A logo of a globe with a graduation cap

Description automatically generated 

**Elder-Care Application**

**A Graduation project submitted in partial fulfillment of the requirements for DEPI.**

**Mobile Application Development**

***Team members***

1. Mohamed Sayed Helayel Mahmoud
2. Mohamed Khaled Sayed Elqalouby
3. Youssef Ramadan Abbas Kamel
4. Joseph Emad fawzy Shafik
5. Rowida Hussein Mahmoud Hassanien
6. Reham Tarek Mahmoud Abdelaziz

Cairo, Oct 2024

**Acknowledgment**

We would like to extend our sincere gratitude to all those who contributed to the success of this project. First and foremost, we are deeply thankful to the Ministry of Communication and Information Technology for their unwavering support and for providing us with the invaluable opportunity to participate in the Digital Egypt Pioneers Initiative.

My heartfelt thanks go to our mentors, who generously shared their knowledge and expertise. Their guidance, patience, and encouragement were essential in helping us overcome challenges and refine our skills throughout the program. The content and insights they provided will undoubtedly shape our future endeavors in the tech industry.

Finally, it has been a pleasure knowing everyone we worked with during this period.

**Abstract**

Elder Care is an innovative mobile application developed to address the specific needs of elderly patients, particularly those diagnosed with Alzheimer's disease or those living alone without regular care. The app is designed to simplify the process of medication management, which can often be a challenge for elderly individuals who may have difficulty remembering their medication schedules.

The application’s primary feature is its ability to provide timely, customizable reminders for medication intake. Users can easily input their medication details, and the app will ensure that alerts are sent at the correct times. If a reminder goes unnoticed or is not acknowledged within a certain period, the app takes proactive measures by automatically notifying a pre-designated contact. This contact can be a nurse, neighbor, caregiver, or a family member who will receive an alert that action may be required.

By bridging the gap between elderly patients and their support networks, Elder Care aims to ensure that critical medications are never missed, significantly reducing the risk of health complications due to forgotten doses. In doing so, the app enhances both the safety and independence of elderly patients, offering peace of mind to their families and caregivers.

In addition to medication reminders, the app may incorporate other features aimed at improving the quality of life for the elderly, such as providing a detailed medication history, access to nearby medical services, and the option to create a personalized health schedule. Overall, Elder Care is designed to be a reliable and user-friendly tool, offering comprehensive support for elderly individuals and contributing to their overall well-being.

Table of Contents

[Chapter1 Introduction and Background](#_Toc508086265) 5-6

[Chapter2 Literature Review](#_Toc508086265) 7

[Chapter3 System Analysis](#_Toc508086265) 8-12

[Chapter4 System Design](#_Toc508086265) 13-18

[Chapter5 System Implementation](#_Toc508086265) 19-21

[Chapter6 Testing methdology](#_Toc508086265) 22-

[Chapter7 Conclusion and future work](#_Toc508086265)

**Introduction**

As the global population continues to age, the number of elderly individuals living with chronic conditions like Alzheimer's disease, dementia, and other age-related challenges is steadily increasing. Many of these individuals live alone or with minimal supervision, which raises significant concerns regarding their ability to manage essential daily tasks, such as adhering to medication schedules. Proper and timely intake of medications is critical for maintaining their health, yet it is often one of the first responsibilities to be neglected as cognitive abilities decline.

To address this growing concern, we developed Elder Care, a mobile application specifically designed to support elderly individuals in managing their medication schedules effectively. The app is particularly aimed at those suffering from Alzheimer's or cognitive impairments and those living independently. Elder Care provides an intuitive solution that not only offers regular reminders for medication but also acts as a safeguard by notifying caregivers, family members, or neighbors if a reminder is missed, ensuring timely intervention.

Our project seeks to bridge the gap between elderly patients and their caregivers, enabling greater independence while reducing the risk of missed medications and the potential health complications that can arise from them. Through this initiative, we aim to provide a practical tool that enhances the quality of life for elderly individuals, allowing them to live more securely and autonomously in their homes while offering peace of mind to their families and caregivers.

**Problem Definition**

1. **Main Problem**:

Elderly patients, especially those with Alzheimer’s or living alone, often forget to take their medications on time, which can lead to severe health complications.

