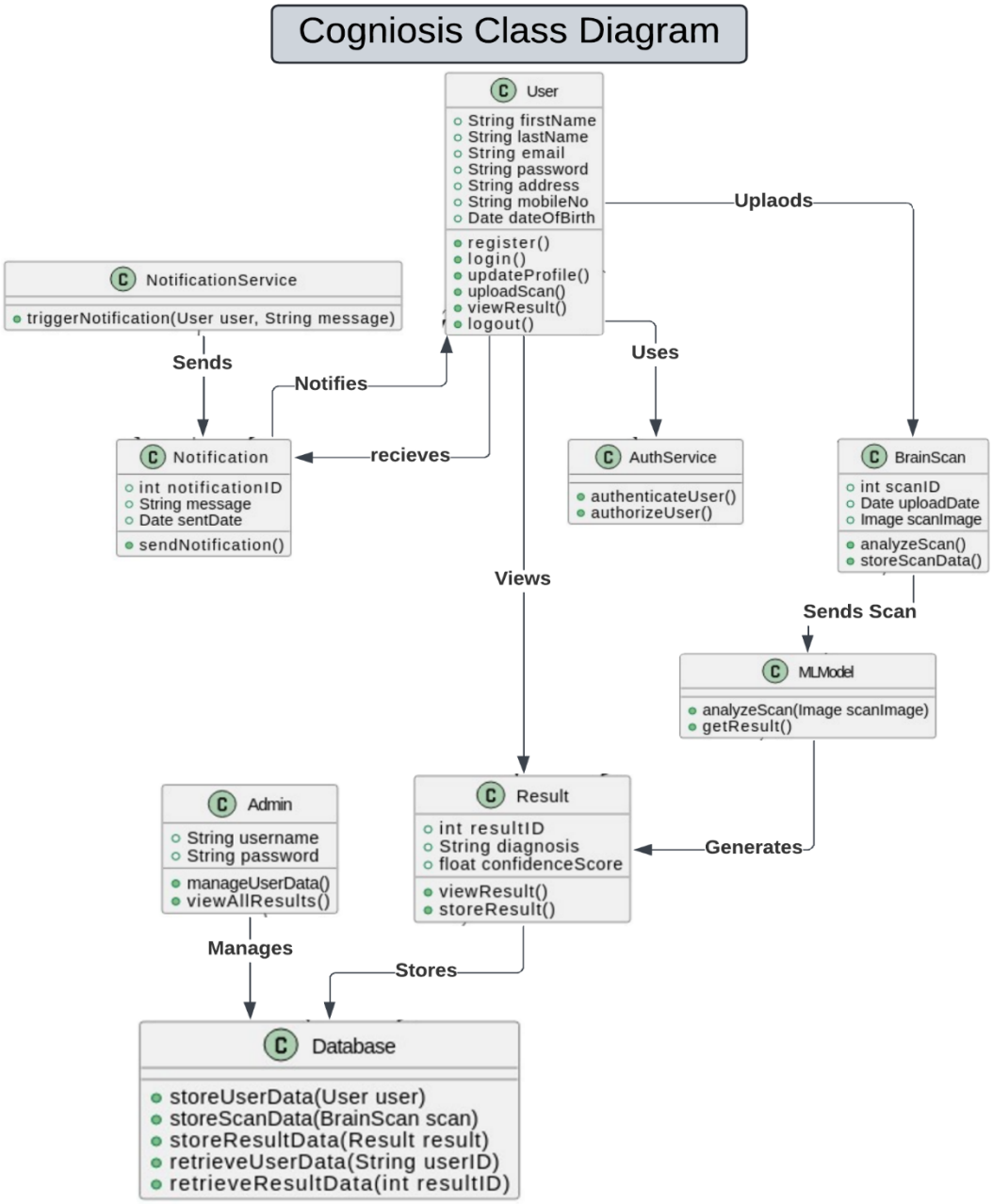

Figure 10 Class Diagram

4.6 Implementation Languages :

Cogniosis utilizes multiple programming languages to leverage the strengths of each for specific tasks.

Table 3 Language Implementation

Programming Language	Usage
Dart	Developing the mobile application using the **Flutter** framework.
Python	Training the ViT model and integrating it with TensorFlow Lite.

4.7 Required Support Software :

Cogniosis relies on several support software components to deliver a seamless and efficient user experience**.**

Table 4 Required Support Software

Support Software	Description
Firebase	Backend services for authentication, real-time database, and secure storage.
TensorFlow Lite	Deploys the ViT model on mobile for efficient on-device machine learning.
Kaggle	Provides the dataset of brain stroke CT images for model training.

