



Cairo University
Faculty of Computers and Artificial Intelligence
Information Systems Department



Graduation Project Report

AlzCare

By

Name	ID
Aya Mohamed Abdelrahman	20211020
Karin Kheir Abdallah Mataa	20210292
Shimaa Amer Mohamed	20211056
Mahmoud Sayed Abdel'aty	20210370
Mahmoud Ali Abdelnnam	20210371
Mario Malak Alabd	20210313

Under the Supervisors of:

Dr. Shimaa Galal

TA. Aliaa Magdy

Department of Information Systems, Faculty of Computers and Artificial Intelligence, Cairo University, Egypt

2025

Acknowledgements

We are immensely thankful to **Dr. Shima M. Galal** for her guidance, mentorship, and unwavering support. Her expertise and dedication have been instrumental in shaping the direction of our graduation project.

We extend our heartfelt thanks to **TA. Aliaa Magdy** for her insightful feedback and encouragement. Her wisdom and counsel have been invaluable in enhancing the quality of the work.

Abstract

Alzheimer's disease is a progressive neurodegenerative disorder that significantly impairs memory and cognitive function, making it difficult for patients to manage daily activities and recognize familiar faces. As the disease advances, caregivers face immense challenges in ensuring patient safety, monitoring medication adherence, and providing real-time support. The purpose of the AlzCare project is to develop a comprehensive, patient-centered application that bridges the gap between Alzheimer's patients and their caregivers, offering digital tools to enhance safety, independence, and communication.

The AlzCare app focuses on intermediate-level Alzheimer's patients, utilizing a combination of geolocation-based features, real-time notifications, and cognitive support systems to address the key challenges in caregiving. The app includes features such as medication reminders, geofencing for safe zone alerts, and face recognition to help patients identify loved ones. Caregivers can track patient locations in real time, receive alerts when patients leave designated safe areas, and get notified if patients miss medications or trigger emergency actions. Additionally, the app syncs with smartwatches for health monitoring and offers cognitive training activities to slow memory decline.

The development of AlzCare follows the Waterfall methodology, ensuring structured, sequential phases from requirement gathering to deployment. Through iterative testing and user feedback, the application has been refined to meet both the functional needs of patients and the administrative needs of caregivers and medical professionals. The significance of this research lies in its ability to empower Alzheimer's patients by promoting independence while providing caregivers with tools for real-time support. This digital solution not only enhances patient care but also reduces the burden on caregivers, allowing for a more effective, connected caregiving experience.

By integrating features such as emergency alerts, face recognition, and safe zone monitoring, the AlzCare project presents a novel approach to addressing the daily complexities faced by Alzheimer's patients and their families. This technology offers a valuable contribution to the field of healthcare by improving patient safety, communication, and overall quality of life.

Table of Contents

CHAPTER 1 INTRODUCTION	1
1.1 Background and Motivation	1
1.2 Problem Statement	1
1.3 Objectives	2
1.4 Project Scope and Limitations	2
1.5 Project Methodology.....	4
1.6 Project Report Outline.....	6
CHAPTER 2 MARKET AND LITERATURE SURVEY	7
2.1 Market comparison	7
2.2 Feature comparison.....	9
2.3 SWOT analysis	10
CHAPTER 3 ALZCARE ANALYSIS	12
3.1 Functional requirements	12
3.1.1 Patient Functional Requirements.....	12
3.1.2 Caregiver Functional Requirements	13
3.1.3 Admin Functional Requirements.....	14
3.2 Non-functional requirements.....	15
3.2.1 Usability:.....	15
3.2.2 Performance:.....	15
3.2.3 Availability:	15
3.2.4 Scalability:.....	15
3.2.5 Security:.....	15
3.2.6 Reliability:	16
3.2.7 Maintainability:.....	16
3.3 Stakeholders and their responsibilities	16
3.3.1 End-Users:.....	16
3.3.2 System Administrators	17
3.4 Task profile	18
3.5 User profile	19
3.6 Use Case Diagram.....	20
3.7 Use Case Scenario.....	21

3.7.1	Press emergency button.....	21
3.7.2	Manage patient memories	22
3.7.3	Manage safe zones	23
3.7.4	Safe zone breach	24
3.7.5	Manage appointment.....	25
3.7.6	Manage medical history	26
3.7.7	Initial assessment	27
3.8	Personas	28
3.9	Image processing model description.....	28
CHAPTER 4	ALZCARE DESIGN AND IMPLEMENTATION.....	29
4.1	Class Diagram	29
4.2	Sequence Diagram.....	30
4.3	System Architecture Diagram.....	30
4.4	Model Architecture Diagram.....	30
4.5	Entity-Relationship-Diagram (ERD).....	30
4.6	Data flow Diagram.....	30
4.7	Deployment Diagram.....	30
4.8	User scenarios	30
4.9	Tools and Technologies	31
4.9.1	Front-End technology	31
4.9.2	Back-End technology	31
4.9.3	Version Control tools.....	31
4.9.4	IDEs and Coding tools	32
CHAPTER 5	<PROJECT TITLE> TESTING AND EVALUATION.....	33
5.1	Section 5.1	33
5.2	Section 5.2	33
CHAPTER 6	CONCLUSIONS AND FUTURE WORK.....	34
6.1	Section 6.1	34
6.2	Section 6.2	34

List of Figures

<i>Figure number</i>	<i>Page Number</i>
Figure 1 - Waterfall methodology	5
Figure 2 - Use case diagram.....	20
Figure 3 - Class diagram.....	29
Figure 4 - System Architecture	30

List of Tables

<i>Table number</i>	<i>Page Number</i>
Table 1 - Market comparison	8
Table 2 - Features Comparison.....	9
Table 3 - Task profile	18
Table 4 - User profile	19
Table 5 – Press emergency button (use-case 01).....	21
Table 6 - Manage patient memories (use-case 02).....	22
Table 7 - Manage patient safe zone (use-case 03).....	23
Table 8 – Safe zone breach (use-case 04).....	24
Table 9 - Manage appointment (use-case 05).....	25
Table 10 - Manage medical history (use-case 06)	26
Table 11 - Initial assessment (use-case 07)	27

List of Abbreviations

Chapter 1 Introduction

1.1 Background and Motivation

Alzheimer's disease profoundly affects memory, cognitive functions, and daily life, creating immense challenges for both patients and caregivers. Traditional caregiving methods, which often depend on manual tracking and personal oversight, are insufficient to address the complexities patients face, such as medication management, personal safety, and daily routines.

