



HELWAN UNIVERSITY
Faculty of
Computers and Artificial Intelligence
MEDICAL INFORMATICS Program

Alzheimer Care

A graduation project dissertation by :

| | |
|-------------------------------------|---------------------|
| <i>Haidy Arafat Galal Said</i> | <i>(20208280)</i> |
| <i>Rodina Salama AlSaid</i> | <i>(20208117)</i> |
| <i>Rodina Sameh Nour Eldin</i> | <i>(20208118)</i> |
| <i>Aya Waheed Eid Abdelazeem</i> | <i>(20208056)</i> |
| <i>Amany Abdallah Muhammad Adly</i> | <i>(20208037)</i> |

*Submitted in partial fulfilment of the requirements for the degree of Bachelor of Science
in Computers & Artificial Intelligence, at the **Medical Informatics** Department, the
Faculty of Computers & Artificial Intelligence, Helwan University*

Supervised by:

Dr. Helal Ahmed

June 2023



جامعة حلوان
كلية الحاسبات والذكاء
الاصطناعي
قسم المعلوماتية الطبية

رعاية مرض الزهايمر

رسالة مشروع التخرج مقدمه من :

- هايدي عرفات جلال سيد (20208280)
رودينا سلامه السيد احمد (20208117)
رودينه سامح نور الدين (20208118)
اية وحيد عيد عبدالعظيم (20208056)
اماني عبدالله محمد عدلي (20208037)

رسالة مقدمه ضمن متطلبات الحصول على درجة البكالوريوس في الحاسبات والذكاء الاصطناعي بقسم
المعلوماتية الطبية، كلية الحاسبات والذكاء الاصطناعي ، جامعة حلوان

تحت اشراف

د. هلال احمد

يونيو 2023

ACKNOWLEDGEMENT

First and foremost, we would express our deepest gratitude and appreciation to our supervisors Dr. Helal Ahmed & Dr. Rofanda Bekir for outstanding guidance, there continuous support and help as they were always here for us providing more than the needed time and effort from them since the very beginning of our project.

We would also like to thank our families, especially our parents, for their encouragement, patience, and assistance over the years. We are forever indebted to our parents, who have always kept us in their prayers and pushed us to Success.

Finally, for our faculty for providing the suitable environment that leaded us to represent the best image that computer science graduates of Helwan University are supposed to represent.

Table of Contents

| | |
|-----------------------|----------|
| Abstract | 6 |
|-----------------------|----------|

Chapter 1: Introduction

| | |
|-------------------------------|----|
| 1.1 Overview | 7 |
| 1.2 Objectives | 8 |
| 1.3 Purpose | 9 |
| 1.4 Scope..... | 9 |
| 1.5 General Constraints | 10 |

Chapter 2: Project “Planning and analysis”

2.1 Project planning

| | |
|---|-----|
| 2.1.1 Feasibility Study | 11 |
| 2.1.2 Gantt Char | 15 |
| 2.2 Analysis and Limitation of existing system..... | 17 |
| 2.3 Need for the new system..... | 17. |
| 2.4 Analysis of the new system | |
| 2.4.1 User requirements..... | 18 |
| 2.4.2 System Requirements..... | 19 |
| 2.4.3 Domain Requirements | 20 |
| 2.4.4 Functional Requirements..... | 21 |
| 2.4.5 Non- Functional Requirements | 23 |
| 2.5 Advantages of the new system..... | 24 |
| 2.6 Risk and Risk Managements..... | 25 |

Chapter 3: Software Design

| | |
|---|----|
| 3.1 Design of database (ERD or Class) Diagram | 27 |
| 3.2 Use case diagram | 28 |
| 3.3 sequence diagram | 32 |
| 3.4 activity diagram | 43 |

Chapter 4: Implementation

| | |
|---------------------------------|----|
| 4.1 Software architecture | 44 |
|---------------------------------|----|

Chapter 5: Testing

| | |
|------------------------------|----|
| 5.1 Unit Testing | 54 |
| 5.2 Integrated testing | 55 |
| 5.3 Additional Testing | 57 |

Chapter 6: Results and Discussion

6.1 Results

| | |
|------------------------------|----|
| 6.1.1 Expected results | 64 |
| 6.1.2 Actual results | 65 |

| | |
|----------------------|----|
| 6.2 Discussion | 66 |
|----------------------|----|

| | |
|-----------------------------|----|
| Chapter 7: Conclusion | 67 |
|-----------------------------|----|

| | |
|------------------------------|----|
| Chapter 8: Future work | 68 |
|------------------------------|----|