4.8 System Model:

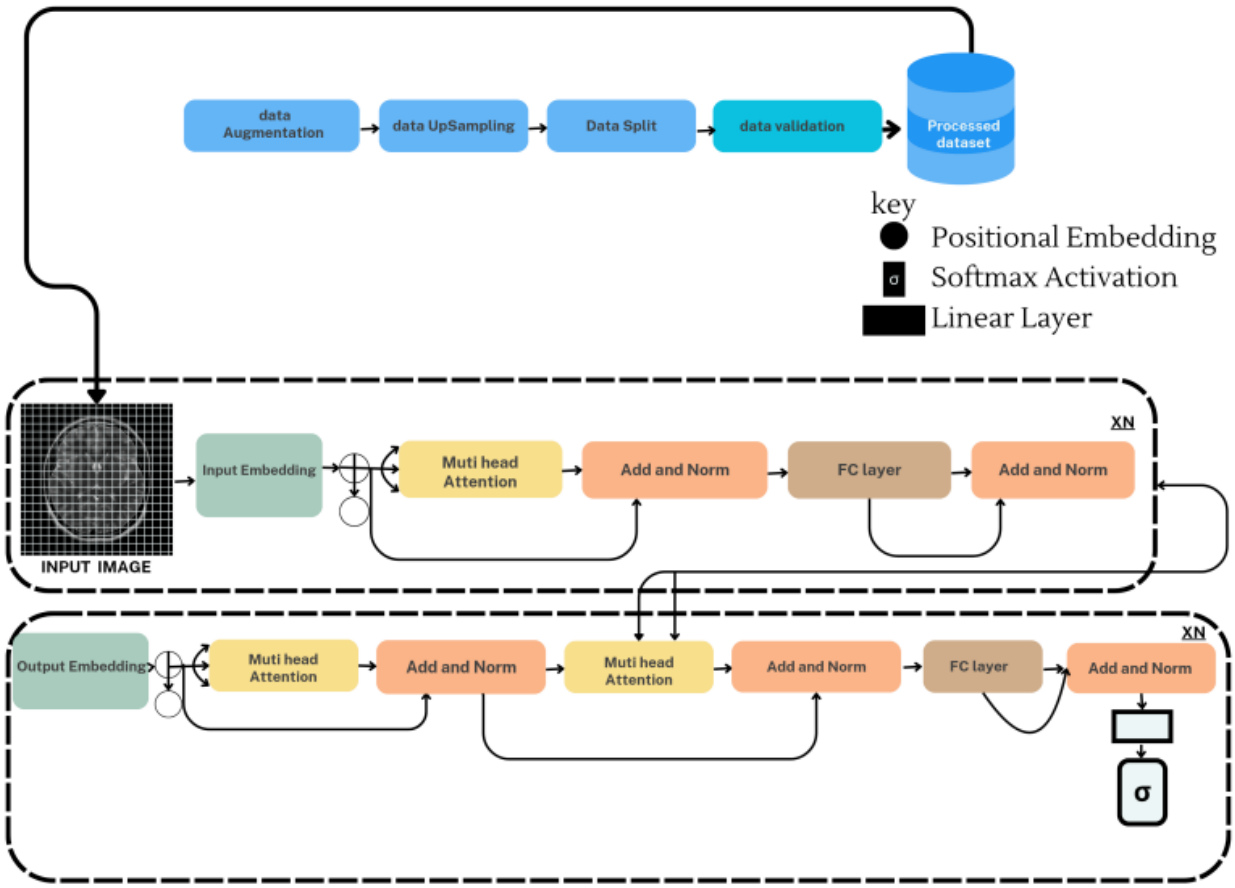

Figure 11 System Model Diagram

Chapter 5: User Interface Design

5.1 User Interface Specification:

5.1.1 User Interface Designs:

5.1.1.1 Splash Screen:

The introductory screen that displays the app's logo and branding.

Figure 12: Splash Screen

A splash screen is a brief introductory screen that appears when an app or website is launched. It typically displays a logo, animation, or branding while the application or site is loading. The

splash screen provides a visually engaging experience and can enhance the initial impression of the app.

5.1.1.2 Sign up Screen:

Enables new users to create an account with the app.

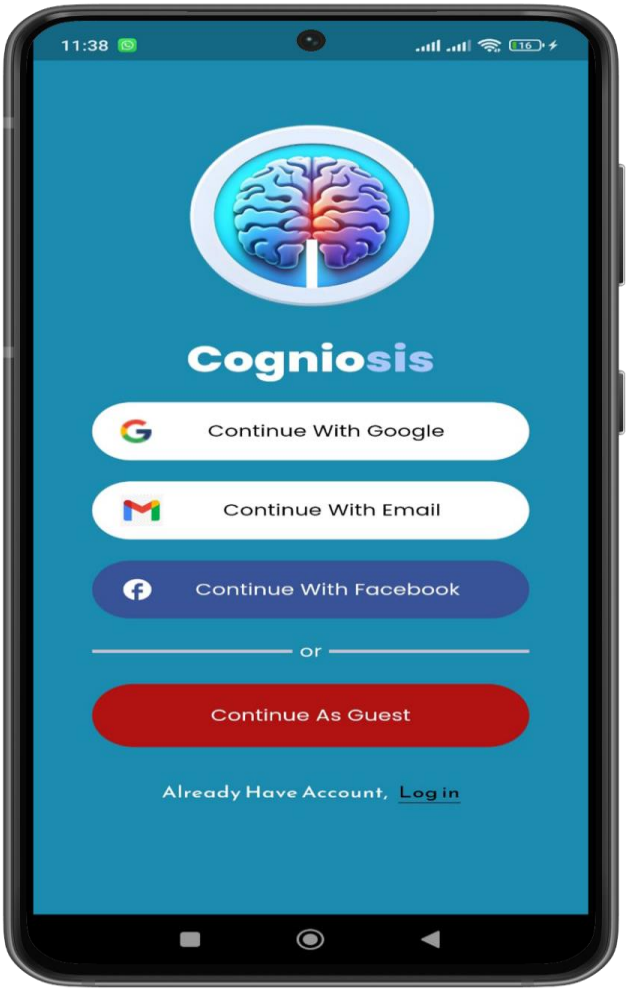

Figure 13: Sign Up Screen

A sign-up screen allows new users to create an account by providing necessary information like name, email, and password. It often includes fields for entering these details, a sign-up button,

and may offer additional options such as agreeing to terms of service. Some screens might also provide social media for quicker registration.