The motivation for developing **AlzCare** arises from the need to create innovative, technology-driven solutions that empower patients to maintain independence. Through features like reminders, face recognition for familiar people, and emergency alerts, the app provides essential support to patients. For caregivers, **AlzCare** offers real-time tools for monitoring, communication, and intervention, enabling them to ensure the well-being of their loved ones more efficiently and effectively.

1.2 Problem Statement

Caring for Alzheimer's patients presents numerous challenges, including managing daily activities, ensuring patient safety, and maintaining effective communication between patients and caregivers. Traditional caregiving methods often fall short, lacking the real-time support needed for medication adherence, accurate patient location tracking, and timely detection of emergencies. Additionally, patients with Alzheimer's frequently experience memory loss and disorientation, making it difficult to recognize loved ones or navigate familiar surroundings.

To overcome these challenges and improve the quality of care, there is an urgent need for a comprehensive, integrated application. Such a solution should provide essential features like medication reminders, face recognition to aid memory retention, geofencing for safety alerts, and AI-powered tools for real-time assistance and caregiver education.

1.3 Objectives

The **AlzCare** app aims to assist intermediate-level Alzheimer's patients by offering critical support features such as medication reminders and geolocation-based safe zone monitoring. The app will notify caregivers in real time if a patient leaves a designated safe area or misses a scheduled medication dose, ensuring patient safety and more effective caregiving. By empowering patients to maintain greater independence and reducing their reliance on constant supervision, the app will help alleviate the sense of being a burden on loved ones.

1.4 Project Scope and Limitations

Scope:

Admin:

- Manage user accounts (patients and caregivers), including creating, updating, and suspending accounts.
- Assign caregivers to specific patients for personalized care.
- Monitor overall system usage, user activity, and generate analytics reports.
- Send announcements and updates to users, ensuring timely communication of important information.
- Manage and oversee emergency protocols and escalations for critical situations.

Patient:

- Receive automated medication reminders to ensure timely adherence to prescribed treatments.
- View and manage saved places using geolocation, providing orientation and familiarity.
- Access memory training exercises designed to strengthen cognitive functions.
- Utilize face recognition to identify and recall familiar people, improving daily interactions.
- Sync with smartwatches to monitor health metrics and enhance overall well-being.
- Trigger emergency actions to alert caregivers in case of distress or emergencies.
- Participate in interactive cognitive tests to track and improve mental health.

Caregiver:

- Monitor the patient's real-time location using geofencing, with alerts for leaving designated safe zones.
- Receive notifications for missed medications, emergencies, or significant changes in patient behavior.
- Control and set up green areas where the patient can move freely, ensuring safety.
- Receive waste management alerts for improved caregiving efficiency.
- Participate in a community forum to exchange advice and experiences with other caregivers.
- Interact with a chatbot to receive caregiving tips, support, and assistance for daily care tasks.

Project limitation:

- **Face Recognition Accuracy:** The reliability of the face recognition feature may be influenced by the quality of images used and the lighting conditions. Additionally, the system may struggle to accurately identify individuals if the patient's face database is outdated or incomplete.
- **Real-Time Location Tracking:** The accuracy of patient tracking via GPS is dependent on the availability of a strong internet connection and good GPS signal, particularly in rural or low-signal areas. Signal interruptions can delay or provide inaccurate location updates.
- **Emergency Response:** While the app can send emergency alerts to caregivers, it cannot guarantee immediate physical response or assistance. The app relies on caregivers receiving and acting on notifications promptly.
- **AI Chatbot Effectiveness:** The AI chatbot's performance is dependent on the quality and breadth of the training data. The chatbot may not always provide accurate or contextually relevant advice, especially in complex caregiving situations.
- **Privacy and Data Security:** Although security measures will be in place, the app involves sensitive personal data such as health information, location, and facial recognition. Any security breach could expose users to privacy risks.
- **User Adoption and Learning Curve:** Some users, particularly elderly patients, may find it challenging to adopt new technologies, potentially limiting the effectiveness of certain features, such as memory training or emergency actions.

1.5 Project Methodology

Used methodology: Waterfall methodology.

For this project we have chosen the waterfall model, a linear and sequential approach to software development. Each phase in the Waterfall model must be completed before moving on to the next, ensuring clear structure and well-defined goals. It suits the AlzCare app due to its focus on thorough planning, patient care, and safety-critical requirements.

Sequential Phases:

The project will be developed in a series of distinct, sequential phases, where progress flows in one direction – from requirements gathering to deployment. Each phase must be fully completed before moving to the next, minimizing risk and ensuring all specifications are met.

1. Requirements Gathering:

- **Objective:** To collect detailed functional and non-functional requirements for the system.
- **Activities:**
 - Interview caregivers, patients, and healthcare professionals.
 - Document all essential system features (e.g., medication reminders, geolocation tracking, face recognition) and system constraints (e.g., data security, device compatibility).

2. System Design:

- **Objective:** To create a blueprint of the system architecture and UI/UX.
- **Activities:**
 - Design the system architecture, database structure, and user interfaces.
 - Plan integrations with external systems like smartwatches and geolocation services.
 - Design the flow of emergency actions and caregiver notifications.

3. Implementation:

- **Objective:** To build the system based on the detailed design specifications.
- **Activities:**
 - Develop the backend logic for user management, location tracking, emergency alerts, etc.
 - Implement the front-end user interface for patients and caregivers.
 - Integrate APIs for smartwatch syncing and GPS tracking.

4. Testing:

- **Objective:** To ensure the system meets all requirements and works as intended.
- **Activities:**
 - Perform unit testing for individual components
 - Conduct system testing to verify the functionality of all features.
 - Execute user acceptance testing with actual caregivers and patients to gather feedback and fix any issues.

5. Deployment:

- **Objective:** To launch the application for use by patients, caregivers, and administrators.
- **Activities:**
 - Deploy the application to the appropriate app stores and web servers.
 - Set up the backend infrastructure for monitoring and managing patient and caregiver data.
 - Configure real-time monitoring for system alerts and location tracking.

6. Maintenance and Support:

- **Objectives:** To provide ongoing support and make updates to the system.
- **Activities:**
 - Monitor the system for performance issues, user activity, and potential bugs.
 - Implement periodic updates for improved functionality and security.
 - Provide support to users through documentation and help desks.