Abstract

This study aims to design a mobile application for Alzheimer's patients using Flutter technology to improve the quality of healthcare for patients and caregivers. The application is divided into two parts, one for patients and the other for doctors/healthcare providers. Doctors/caregivers can upload MRI scans and analyze them using image processing algorithms and machine learning to determine if the patient has Alzheimer's disease. The MMSE test can also be conducted by doctors/healthcare providers to determine the stage of the disease. The application includes a chatting section that allows doctors/healthcare providers to communicate with each other. The login data for doctors/caregivers and the chat messages are saved and stored.

Chapter 1

Introduction

1.1 Overview

The project is a mobile application developed using Flutter that aims to assist patients with Alzheimer's disease and their caregivers. The application is divided into two sections - one for patients/caregivers and one for doctors. The app uses image processing and machine learning algorithms to analyze MRI scans and the MMSE test to determine if the user is an Alzheimer patient or not and what his current stage of the disease is. Add to this, the application features securely storing and analyzing caregiver's / doctors' data (login information & chat messages).

1.2 Objectives

1. *provide a mobile application that can assist patients with Alzheimer's disease and their caregivers.*
2. *Develop an Image Processing system that uses Deep Learning algorithm “CNN” to accurately analyze MRI scans and administer the MMSE test to determine whether the user is an Alzheimer patient or not and the patient's current stage of the disease.*
3. *Provide high-level security features that include encrypting data and applying strict security rules.*
4. *Develop and manage an efficient database to store and retrieve caregiver's / doctors' data.*

1.3 Purpose

The main goal of this project is to develop a mobile application that can aid patients suffering from Alzheimer's disease and their caregivers in effectively managing the illness. The application utilizes state-of-the-art technology, such as image processing and machine learning algorithms, to accurately analyze MRI scans and conduct the MMSE test to determine the current stage of the disease. Chatting between the caregivers & doctors is also available. These objectives can be achieved by developing a user-friendly and secure mobile application that complies with medical regulations and ethical principles.

1.4 Scope

- Developing a mobile application using Flutter.*
- Designing and implementing image processing algorithms to analyze MRI scans and diagnose Alzheimer's disease.*
- Training machine learning models on a large dataset of clinical data to determine stages of the disease.*
- Designing and developing the user interface for the application, including displaying results and enabling communication between doctors and caregivers.*
- Securing data and protecting privacy, as well as complying with medical regulations and professional standards.*
- Creating project documentation and monitoring progress and timelines.*

1.5 General Constraints

- 1. The application must be developed in a manner that complies with healthcare data protection laws.*
- 2. The user interface must be designed in a way that makes it easy for patients to use the application.*
- 3. The application must be developed in a way that is scalable to support additional features in the future.*
- 4. The application must be able to work on multiple platforms such as laptops, tablets, and smartphones.*
- 5. Continuous technical support must be provided for the application to ensure it works correctly.*

Chapter 2

Project “Planning & Analysis”

2.1 Project Planning

2.1.1 Feasibility Study

Feasibility Study for Developing a Flutter Application for Alzheimer's Disease Patients

Feasibility is defined as the practical extent to which a project can be performed successfully. To evaluate feasibility, a feasibility study is performed, which determines whether the solution considered to accomplish the requirements is practical and workable in the software. Information such as resource availability, cost estimation for software development, benefits of the software to the organization after it is developed and cost to be incurred on its maintenance are considered during the feasibility study.

Introduction:

Alzheimer's disease is a progressive neurodegenerative disease that affects cognitive functioning, memory, and behavior. It is estimated that around 50 million people worldwide suffer from some form of dementia, with Alzheimer's disease being the most common type. To help Alzheimer's patients and their caregivers, a Flutter-based mobile application could be developed. The purpose of this feasibility study is to evaluate the technical, financial, and operational feasibility of developing a Flutter application for Alzheimer's patients.

Technical Feasibility:

Flutter is a mobile application development framework that allows for the creation of high-performance, cross-platform applications. It provides a rich set of UI widgets and tools that make it easy to build beautiful and responsive applications. The technical feasibility of developing a Flutter application for Alzheimer's patients is high, as the framework is well-suited for the task.