5.1.1.3 Registration Screen:

Allows users to add their information into their accounts.

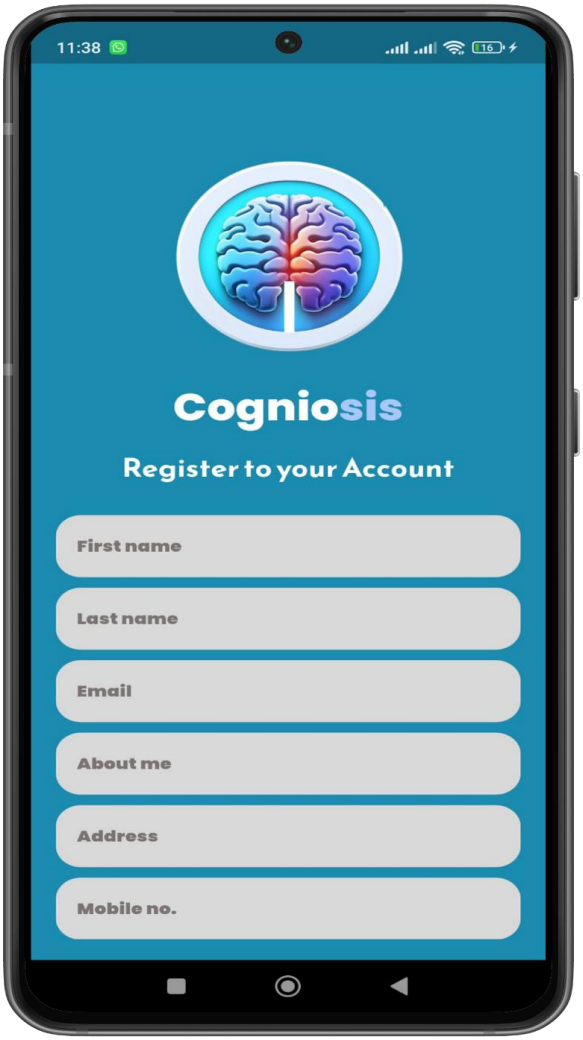

Figure 14: Registration Screen

A registration screen allows new users to create an account by entering their personal details, such as name, email, and password. It typically includes fields for required information, validation checks, and a submit button. The screen may also offer options for account creation through social media or other services.

5.1.1.4 Homepage Screen:

The main dashboard where users can navigate to various features.

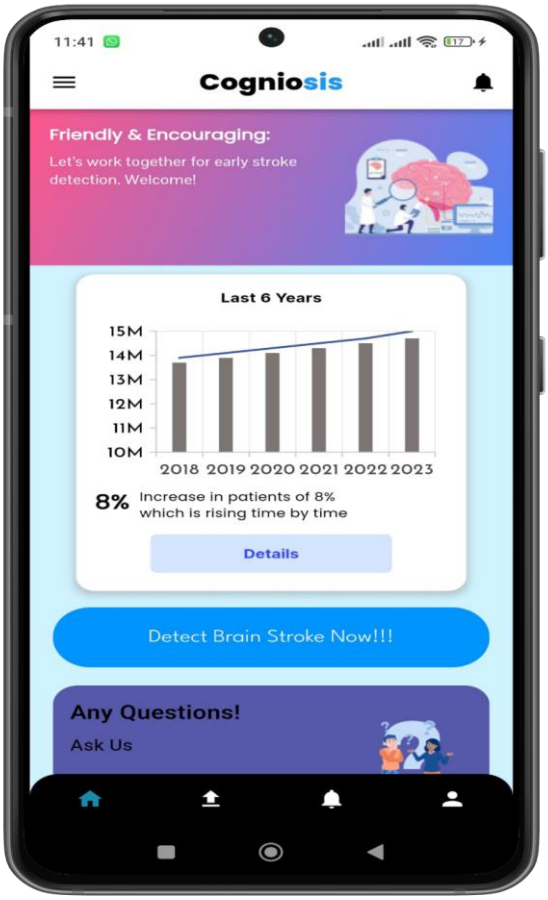

Figure 15: Homepage Screen

The homepage screen is the main landing page of a website or app, providing an overview and access to key features or sections. It often includes a navigation menu, highlights of important content or updates, and links to various functionalities. The homepage serves as the starting point for user interaction and exploration.

5.1.1.5 Image Upload and Capture Screen:

Interface for users to upload X-ray images.