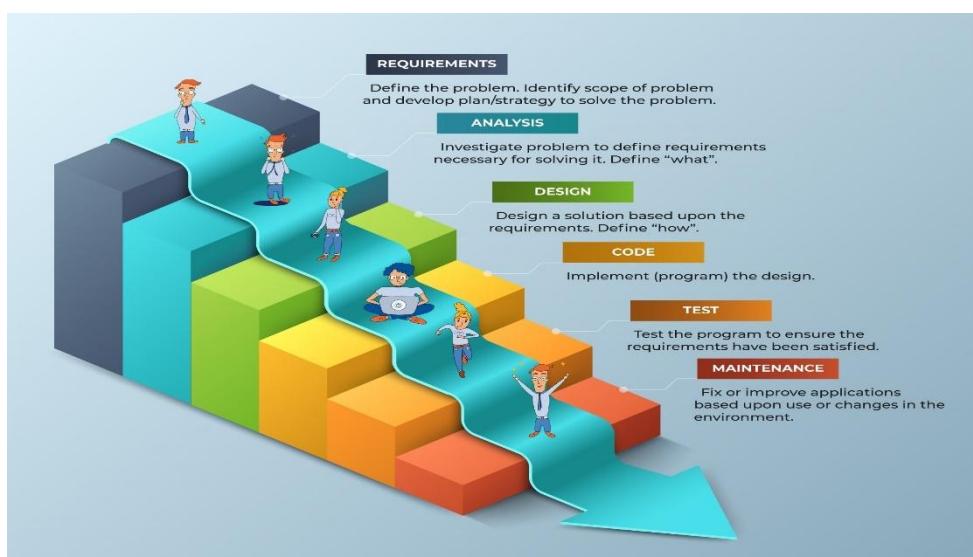


Figure 1 - Waterfall methodology

1.6 Project Report Outline

The following chapters of this report provide a comprehensive view of the design and analysis aspects of our Alzheimer patient support system.:

Chapter 2: Market and Literature Survey

This chapter explores the current solutions available for Alzheimer's patients, especially those in the early stages of the disease. It includes a detailed literature review of existing technologies and methodologies, identifying their strengths and limitations. This analysis helps to position our project within the context of current practices and highlights areas where our solution can bring innovation.

Chapter 3: System Analysis

This chapter defines the core problem and outlines the functional and non-functional requirements of the system. It identifies the key stakeholders, their roles, and responsibilities in the project. Detailed system scenarios and essential UML diagrams, such as use case, class, and sequence diagrams, are presented to visualize and clarify the system's design.

Chapter 4: System Design and Implementation

This section provides a thorough description of the design and implementation process for the Alzheimer's support system. It includes the system's architecture, user interface prototypes, data models, and the rationale behind the choice of programming languages and development tools. The chapter also showcases prototype screens that illustrate the user interaction with the system.

Chapter 5: System Testing and Evaluation

Here, we detail the test cases used to validate the system's performance and effectiveness. The chapter includes testing tools, results for each system requirement, and an evaluation of the project in terms of cost, time, environmental impact, ethics, and social and political implications.

Chapter 6: Conclusions and Future Work

This chapter summarizes the project's findings, emphasizing how the system addresses the needs of Alzheimer's patients in the early stages. It also outlines potential future enhancements to expand the system's capabilities and improve its support for patients and caregivers.

Chapter 2 Market and Literature Survey

2.1 Market comparison

Platform	Features/Functionality	Pros	Cons
alzbetter.com	Training for caregivers on Alzheimer's symptoms (e.g., hallucinations, agitation, wandering), with techniques to remind patients of nostalgic content (shows, movies, music).	Comprehensive caregiver training, tailored Alzheimer's care, and emotional support techniques.	Limited to training; lacks interactive patient-caregiver features.
alzconnected.org	Forum for sharing personal experiences and stories related to Alzheimer's, where patients, caregivers, family members, and professionals contribute posts.	Rich user-generated content, helpful insights from real-life experiences, community support.	No interactive or care management features; solely a discussion forum.
trualta.com	Provides caregiver training for multiple conditions, including Alzheimer's, diabetes, and heart disease. Offers online courses and resources for non-healthcare professionals in caregiving	Extensive training for a wide range of conditions, flexible options for novice caregivers.	General caregiving focus; lacks Alzheimer-specific or AI-driven features.
Rafiq Al-Zakira	Stores images/names of relatives and caregivers, medication and daily reminders, cognitive tests, location tracking, and face recognition using AI.	- Cognitive tests - Location tracking via Google Maps - Cross-platform UI.	- No Caregiver Interface - AI face recognition might need refinement.

Alzheimer Care Partner	Medication reminders, journal entries, caregiver collaboration, symptom tracking, and resource library. Includes a tracker for monitoring symptom progression.	Detailed symptom tracking, caregiver collaboration, and a resource library, making it useful for Alzheimer's care.	Limited medication database: resource library is static and lacks AI-driven personalization.
Life360	Location tracking, location history, and quick notes for reminders (e.g., medication reminders).	Real-time location tracking, accessible history of locations, flexible reminders.	General family-tracking app, not specifically designed for Alzheimer's or elderly care.
Alzheimer's App	Provides memory tests, Alzheimer's-related articles, notifications for test results, and basic reporting on cognitive assessment.	Easy access to cognitive tests, educational resources, and test result notifications.	Limited scope, mainly focused on assessments rather than ongoing support or caregiver tools.
CaringBridge	Private journal for health updates, meal and care planning, community support network for friends and family, ad-free experience focused on privacy and patient-caregiver connection.	Enables secure communication, community-driven support, and organizational tools for care planning.	Limited features beyond journaling and community support; not specific to Alzheimer's.

Table 1 - Market comparison

2.2 Feature comparison

Feature\Platform	AIZacare	Alzbetter	Rafiq Al-Zakira	Alzheimer Care Partner	Life360	Alzheimer App	Caring-Bridge	Alz connected
Forum for sharing personal experiences	✓	✗	✗	✗	✗	✗	✗	✓
Reminders for medication	✓	✗	✓	✓	✓	✗	✗	✗
Journal entries for tracking care	✓	✗	✗	✓	✗	✗	✓	✗
Symptom tracking	✗	✗	✗	✓	✗	✗	✗	✗
Community support	✓	✓	✗	✗	✗	✗	✓	✓
Cognitive tests	✓	✗	✓	✗	✗	✓	✗	✗
Location tracking	✓	✗	✓	✗	✓	✗	✗	✗
AI face recognition	✓	✗	✓	✗	✗	✗	✗	✗
Personalized resource library	✓	✗	✗	✓	✗	✓	✗	✗
Health data analytics	✗	✗	✗	✗	✗	✗	✗	✗
Flexible reminders	✓	✗	✓	✗	✓	✗	✗	✗