1. **Sub-problems:**

* **Medication Management:** Difficulty in keeping track of multiple medications and their schedules.
* **Emergency Response:** Lack of immediate assistance if a medication is missed.
* **User Accessibility:** Ensuring the app is user-friendly for elderly patients with limited technical skills.
* **Remote Monitoring:** Family members living abroad or far away cannot monitor the medication intake of their elderly relatives.

**Literature Review**

The challenge of medication management in elderly patients, particularly those with Alzheimer’s or cognitive decline, has been well-documented. Research highlights that medication non-adherence leads to increased hospitalizations and poor health outcomes. Kripalani et al. (2007) emphasize that forgetfulness, cognitive impairment, and complex regimens are major barriers for elderly individuals in adhering to prescribed medications.

Mobile health (mHealth) solutions have emerged to address these issues, with apps like Medisafe offering reminders to improve medication adherence. However, these apps often fall short for elderly patients with cognitive impairments, as noted by Aramendi et al. (2016), who stress that more robust solutions are needed. Existing apps typically lack the integration of caregiver support, a key element for Alzheimer's patients. Studies by Chan et al. (2020) found that involving caregivers through automatic notifications significantly improved adherence rates.

Despite the benefits of current apps, many are not user-friendly for older adults. As Kumar et al. (2022) note, intuitive interfaces and caregiver integration are critical in providing a comprehensive solution.

The Elder Care app addresses these gaps by offering not only medication reminders but also automatic notifications to caregivers when reminders are missed, ensuring timely intervention. This approach combines ease of use for elderly patients with a proactive system that supports both the patient and their caregivers, enhancing overall safety and well-being.

**System Analysis**

**System Objective**

1. **Main Objective:**

to provide a reliable and user-friendly mobile application that ensures elderly patients take their medications on time and receive immediate assistance if they miss a dose.

1. **Sub-objectives:**

* **Medication Reminders:** Provide timely notifications for medication intake.
* **Automatic Alerts:** Trigger automatic calls to emergency contacts if the alarm is not acknowledged.
* **User Authentication:** Secure user authentication using Google Sign-In.
* **Profile Management:** Allow users to manage their profiles and emergency contacts.
* **Data Storage:** Store medication schedules and user data securely in Firestore.

**System Features**

1. **User Authentication:**

* Google Sign-In: Secure and easy authentication using Google accounts.

1. **Medication Management:**

* Add/Edit Medication: Users can add or edit medication details, including name, dosage, schedule, and notes.
* Medication Overview: A dashboard displaying an overview of all medications and their schedules.

1. **Alarm System:**

* Timely Notifications: Schedule and display notifications for medication reminders.
* Background Services: Ensure alarms trigger even if the app is closed using background services.
* Missed Alarm Handling: Trigger automatic calls to emergency contacts if the alarm is not acknowledged within a specified period.

1. **User Profile Management:**

* Profile Screen: Manage user information and emergency contacts.
* Settings Screen: Customize app settings, including notification preferences and alarm sounds.

1. **Emergency Response:**

* Automatic Call Feature: Integrate with Twilio API to make automatic calls to emergency contacts if the alarm is missed.

1. **Data Storage:**

* Firestore Integration: Store user data, medication schedules, and alarm details securely in Firestore.

**Analysis**

**Day 1-2: Planning and Analysis**

1. **Define Requirements:**

* Functional Requirements: Medication schedule, alarm system, automatic call feature, user authentication, etc.
* Non-Functional Requirements: Performance, usability, reliability, etc.
* UML Diagrams: Context Diagram, Use Case Diagram, Activity Diagram , Sequence Diagram, Class Diagram and Erd Diagram .