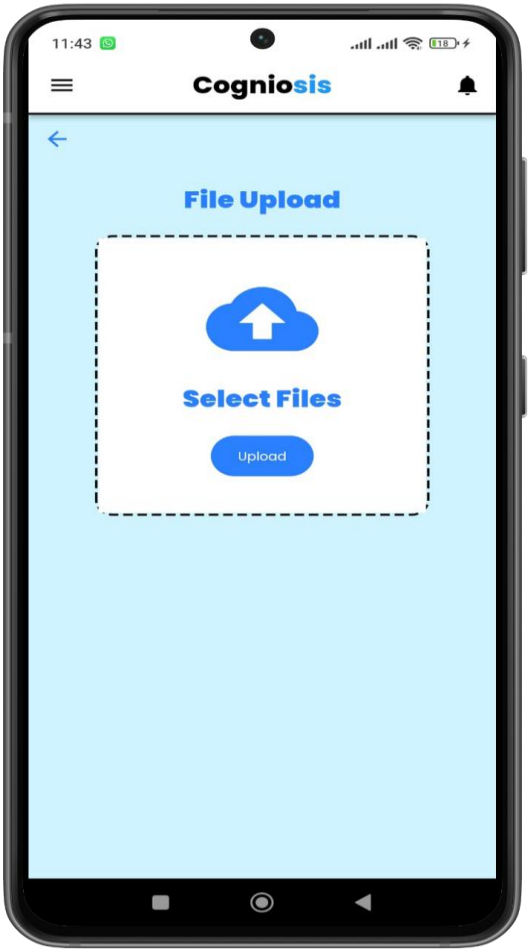

Figure 16: Image Uplaod and Capture Screen

An image upload and capture screen allows users to select and upload images from their device. It usually features an interface for choosing files, a preview of the selected image, and an upload button. This screen may also display progress indicators and provide feedback on successful or failed uploads.

5.1.1.6 Preview CT Scan Screen:

Displays the uploaded X-ray image for review before processing.

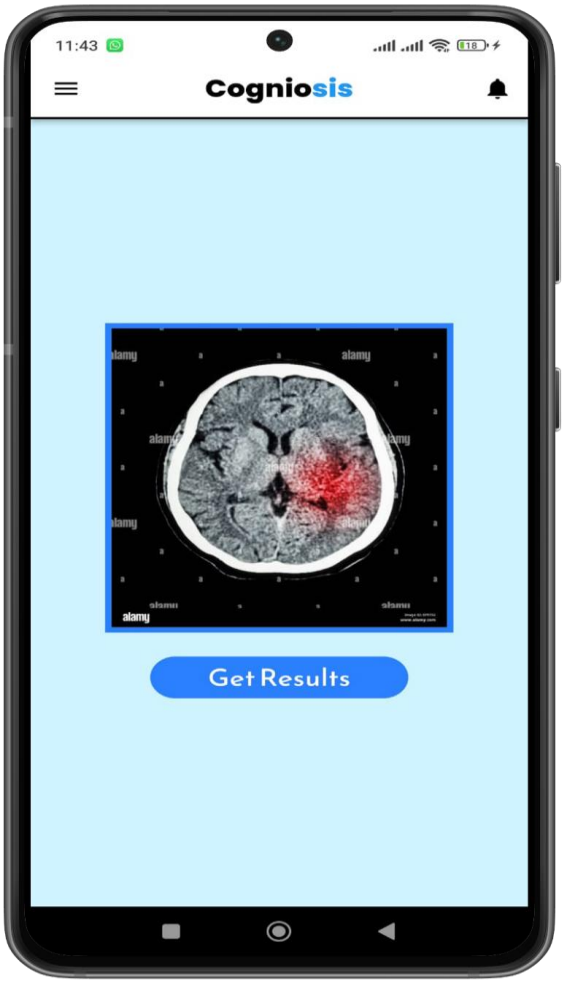

Figure 17: Preview CT Scan Screen

A "Preview CT Scan" screen allows users to view and examine the CT scan images they have uploaded. It typically includes tools for zooming, panning, and navigating through different

slices of the scan. This screen helps users review and ensure the quality of their medical imaging before further processing or analysis.

5.1.1.7 Results Screen:

Shows the outcome of the disease detection analysis.

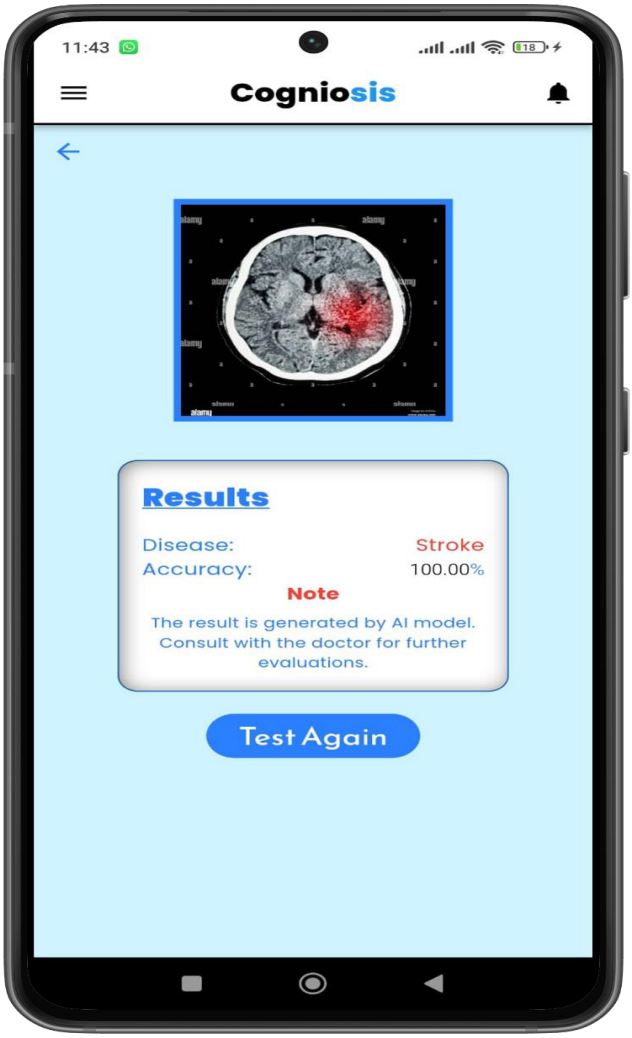

Figure 18: Results Screen

A results screen displays outcomes or data based on user input or actions, such as search results, test scores, or performance metrics. It presents relevant information in a clear format, often with options to filter or sort the results for better user understanding and interaction.