Table 2 - Features Comparison

2.3 SWOT analysis

3.5.1 Strengths:

1. **Innovative Features:** AlzCare offers unique and vital features such as medication reminders, face recognition for familiar people, and emergency alerts, all tailored to meet the specific needs of Alzheimer's patients.
2. **Real-time Monitoring:** The app provides real-time tracking of patients' activities and locations, which enhances patient safety and allows caregivers to act promptly in case of emergencies.
3. **Empowers patients:** By helping patients maintain their independence (e.g., through reminders and alerts), the app alleviates the emotional burden on both patients and caregivers.
4. **User-Centric design:** Designed with simplicity in mind, AlzCare is easy to use for both Alzheimer's patients and their caregivers, who may have varying levels of tech proficiency.
5. **Data integration and AI support:** Leveraging AI for real-time assistance and caregiver education can improve the quality of caregiving and decision-making

3.5.2 Weakness:

1. **Technological barriers:** Some Alzheimer's patients may face challenges with smartphone use, especially if they are not familiar with technology. This could limit app adoption.
2. **Dependence on Internet Access:** Features like real-time location tracking, medication reminders, and alerts require a reliable internet connection, which may not always be available in remote areas.
3. **Battery consumption:** Real-time tracking and face recognition may consume significant battery power, leading to shorter usage times.

3.5.3 Opportunities:

1. **Growing Market for Elderly Care Solutions:** The aging population and increasing awareness of Alzheimer's disease create a large and expanding market for healthcare apps like AlzCare.
2. **Partnerships with Healthcare Providers:** Collaborations with healthcare institutions, senior living communities, or doctors could increase the app's credibility, adoption, and distribution.
3. **Global Expansion:** As Alzheimer's awareness increases worldwide, the app has the potential to expand beyond regional markets, offering localization features for multiple languages and cultures.

4. **Technological Advancements:** Incorporating new advancements in AI, wearable health technologies, and improved geolocation tracking can improve the app's functionality and offer additional services, such as predictive health analytics.
5. **Government Support and Grants:** Governments or non-profit organizations may offer financial support, incentives, or grants for technology solutions aimed at improving elderly care.

3.5.4 Threats

1. **Competition from Other Health Tech Solutions:** Several companies already offer similar solutions for elderly care and Alzheimer's patients, such as wearable trackers and medication reminder apps. Established competitors may pose a significant challenge.
2. **Data Privacy Concerns:** Given the sensitivity of patient data, the app must comply with privacy regulations (e.g., HIPAA, GDPR) to avoid legal issues. Any data breach could harm the app's reputation and user trust.
3. **Economic Downturns:** Economic factors could affect potential users' willingness to pay for premium services, especially if the app is subscription-based.
4. **Technological Challenges:** The app's reliance on complex technologies like AI and face recognition could encounter challenges in terms of accuracy and reliability, particularly with patients who have advanced Alzheimer's symptoms.
5. **Regulatory Changes:** Any changes in healthcare laws or mobile app regulations could require significant updates or adaptations to ensure compliance, potentially disrupting the app's operations.

Chapter 3 AlzCare Analysis

This chapter describes the main problem functional, and non-functional requirements, stakeholders and their responsibilities, and system scenarios. Students are free to elicit a suitable set of UML diagrams to fully describe the system in detail. Examples of essential UML diagrams are use case, class, and sequence diagrams.

3.1 Functional requirements

3.1.1 Patient Functional Requirements

3.1.1.1 Medication reminder

- The system must provide medication reminders to the patient with dosage instructions and specific notes, such as whether the medication requires an empty stomach.
- Patients must be able to add or delete reminders as needed.

3.1.1.2 Face Recognition

- Patients must be able to upload or take a picture to identify family members with relevant details.
- For unknown or famous individuals, the system must gather relevant information from the internet.

3.1.1.3 Location Memory

- Patients must be able to save and manage important locations, such as home, work, and relatives' homes.

3.1.1.4 Important Date Reminders

- The system must provide reminders for significant dates like birthdays and anniversaries.

3.1.1.5 Emergency Shortcut

- A quick shortcut must be available for the patient to notify the caregiver immediately in case of an emergency.

3.1.1.6 Memory-Strengthening Games

- The system must include games like Sudoku, memory puzzles, and location-matching games to help improve cognitive skills.

3.1.1.7 Daily Memory Journal

- Patients document daily memories and add photos to create a memory journal.

3.1.1.8 Initial Assessment

- Upon registration, the system must conduct an interactive test to estimate the patient's cognitive stage.
- The system must calculate the assessment score and classify the degree of disease.

3.1.2 Caregiver Functional Requirements

3.1.2.1 Medication Compliance Alerts

- The system must notify caregivers if the patient does not take their medication within 15 minutes of the scheduled time.
- The system must inform caregivers once the medication has been taken.

3.1.2.2 Doctor Appointment Tracker

- Caregivers must be able to add, edit, and delete doctor appointments.
- The system must provide reminders with appointment details, such as the clinic location, doctor's name, and contact information.

3.1.2.3 Patient Location Tracking

- Caregivers must have access to the patient's real-time location whenever required.

3.1.2.4 AI Chatbot Assistance

- Caregivers must be able to interact with an AI chatbot to receive Alzheimer's care-related assistance.

3.1.2.5 Safe Zone Alerts

- The system must allow caregivers to set safe zones with specific time ranges.
- Notifications must be sent to caregivers if the patient leaves the predefined safe zone during the specified time.

3.1.2.6 AI-Powered Educational Resources

- The system must provide Alzheimer's care-related educational materials in a community section, curated by AI.

3.1.2.7 Medical History Management

- Caregivers must be able to log and manage the patient's medical records, including medications, tests, and treatments.

3.1.2.8 Shared Memories

- Caregivers must be able to share memories with the patient, which will appear in the patient's memory journal.

3.1.2.9 Community Support

- The system must include a community section for caregivers to ask questions and exchange information
- Connect with others who care for Alzheimer's patients.

3.1.3 Admin Functional Requirements

3.1.3.1 User Management

- The admin must be able to manage user accounts for both patients and caregivers, including creating, updating, and deleting accounts.

3.1.3.2 Role Assignment

- The admin must be able to assign or revoke caregiver roles for specific patients.

3.1.3.3 System Monitoring

- The admin must have access to monitor system performance, uptime, and error logs to ensure consistent functionality.

3.1.3.4 Data Security and Privacy

- The admin must ensure compliance with data privacy regulations and monitor data access permissions.