1. **Team Roles:**

* Team Leader: Oversees the project, manages Jira tasks, and participates in development and testing (MS).
* Backend Developer:
* Sets up Firebase, handles database and authentication. (MS, MQ, YR, RT, RH)
* Frontend Developer: Designs UI/UX, implements Flutter screens. (MS, MQ, YR, RT, RH) Testing Engineer: Writes test cases, performs testing. (MS)
* DevOps Engineer: Manages Git, GitHub, CI/CD pipelines. (MS)

**Day 3-5: Design (MQ, RT, RH)**

**1. UI/UX Design:**

* Create wireframes and mockups for the app screens.
* Tools: Figma,Sketch. (MQ, RT, RH)

1. **Database Design:**

* Design Firestore database schema for storing user data, medication schedules, etc. (MS)

**Day 3-5: Setup and Initial Development (Parallel with Design)**

1. **Setup Project: (MS)**

* Initialize Flutter project in Android Studio.
* Set up Firebase project and integrate it with Flutter.
* Configure Google Sign-In.

1. **Version Control: (MS)**

* Initialize Git repository.
* Create GitHub repository and set up branch strategy

**Day 5-20: Core Development (MS, MQ, YR, RT, RH)**

1. **Authentication:**

* Implement Google Sign-In using Firebase Authentication.

1. **Medication Schedule:**

* Create screens for adding, editing, and viewing medication schedules.
* Store schedules in Firestore.

1. **Alarm System:**

* Implement alarm notifications using Flutter Local Notifications package.
* Set up background tasks to check for missed alarms.

**Day 16-18: Advanced Features ( MS , MQ , YR)**

1. **Automatic Call Feature:**

* Integrate Twilio API for making calls.
* Implement logic to trigger calls if the alarm is not canceled.

1. **User Management:**

* Create screens for managing user profiles and emergency contacts.

**Day 19-20: Testing and QA (MS)**

1. **Unit Testing:**

* Write unit tests for individual components.

1. **Integration Testing:**

* Test the integration of different modules.

1. **User Acceptance Testing (UAT):**

* Conduct UAT with a small group of users.

**Day 21-23: Deployment and Review**

1. **Deployment:**

* Set up CI/CD pipeline using GitHub Actions.
* Deploy the app to Google Play Store.

1. **Review and Feedback:**

* Collect feedback from users.
* Plan for future updates and improvements

**System Design**

**Use Case Diagram**

A diagram of a computer

Description automatically generated

**Context Diagram**

A screenshot of a computer

Description automatically generated

**Class Diagram**

A screenshot of a computer

Description automatically generated

**Activity Diagram**

A black and white image of text

Description automatically generated

**ERD**

A screenshot of a computer

Description automatically generated

**Sequence Diagram**

A screen shot of a computer

Description automatically generated

**System Implementation**

**Tools and Technologies**

**Flutter & Dart**: For app development.

**Firebase**: For back-end services.

**Android Studio**: For development environment.

**Git & GitHub**: For version control.

**Jira**: For project management.

By following this plan, your team should be able to develop the Elder Care app within the given timeframe. Good luck with your project! If you have any specific questions or need further details on any step, feel free to ask. To ensure that the alarm rings on time even if the app is not open and the phone is left somewhere, we need to implement background services and notifications effectively. Here’s a detailed breakdown of how this can be achieved and where it fits into each module of the implementation