5.1.1.8 Profile Management Screen:

Allows users to view their profile information.

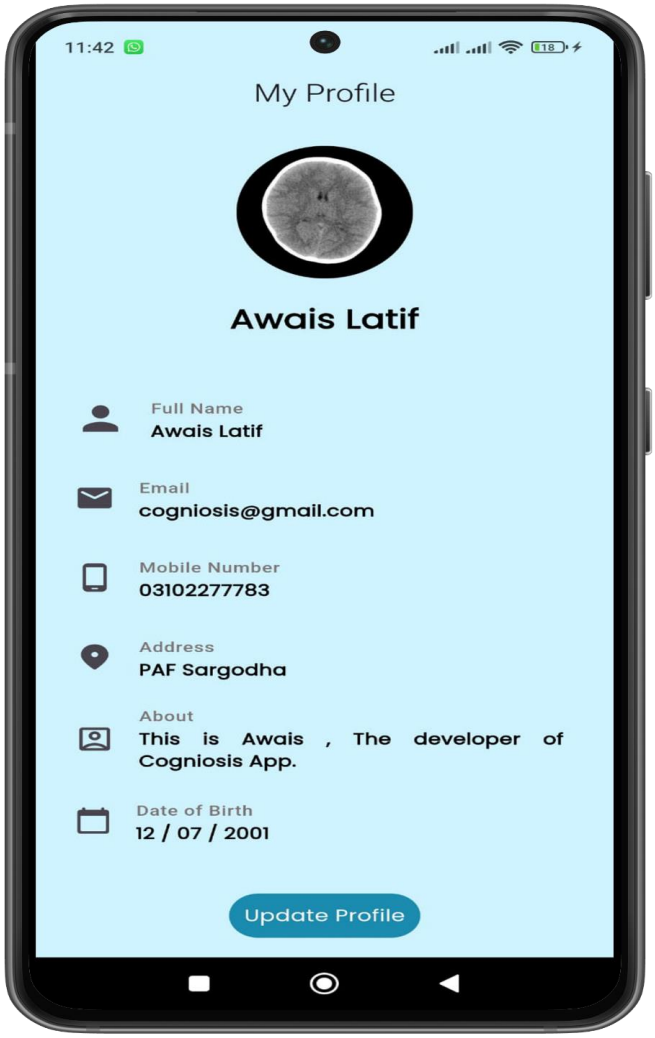

Figure 19: Profile Management Screen

A profile management screen allows users to view and edit their personal information, such as name, email, profile picture, and password. It often includes options for updating contact details, managing preferences, and viewing account activity. This screen helps users maintain and customize their account settings.

5.1.1.9 About Us Screen:

Provides information about the app and its developers.

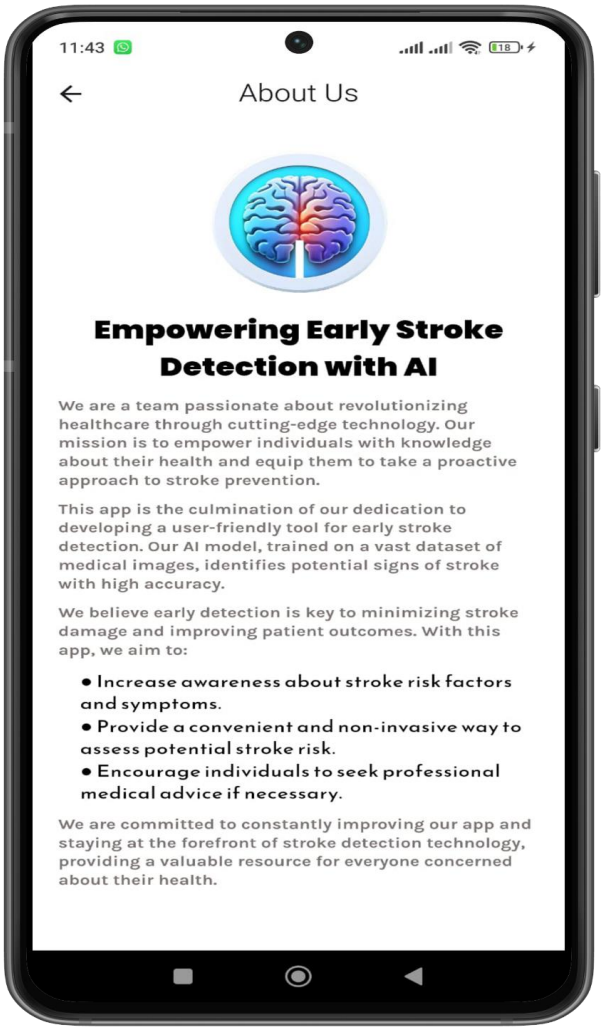

Figure 20: About Us Screen

An "About Us" screen offers information about the organization or team behind the app or website. It typically includes details like the company's mission, history, team members, and

values. This screen helps users understand who is behind the product or service and can build trust and credibility.

5.1.1.10 FAQs Screen:

Displays frequently asked questions and their answers.