3.1.3.5 Content Moderation

- The admin must moderate content in the community support section to prevent misuse or sharing of harmful information.

3.1.3.6 Audit Logs

- The system must provide audit logs for the admin to track system usage, updates, and user actions.

3.2 Non-functional requirements

3.2.1 Usability:

- The app has a simple, intuitive user interface to ensure ease of use for both Alzheimer's patients and caregivers.
- The design is accessible, with high contrast, large fonts, and easy navigation for patients with cognitive impairments.
- If an error occurs by a user, the application should explain what went wrong so that users can fix their errors

3.2.2 Performance:

- The app loads within 3 seconds and provides quick responses to user interactions to ensure smooth usage for patients and caregivers.
- The app can handle multiple real-time events (e.g., location tracking, notifications) without significant lag or delays.
- The app optimizes data synchronization to ensure that the real-time data (medication reminders, location alerts) are updated promptly.

3.2.3 Availability:

- The app should be available 24/7, with minimal downtime, to always support caregivers and patients.
- The app should have a cloud-based backend with redundancy measures to ensure high availability even in the event of server failure.

3.2.4 Scalability:

- The app can handle an increasing number of users (patients and caregivers) without performance degradation.
- It can integrate with new features in the future without major refactoring.

3.2.5 Security:

- The app uses end-to-end encryption for sensitive data, such as patient information and location data, to ensure privacy and security.
- Access control mechanisms should be implemented to ensure that only authorized caregivers can view and modify sensitive information about the patient.
- The app complies with relevant data protection regulations to ensure patient privacy and data security.

3.2.6 Reliability:

- The app should have a low crash rate (less than 1% of users experiencing crashes per month) and maintain consistent performance across different devices.
- The app should feature automatic backups of patient data, ensuring that in the case of an unexpected shutdown or error, data is not lost.

3.2.7 Maintainability:

- If the application needs maintenance, it must be done effectively and efficiently without affecting its quality, usability, or other features.

3.3 Stakeholders and their responsibilities

3.3.1 End-Users:

3.3.1.1 Patients

- Receive automated medication reminders to ensure timely adherence to prescribed treatments.
- View and manage saved places using geolocation, providing orientation and familiarity.
- Access memory training exercises designed to strengthen cognitive functions.
- Utilize face recognition to identify and recall familiar people, improving daily interactions
- Sync with smartwatches to monitor health metrics and enhance overall well-being.
- Trigger emergency actions to alert caregivers in case of distress or emergencies.
- Participate in interactive tests to track and improve mental health.

3.3.1.2 Caregivers

- Monitor the patient's real-time location using geofencing, with alerts for leaving designated safe zones.
- Receive notifications for missed medications, emergencies, or significant changes in patient behaviour.

- Control and set up green areas where the patient can move freely, ensuring safety.
- Receive waste management alerts for improved caregiving efficiency.
- Participate in a community forum to exchange advice and experiences with other caregivers.
- Interact with a chatbot to receive caregiving tips, support, and assistance for daily care tasks. Use the AI chatbot for caregiving tips and assistance with daily care tasks.
- Communicate regularly with healthcare professionals and family members regarding the patient's progress and care needs.

3.3.2 System Administrators

3.3.2.1 Admins

- Manage user accounts (patients and caregivers), including creating, updating, and suspending accounts.
- Assign caregivers to specific patients for personalized care.
- Monitor system usage, user activity, and generate analytics reports.
- Send announcements and updates to users, ensuring timely communication of important information.
- Manage and oversee emergency protocols.

3.3.2.2 Developers

- Develop and maintain the app, ensuring that it runs smoothly and all features function as intended.
- Design a user-friendly interface for both patients and caregivers, keeping accessibility in mind for users who may have limited technological literacy.
- Develop and improve the AI chatbot and face recognition systems, ensuring that they are accurate, secure, and user-friendly.
- Ensure the app's technical architecture supports real-time data processing and integrates effectively with wearable devices and external health metrics.
- Continuously update the app to fix bugs, improve performance, and ensure compatibility with new devices or OS updates.

3.4 Task profile

Task	Admin	Patient	Caregiver
Manage User Account	✓		
Send announcements and updates	✓		
Set emergency protocols	✓		
Monitor system usage and analytics	✓		
Receive medication reminders		✓	
View/manage saved places		✓	
Trigger emergency actions		✓	
Participate in memory exercises		✓	
Use face recognition for familiar faces		✓	
Track real-time location			✓
Set geofencing zones			✓
Receive missed medication alerts			✓
Participate in a caregiver forum			✓
Interact with the AI chatbot		✓	✓

Table 3 - Task profile

3.5 User profile

User	Patient	Caregiver	Admin
Age	50+	20-40	30-60
Language	Arabic – English	English	English
Tech Experience	Low to medium	Medium	High
Domain Expertise	Low to medium	Medium	High
Task Knowledge	Low (Improves with use)	Medium (Improves with use)	High
Expectations	Ease of use	Ease of use	Comprehensive