**Alarm Functionality**

1. **Setting Up Background Services**

**Module**: Alarm System

**Implementation**:

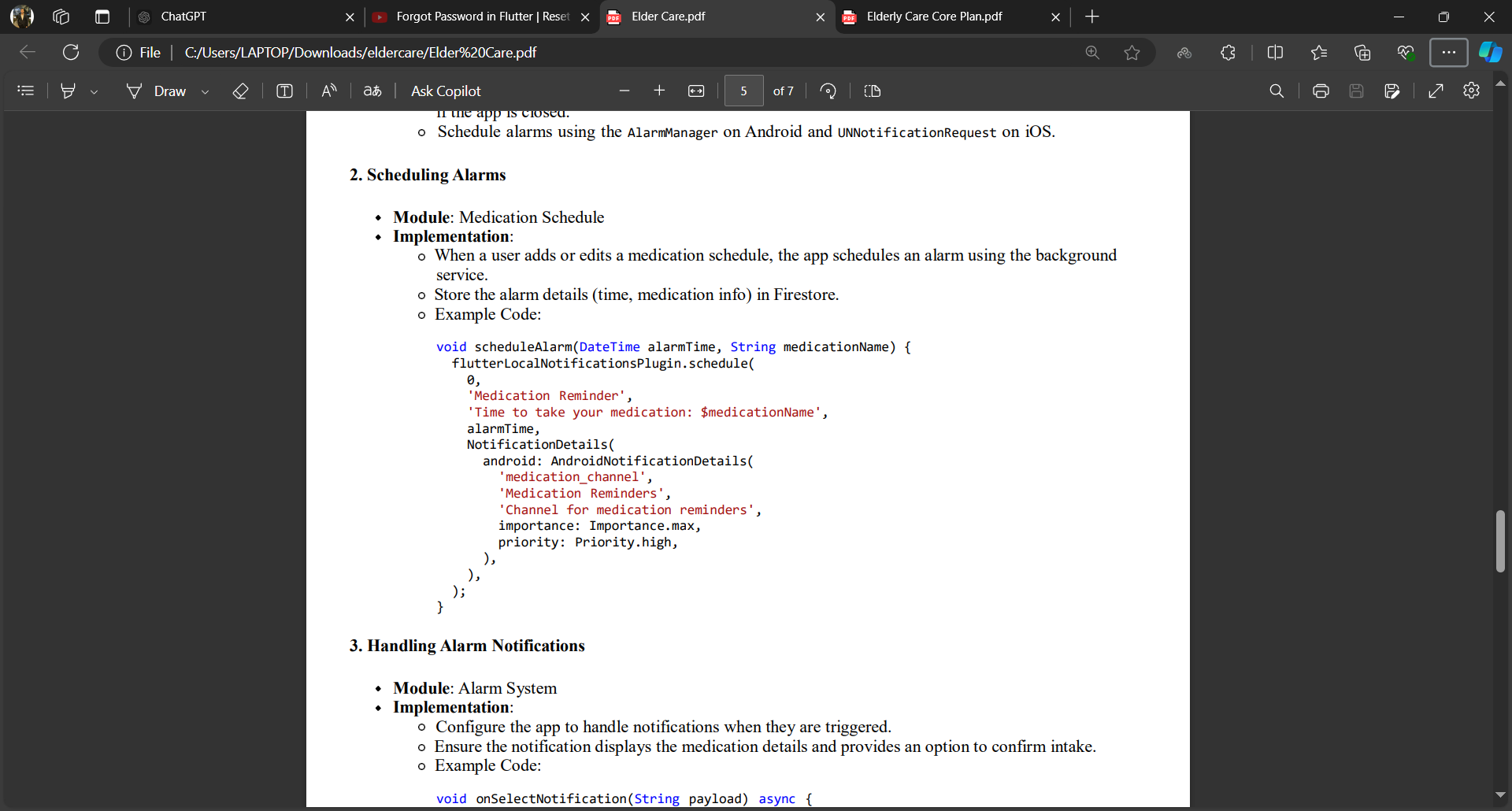
* Use the flutter\_local\_notifications package to schedule notifications.
* Implement background services using android\_alarm\_manager\_plus to ensure the alarm triggers even if the app is closed.
* Schedule alarms using the AlarmManager on Android and UNNotificationRequest on iOS.

1. **Scheduling Alarms**

**Module**: Medication Schedule

**Implementation**:

* When a user adds or edits a medication schedule, the app schedules an alarm using the background service.
* Store the alarm details (time, medication info) in Firestore.
* Example Code:



1. **Handling Alarm Notifications**

**Module**: Alarm System

**Implementation**:

* Configure the app to handle notifications when they are triggered.
* Ensure the notification displays the medication details and provides an option to confirm intake.
* Example Code

A screenshot of a computer

Description automatically generated

1. **Background Task for Missed Alarms**

**Module**: Alarm System

**Implementation**:

* Use background tasks to check if the alarm was acknowledged within a certain period.
* If not acknowledged, trigger the automatic call using Twilio API.
* Example Code:

A screenshot of a computer

Description automatically generated

1. **Automatic Call Feature**

**Module**: Advanced Features

**Implementation**:

* Integrate Twilio API to make calls.
* Implement logic to trigger the call if the alarm is not canceled within the specified time.
* Example Code:

A screenshot of a computer

Description automatically generated