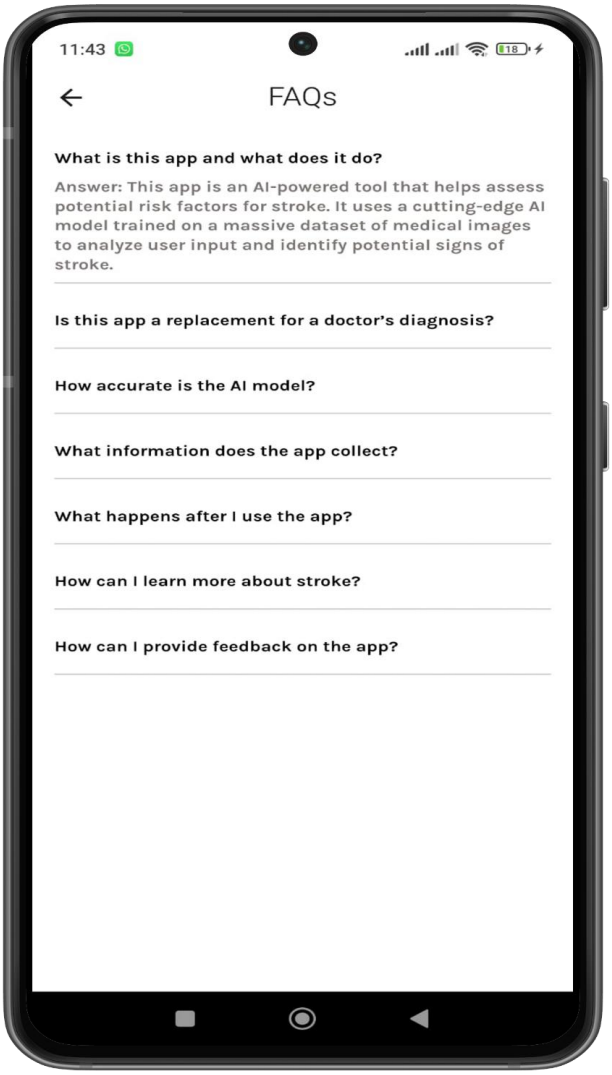

Figure 21: FAQs Screen

An FAQ screen provides users with a list of frequently asked questions and their answers. It is designed to help users find quick solutions to common issues or queries without needing to contact support. The screen typically includes a search function or categories to help users easily navigate and find relevant information.

5.1.1.11 Contact Us Screen:

Enables users to contact support or ask questions.

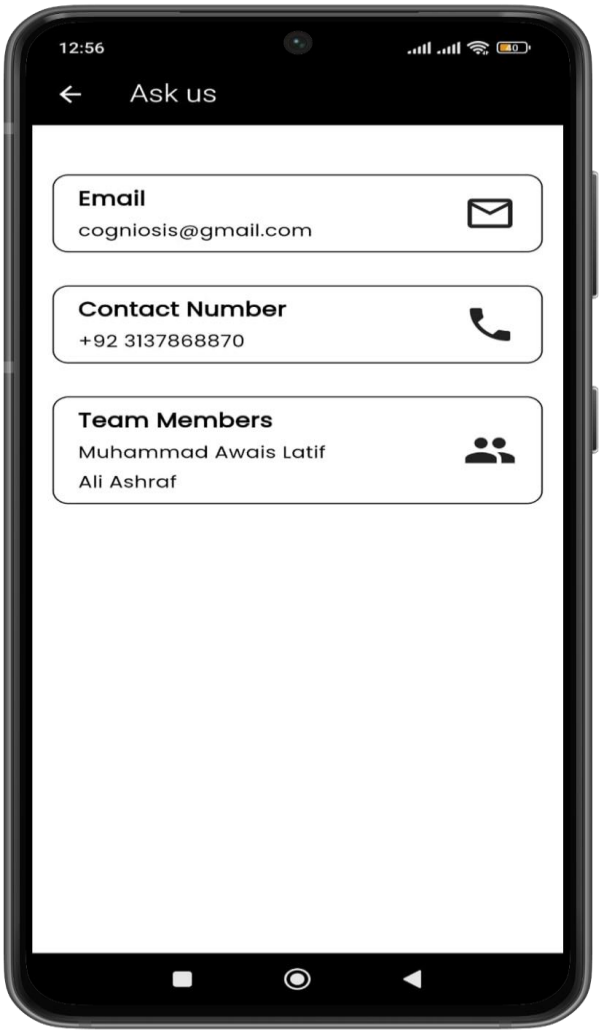

Figure 22: Contact Us Screen

A "Contact Us" page is a section of a website or app that provides users with ways to get in touch with the organization or support team. It usually includes contact forms, email addresses, phone numbers, and sometimes a physical address or map. This page facilitates communication and helps users seek assistance or provide feedback.