Table 4 - User profile

3.6 Use Case Diagram

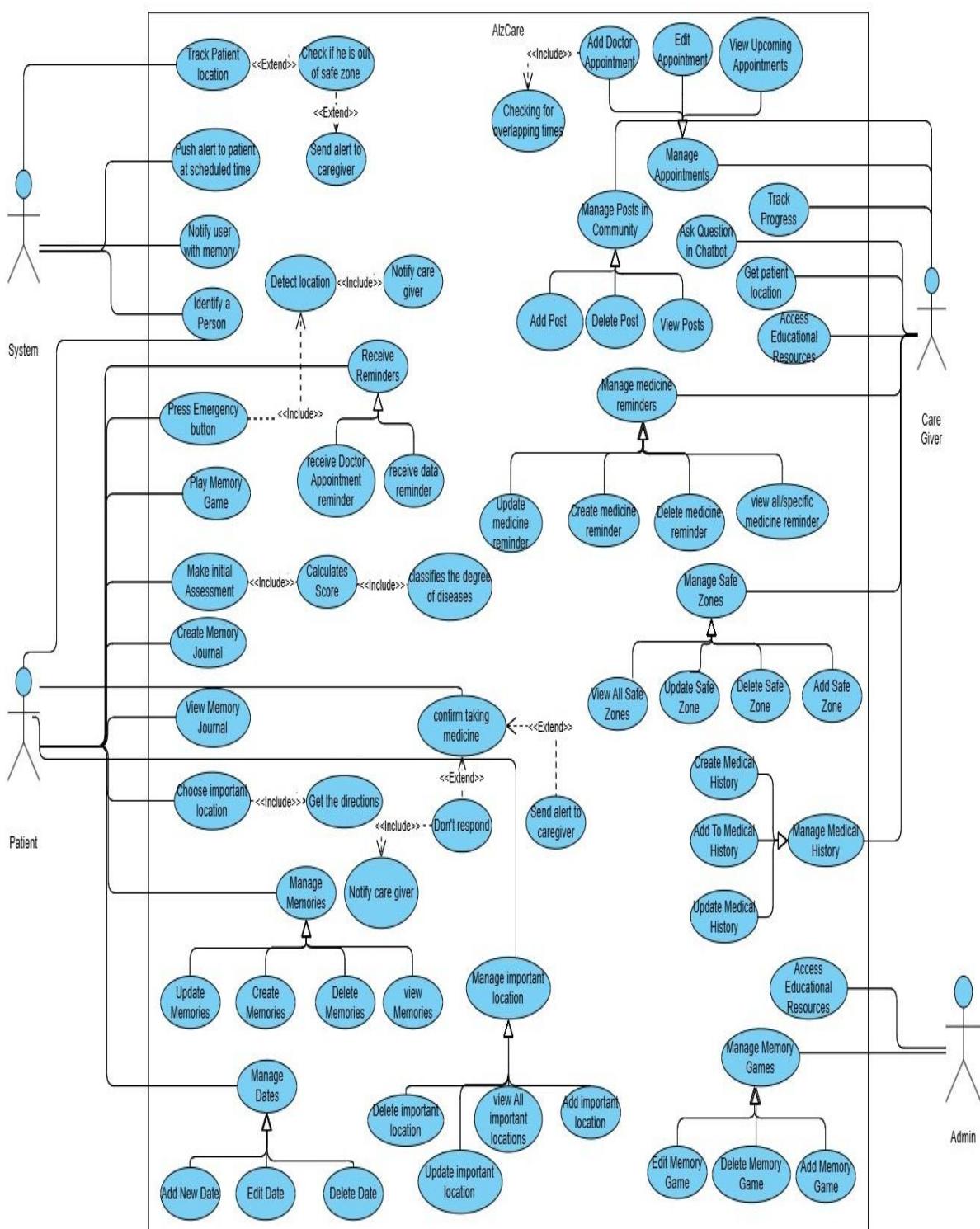


Figure 2 - Use case diagram

3.7 Use Case Scenario

3.7.1 Press emergency button

Table 5 – Press emergency button (use-case 01)

Use case ID:	01	
Use case name:	Press emergency button	
Actors:	Patient	
Pre-conditions:	1. Patient is logged into the system 2. Caregiver is registered to the system 3. Caregiver is assigned to patient	
Post-conditions:	Caregiver is notified about patient location and emergency	
Flow of events:	User Action	System Action
	1. Patient press the emergency button	
		2. System determines patient location
		3. System notifies the caregiver
	4. Caregiver receives the notification	
		5. System logs the emergency event in DB
Exceptions:	User Exceptions	System Exceptions
	1. Misuse or accidental press of the button.	
		2. Network issue causing delay notification
		3. Inaccurate location detection
		4. Notification delivery failure
Includes:	Location tracking for emergency help.	
Notes & Issues:	<ul style="list-style-type: none"> - The notification can be sent via email, or in-app notification. - System should retry location detection if it fails. 	

3.7.2 Manage patient memories

Table 6 - Manage patient memories (use-case 02)

Use case ID:	02	
Use case name:	Manage Patient Memories	
Actors:	Caregiver	
Pre-conditions:	1. The caregiver is logged into the system 2. Patient profile exists in system 3. Caregiver is assigned to patient	
Post-conditions:	The database for patient's memory is updated	
Flow of events:	User Action	System Action
	1. Caregiver log into system	
	2. Caregiver go to patient profile page	
	3. Caregiver adds, delete or update memory	
		4. System validates memory details
		5. System update patient memory in DB
Exceptions:	User Exceptions	System Exceptions
	1. Accidental deletion of critical data.	
	2. Invalid data submission	
		3. Accidental overwriting of critical data.
Includes:	No includes.	
Notes & Issues:	<ul style="list-style-type: none"> - Confirm before deletion and apply soft delete. - Insure data integrity during updates. 	

3.7.3 Manage safe zones

Table 7 - Manage patient safe zone (use-case 03)

Use case ID:	03	
Use case name:	Manage Safe Zone	
Actors:	Caregiver	
Pre-conditions:	1. The caregiver is logged into the system 2. Patient profile exists in system 3. Caregiver is assigned to patient	
Post-conditions:	The database for patient's safe zone is updated	
Flow of events:	User Action	System Action
	1. Caregiver log into system	
	2. Caregiver go to patient profile page	
	3. Caregiver adds, delete or update safe zone	
		4. System validates safe zone details
		5. System update patient safe zone in DB
		6. System confirm update to caregiver
Exceptions:	User Exceptions	System Exceptions
	1. Safe zone overlapping or conflict	
		2. Map rendering or visualization error
		3. Database update failed
Includes:	No includes.	
Notes & Issues:	<ul style="list-style-type: none"> - Confirm before deletion and apply soft delete. - Insure data integrity during updates. - Ensure proper visualization of safe zones on a map. 	

3.7.4 Safe zone breach

Table 8 – Safe zone breach (use-case 04)

Use case ID:	04	
Use case name:	Notify safe zone breach	
Actors:	System	
Pre-conditions:	1. Patient is logged into the system 2. Safe zone are defined for the patient 3. System is running and monitoring patient location	
Post-conditions:	<ul style="list-style-type: none"> • Caregiver is alerted about breach • System logs the breach event in DB 	
Flow of events:	User Action	System Action
	1. Patient moves outside the safe zone	2. System detects patient outside safe zone
		3. System sends alert to caregiver
		4. System notifies patient he's breaching
		5. System logs the breaching event in DB
Exceptions:	User Exceptions	System Exceptions
	1. Moving close to safe zone boundaries may cause confusion	2. Temporary GPS error
		3. False positive breach detection
		4. Notification delivery failure
Includes:	No includes.	
Notes & Issues:	<ul style="list-style-type: none"> - Test edge cases where the patient moves near the boundary of a safe zone. - Include an option for the caregiver to acknowledge or dismiss alerts. 	