Financial Feasibility:

To develop a Flutter application for Alzheimer's patients, the costs associated with development, testing, and deployment must be considered. The cost of developing a Flutter application can vary depending on the complexity and functionality of the application. The cost can be minimized by using open-source libraries and frameworks, and by leveraging existing resources. The financial feasibility of developing a Flutter application for Alzheimer's patients is moderate, as it can require significant investment in development and maintenance.

Operational Feasibility:

The operational feasibility of a Flutter application for Alzheimer's patients depends on the ability of the application to meet the needs of the target users and to be integrated into their daily lives. The application should be easy to use, intuitive, and accessible to Alzheimer's patients, who may have cognitive and functional impairments. To ensure operational feasibility, user-centered design principles should be applied, and user testing should be conducted throughout the development process.

Market Feasibility:

The market feasibility of a Flutter application for Alzheimer's patients is high, as there is a growing demand for mobile applications and digital tools that can help patients and caregivers manage the disease. According to a report by Grand View Research, the global market for digital health is expected to reach \$509.2 billion by 2025, with a significant portion dedicated to applications for dementia and Alzheimer's disease.

Legal and Ethical Feasibility:

Developing a Flutter application for Alzheimer's patients requires compliance with legal and ethical standards. The application should comply with data privacy regulations and protect the personal information of users. It should also respect the autonomy and dignity of Alzheimer's patients and their caregivers and avoid stigmatizing or discriminating against them.

Conclusion:

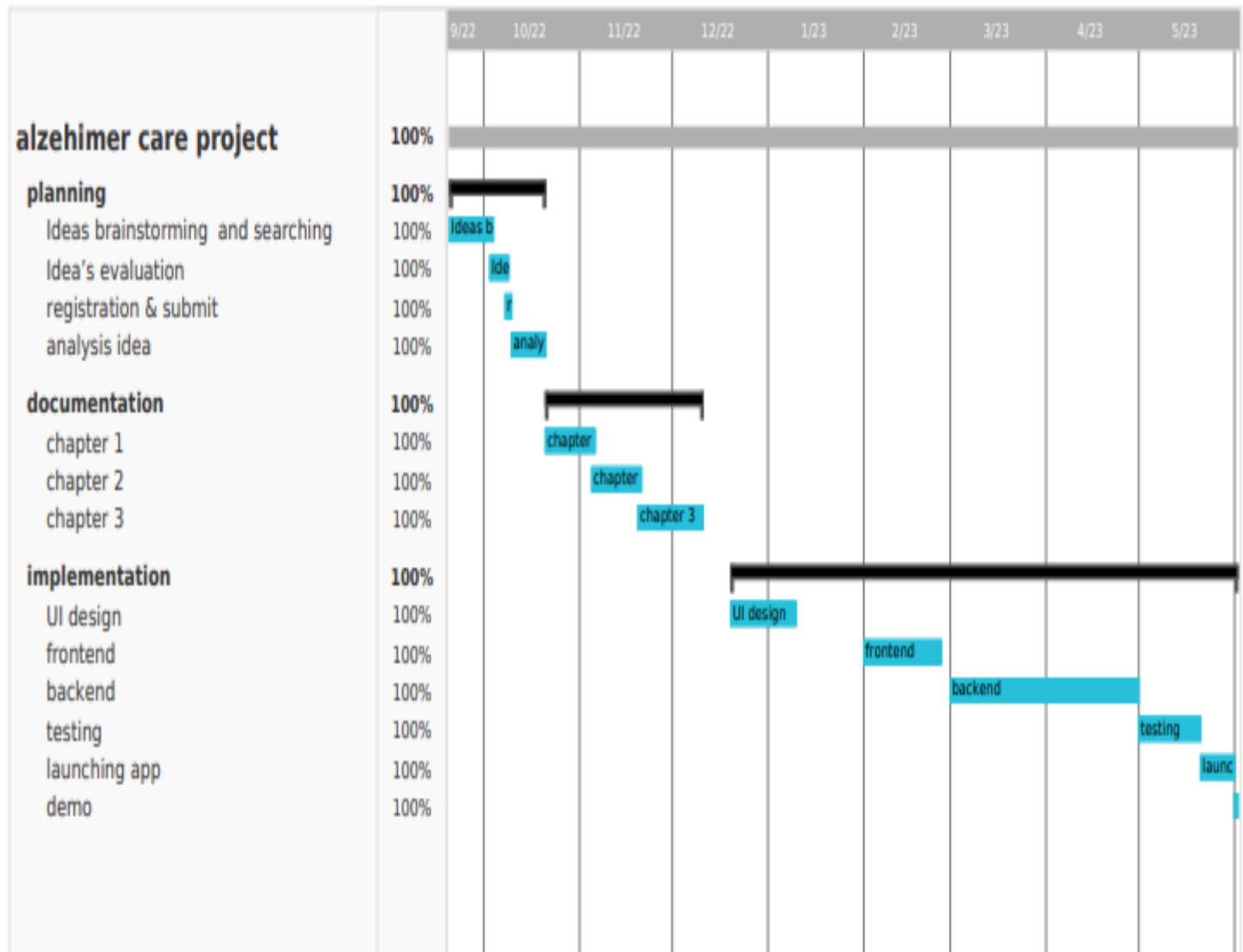
In conclusion, developing a Flutter application for Alzheimer's patients is technically feasible and has high market potential. However, it requires a significant investment in development and maintenance, and user-centered design principles should be applied to ensure operational feasibility. Legal and ethical considerations should also be addressed to ensure compliance with regulations and to protect the rights and dignity of users. Overall, the feasibility study suggests that developing a Flutter application for Alzheimer's patients is a viable and worthwhile endeavor that has the potential to improve the quality of life for patients and their caregivers.