5.1.1.12 Header Menu:

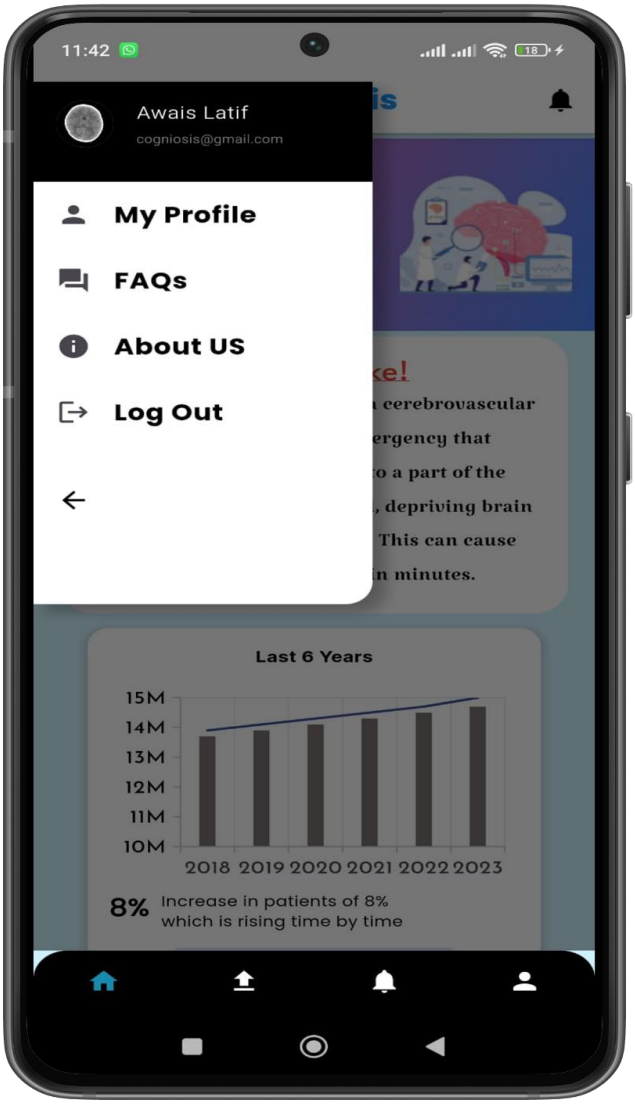

Figure 23: Header Menu

A header menu typically refers to a navigation element at the top of the screen that provides access to key features or sections of the app. It often includes icons or text for actions like settings, search, or user profile. This menu enhances user navigation by offering quick access to important functions.

Chapter 6: System Implementation

6.1 Items/Functions to be Tested:

Table 5 Items/Functions to be Tested

Item/Function	Description
User Authentication	Sign-up, login, and authentication processes
Image Upload and Capture	Uploading images from device or capturing new images
Image Analysis	Processing and analyzing CT images with the ML model
Results Display	Displaying diagnostic results with confidence scores
User Profile Management	Viewing and updating user profile information
Navigation	Navigating between different screens and sections
Performance	Application performance under various conditions
Security	Ensuring secure data transmission and storage

6.2 Description of Test Cases:

Table 6 Description of Test Cases

Test Case ID	Test Case Description	Expected Result
TC-001	Test user sign-up functionality	User successfully signs up with valid details
TC-002	Test user login functionality	User successfully logs in with valid credentials
TC-003	Test image upload from device	Image is successfully uploaded and displayed
TC-004	Test image capture using camera	Image is successfully captured and displayed
TC-005	Test CT image analysis	Image is analyzed and results are displayed
TC-006	Test results display with confidence scores	Results are displayed with correct confidence scores
TC-007	Test user profile update	User profile is updated successfully
TC-008	Test navigation between screens	Smooth navigation without errors
TC-009	Test application performance under load	Application remains responsive under heavy load
TC-010	Test secure data transmission	Data is transmitted securely using HTTPS
TC-011	Test secure data storage	Data is stored securely in Firebase Firestore

6.3 Justification of Test Cases:

Table 7 Justification of Test Cases

Test Case ID	Justification
TC-001	Essential to verify that users can sign up to the app without issues
TC-002	Ensures that users can log in and access their accounts
TC-003	Validates the core functionality of uploading CT images
TC-004	Ensures the app can capture images using the device camera, which is crucial for user experience
TC-005	Confirms the ML model is correctly analyzing CT images and providing results
TC-006	Verifies that diagnostic results are accurately displayed to the user
TC-007	Ensures users can manage and update their profile information
TC-008	Checks that the app navigation is intuitive and error-free
TC-009	Validates the app's performance and responsiveness under various conditions
TC-010	Ensures that data transmission is secure, protecting user information
TC-011	Confirms that user data is securely stored, meeting privacy and security requirements

6.4 Test Run Procedures and Results:

Table 8 Test Run Procedures and Results

Test Case ID	Procedure	Result	Status
TC-001	1. Open the app 2. Navigate to sign-up screen 3. Enter valid details 4. Submit form	User signs up successfully	Pass
TC-002	1. Open the app 2. Navigate to login screen 3. Enter valid credentials 4. Submit form	User logs in successfully	Pass
TC-003	1. Navigate to image upload screen 2. Select an image from device 3. Upload image	Image is uploaded and displayed	Pass
TC-004	1. Navigate to image capture screen 2. Capture an image using the camera 3. Save and display image	Image is captured and displayed	Pass
TC-005	1. Upload or capture an image 2. Initiate image analysis 3. View results	Results are displayed correctly	Pass
TC-006	1. Analyze an image 2. View results with confidence scores	Results and confidence scores displayed correctly	Pass
TC-007	1. Navigate to profile screen 2. Update profile details 3. Save changes	Profile is updated successfully	Pass
TC-008	1. Navigate through different screens and sections	Smooth navigation without errors	Pass
TC-009	1. Perform various tasks under simulated heavy load	App remains responsive	Pass
TC-010	1. Monitor data transmission using network tools	Data transmitted securely	Pass
TC-011	1. Check data storage practices in Firebase Firestore	Data stored securely	Pass

6.5 Regression Testing:

Table 9 Regression Testing

Test Case ID	Previous Functionality	Test Scenario	Expected Outcome	Actual Outcome	Pass/ Fail
RTC01	User Registration	Register new user	User registered	User registered	Pass
RTC02	Password Recovery	Recover password	Password reset link sent	Password reset link sent	Pass
RTC03	Data Export	Export data to CSV	Data exported	Data exported	Pass

Chapter 7: Future Enhancements and Expansion

7.1 Introduction:

Cogniosis is designed with a vision for continuous improvement and scalability, ensuring that it remains a cutting-edge tool for brain disease diagnostics. This chapter outlines the planned enhancements and expansions that will be implemented in the future to further extend the app's functionality, improve user experience, and increase its accessibility. These future enhancements include the addition of more brain diseases, deployment on iOS and web platforms, and the implementation of continuous integration/continuous deployment (CI/CD) pipelines.