3.7.5 Manage appointment

Table 9 - Manage appointment (use-case 05)

Use case ID:	05	
Use case name:	Manage appointment	
Actors:	Caregiver	
Pre-conditions:	1. Caregiver is logged into the system 2. Patient profile exists in the system 3. Caregiver is assigned to the patient	
Post-conditions:	<ul style="list-style-type: none"> • Remind patient with appointment 	
Flow of events:	User Action	System Action
	1. Caregiver log into system	
	2. Caregiver go to patient appointment page	
	3. Caregiver adds, delete or update memory	
		4. System validates appointment details
		5. System update patient appointment in DB
Exceptions:	User Exceptions	System Exceptions
	1. Invalid data submission	
	2. Appointment conflict	
		3. Database update failed
Includes:	Overlapping time validation is included in both adding and editing appointments.	
Notes & Issues:	<ul style="list-style-type: none"> - Confirm before deletion and apply soft delete. - Insure data integrity during updates. - Handle appointments conflict 	

3.7.6 Manage medical history

Table 10 - Manage medical history (use-case 06)

Use case ID:	06	
Use case name:	Manage medical history	
Actors:	Caregiver	
Pre-conditions:	1. Caregiver is logged into the system 2. Patient profile exists in the system 3. Caregiver is assigned to the patient	
Post-conditions:	<ul style="list-style-type: none"> • Remind patient with appointment 	
Flow of events:	User Action	System Action
	1. Caregiver log into system	
	2. Caregiver go to patient profile page	
	3. Caregiver select medical history	
	4. Caregiver adds, delete or update memory	
		5. System validates history details
		6. System update patient history in DB
		7. System confirm update to caregiver
Exceptions:	User Exceptions	System Exceptions
	1. Invalid data submission	
		2. Database update failed
Includes:	Validation for duplicate or conflicting entries in medical history.	
Notes & Issues:	<ul style="list-style-type: none"> - Confirm before deletion and apply soft delete. - Insure data integrity during updates. 	

3.7.7 Initial assessment

Table 11 - Initial assessment (use-case 07)

Use case ID:	07	
Use case name:	Initial assessment	
Actors:	Patient	
Pre-conditions:	1. Patient is logged into the system	
Post-conditions:	<ul style="list-style-type: none"> • System redirect patient to the main page 	
Flow of events:	User Action	System Action
	1. Patient open the application	
		2. System redirects patient to assessment page
	3. Solve the assessment	
		4. System calculates the score and classify the degree of disease
		5. System redirects patient to home page
Exceptions:	User Exceptions	System Exceptions
	1. Skip required questions	
	2. Exit the application	
		3. Error in redirecting
		4. Errors in calculation
Includes:	Validation and calculation and notify with result	
Notes & Issues:	<ul style="list-style-type: none"> - Ensure the assessment interface is user-friendly and accessible. - Provide clear guidance to patients about mandatory fields or unanswered questions. 	

3.8 Personas

3.9 Image processing model description

Chapter 4 AlzCare Design and Implementation

This chapter describes in depth how every aspect of the project was designed, implemented, or created. It should contain the system prototype screens and the used tools and programming languages along with selection reasons.