7.2 Support for Additional Brain Diseases

Expanding the diagnostic capabilities of **Cogniosis** to include more brain diseases is a critical enhancement that will make the app a comprehensive tool for neurological diagnostics. Future plans include:

I. **Inclusion of More Diseases**:

- **Alzheimer's Disease**: Integrating diagnostic models to detect early signs of Alzheimer's disease using brain imaging data.
- **Parkinson's Disease**: Developing capabilities to identify Parkinson's disease through characteristic patterns in brain scans.
- **Brain Tumors**: Incorporating detection models for various types of brain tumors, facilitating early diagnosis and treatment planning.
- **Multiple Sclerosis (MS)**: Adding support for identifying MS through advanced imaging analysis techniques.

II. **Data Collection and Model Training**:

- **Dataset Expansion**: Acquiring and curating comprehensive datasets for each new brain disease from reliable sources like Kaggle and medical institutions.
- **Advanced Model Training**: Utilizing state-of-the-art machine learning techniques, such as convolutional neural networks (CNNs) and ViTs, to train highly accurate diagnostic models.

III. **User Interface Adaptation**:

- **Enhanced UI Elements**: Updating the user interface to include new diagnostic options while maintaining a user-friendly experience.
- **Disease Selection**: Providing users with a straightforward method to select the specific disease they wish to diagnose.
-

IV. **Educational Content**:

- **Informational Updates**: Expanding educational resources to include comprehensive information on newly supported diseases.
- **Multimedia Content**: Adding videos, infographics, and interactive guides to improve user understanding and engagement.

7.3 Deployment on iOS and Web Platforms:

To maximize the reach and accessibility of **Cogniosis**, future plans include deploying the application on iOS and web platforms. This expansion will ensure that users can access the app on a variety of devices and operating systems.

I. **Unified User Experience**:

- **Consistent Interface**: Maintaining a consistent and intuitive user interface across all platforms to ensure a seamless user experience.
- **Feature Parity**: Ensuring that all functionalities available on the mobile app are also accessible on the iOS and web versions.

7.4 Continuous Integration and Continuous Deployment (CI/CD)

To streamline the development process and ensure rapid delivery of updates and new features, **Cogniosis** will implement CI/CD pipelines. This approach will automate the build, testing, and deployment processes, ensuring that new features and bug fixes are delivered to users efficiently.

I. **CI/CD Pipelines**:

- **Automated Builds**: Setting up CI/CD pipelines to automatically build the application whenever new code is committed to the repository.
- **Automated Testing**: Integrating automated testing to ensure that new code changes do not introduce bugs or regressions.
- **Continuous Deployment**: Automating the deployment process to deliver new features and updates to users quickly and reliably.

II. **Benefits of CI/CD**:

- **Faster Release Cycles**: Enabling rapid iteration and frequent releases, ensuring that users benefit from the latest improvements and features.
- **Improved Quality**: Ensuring that automated testing catches issues early in the development process, leading to a more stable and reliable application.
- **Developer Efficiency**: Reducing the manual effort required for builds and deployments, allowing developers to focus on feature development and innovation.

7.5 User Feedback and Iterative Improvements

Incorporating user feedback is crucial for the continuous improvement of **Cogniosis**. Future plans include robust mechanisms for collecting and acting on user feedback to refine and enhance the app.

I. **Feedback Mechanisms**:

- **In-App Surveys**: Implementing surveys and feedback forms within the app to gather user opinions and suggestions.
- **Beta Testing Programs**: Engaging users in beta testing programs to test new features and provide feedback before official releases.

II. **Community Engagement**:

- **User Forums and Support Groups**: Establishing online forums and support groups where users can share experiences, ask questions, and provide feedback.
- **Social Media Integration**: Leveraging social media platforms to engage with users, announce updates, and gather feedback.

III. **Data-Driven Decisions**:

- **Analytics and Metrics**: Utilizing analytics to monitor user behavior and identify areas for improvement.
- **Iterative Development**: Adopting an iterative development approach, where user feedback is continuously incorporated into development cycles to refine and enhance the app.

7.6 Strategic Partnerships and Collaborations

Forming strategic partnerships and collaborations with medical institutions, research organizations, and technology companies will be essential for the future growth and success of **Cogniosis**. Key aspects include:

I. **Medical Institutions**:

- **Clinical Trials and Studies**: Partnering with hospitals and clinics to conduct clinical trials and validate the app's diagnostic capabilities.
- **Data Sharing Agreements**: Establishing data-sharing agreements to access diverse and comprehensive datasets for model training and validation.

II. **Research Organizations**:

- **Joint Research Initiatives**: Collaborating on research initiatives to explore new diagnostic techniques and improve existing models.
- **Funding and Grants**: Seeking funding and grants to support research and development efforts.

III. **Technology Companies**:

- **Technology Integration**: Collaborating with technology companies to integrate advanced tools and platforms, such as cloud services, AI frameworks, and AR/VR technologies.

Innovation and Development: Participating in innovation programs and hackathons to drive continuous improvement and explore new ideas.