4.1 Class Diagram

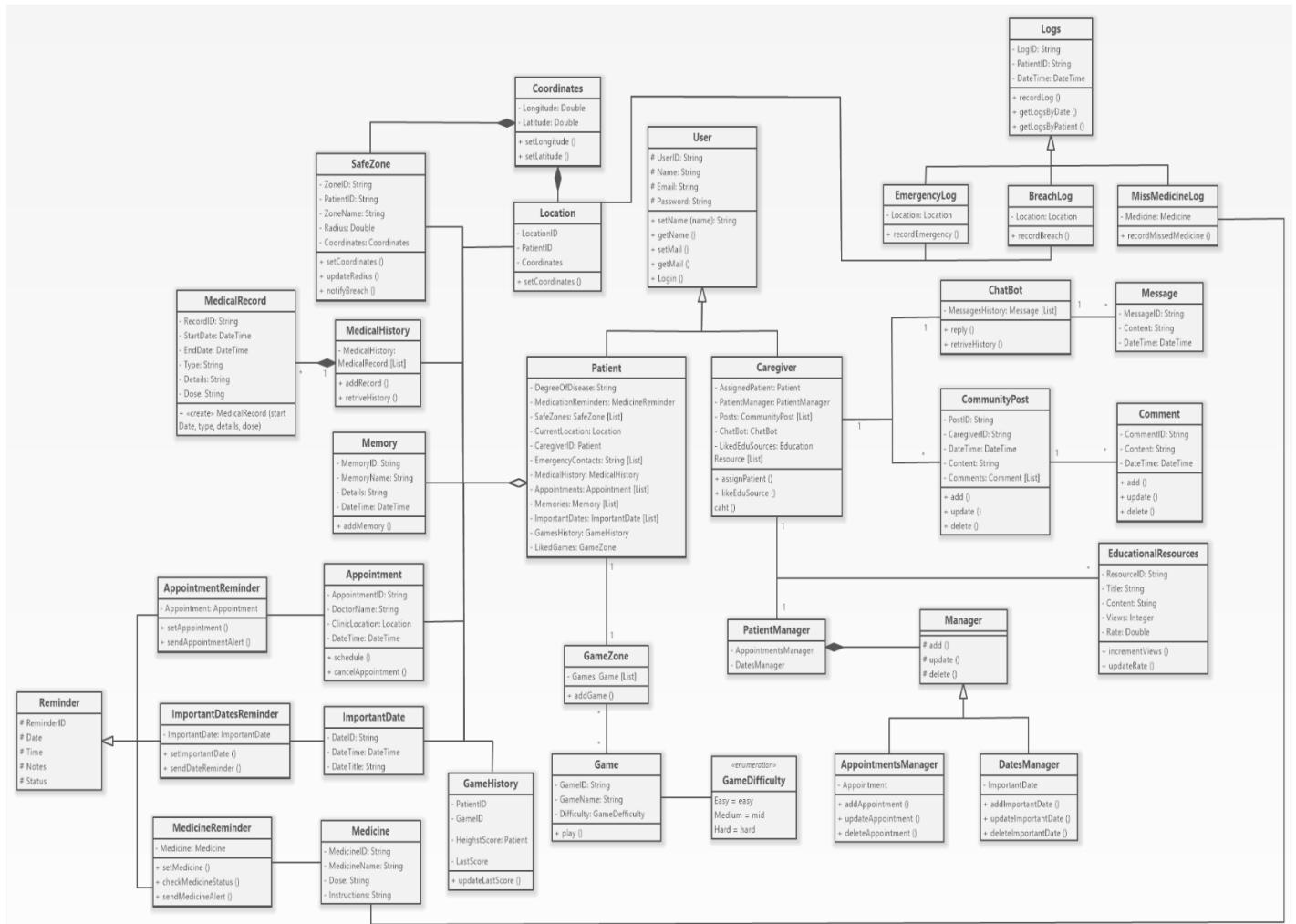


Figure 3 - Class diagram

4.2 Sequence Diagram

4.3 System Architecture Diagram

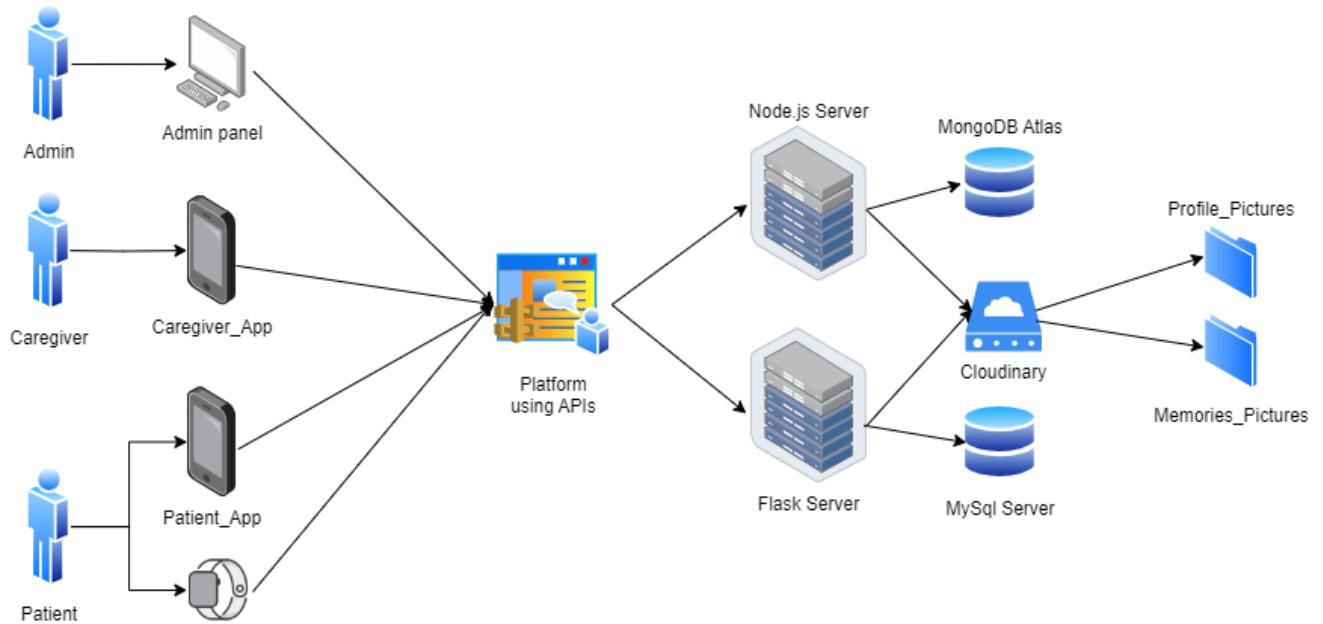


Figure 4 - System Architecture

4.4 Model Architecture Diagram

4.5 Entity-Relationship-Diagram (ERD)

4.6 Data flow Diagram

4.7 Deployment Diagram

4.8 User scenarios

4.9 Tools and Technologies

4.9.1 Front-End technology

- Framework: Flutter
 - Flutter is a modern UI toolkit for building natively compiled applications for mobile, web, and desktop from a single codebase.
 - It provides a rich set of pre-built widgets, fast performance, and a reactive framework for building dynamic user interfaces.

4.9.2 Back-End technology

- Node.js
 - Node.js is a server-side JavaScript runtime designed for building scalable and fast network applications.
 - It is used in this project to handle real-time operations, such as sending notifications and managing user sessions.
- Flask
 - Flask is a lightweight Python web framework ideal for creating robust back-end services and APIs.
 - It is used in this project to manage features such as AI-based assessments, face recognition, and other computational tasks.

4.9.3 Version Control tools

- Git
 - **Git** is employed as the version control system for the project.
 - It enables developers to:
 - Collaborate effectively.
 - Track changes to the codebase.
 - Revert to earlier versions when needed
 - The use of platforms like GitHub or GitLab is recommended to enhance collaboration and streamline workflow management.
- GitHub
 - **GitHub** is cloud-based hosting services for git repositories
 - It provides remote collaboration feature

4.9.4 IDEs and Coding tools

- **PyCharm**
 - A Python-specific **IDE** widely used for back-end development with frameworks like Flask and Django.
 - Provides excellent debugging tools, smart code navigation, and a variety of plugins to improve productivity.
- **VS Code**
 - A versatile and lightweight **code editor** used for front-end (Flutter, React) and back-end (Node.js, Python) development.
 - Features an extensive marketplace of extensions for JavaScript, TypeScript, and other programming languages, improving the coding experience.
- **Data Grip**
 - A powerful IDE for managing relational and NoSQL databases.
 - Offers intelligent query assistance, visual data exploration, and seamless version control integration.
- **MongoDB Compass**
 - A user-friendly **GUI tool** designed for MongoDB database management and exploration.
 - Provides schema visualization, query building, and in-depth data analysis capabilities.
- **WebStorm**
 - A specialized **IDE** for back-end JavaScript development, particularly with Node.js.
 - Offers intelligent code completion, built-in tools for debugging, and seamless integration with databases and version control systems.
- **Android Studio**
 - The official **IDE** for Android application development, supporting Kotlin, Java, and C++.
 - Includes a suite of tools for UI design, debugging, and performance profiling, along with built-in emulators.
- **Postman**
 - A powerful **API testing tool** for developing, testing, and documenting APIs in Node.js, Python, and other back-end technologies
 - Provides features for making HTTP requests, automating test scripts, and monitoring APIs for performance and reliability.

Chapter 5 <Project title> Testing and Evaluation

This chapter contains some of the detailed system test cases, used testing tools, and the test results of each system requirement. It also includes a project evaluation in terms of

- *Cost (if possible)*
- *Time compared to contemporary systems (if possible)*
- *Environmental impact (if any)*
- *Ethics (if any)*
- *Social and Political Impact (if any)*

5.1 Section 5.1

5.2 Section 5.2

Chapter 6 Conclusions and Future Work

This chapter includes conclusions to validate both the system need and how the presented system solved the problem stated. Finally, describe future suggestions to improve the system.

6.1 Section 6.1

6.2 Section 6.2