2.1.2 Gantt Chart

| | Task Name | Start | End | Duration |
|----|-----------------------------------|------------|------------|----------|
| 1 | Total project | 20/9/2022 | 1/6/2023 | 215 |
| 2 | Ideas brainstorming and searching | 20/9/2022 | 3/10/2022 | 13 |
| 3 | Idea's evaluation | 3/10/2022 | 8/10/2022 | 5 |
| 4 | Registration & submit | 8/10/2022 | 9/10/2022 | 1 |
| 5 | Analysis idea | 10/10/2022 | 20/10/2022 | 10 |
| 6 | Documentation | 21/10/2022 | 10/12/2022 | 50 |
| 7 | Chapter 1 | 21/10/2022 | 5/11/2022 | 15 |
| 8 | Chapter 2 | 5/11/2022 | 20/11/2022 | 15 |
| 8 | Chapter 3 | 20/11/2022 | 10/12/2022 | 20 |
| 9 | UI Design | 20/12/2022 | 9/1/2023 | 20 |
| 10 | Implementation | 1 /2/2023 | 6/6/2023 | 116 |
| 11 | Frontend | 1/ 2/2023 | 25/2/2023 | 25 |
| 12 | Backend | 1/3/2023 | 30/4/2023 | 60 |
| 13 | Testing | 1/5/2023 | 20/5/2023 | 20 |
| 14 | Launching app | 21/5/2023 | 31/5/2023 | 10 |
| 15 | Demo video | 1/6/2023 | 1/6/2023 | 1 |



Created with Free Edition



2.2 Analysis & Limitation of existing system

There are no existing or similar systems to our application, this is the first version of our application, we started it from scratch by collecting & preparing data, training the model " CNN ", design the GUI and finally build the application.

2.3 Need for the new system.

We need to build an application that helps doctors and patients with Alzheimer's & their caregivers that contains:

- *A comprehensive and efficient communication platform for caregivers and doctors to share patient data and collaborate in real-time.*
- *Integration of machine learning and image processing technologies to aid in the diagnosis of Alzheimer's.*
- *A secure and centralized database to store patient data and ensure privacy.*
- *An easy-to-use interface for caregivers and doctors to access patient data and communicate with each other.*

2.4 Analysis of the new system

2.4.1 User Requirements

- *The mobile application should be easy to use and navigate for both caregivers and doctors.*
- *The application should allow doctors and caregivers to securely communicate and share information with each other.*
- *The application should be able to analyze MRI scans using image processing algorithms and machine learning algorithms to accurately diagnose Alzheimer's disease and determine its stage.*
- *The application should include an MMSE test that caregivers can administer to patients as a preliminary test for Alzheimer's disease.*
- *The application should securely store login credentials and chat messages between caregivers and doctors.*

2.4.2 System Requirements

- *The application should be developed using Flutter framework for mobile app development.*
- *The application should be compatible with Android and iOS mobile devices.*
- *The application should have a secure login system for caregivers and doctors to access the application.*
- *The application should use image processing and machine learning algorithms to analyze MRI scans and diagnose Alzheimer's disease accurately.*
- *The app should use secure storage to store login credentials.*
- *The application should be compliant with medical regulations and standards, such as HIPAA in the United States*

2.4.3 Domain Requirements

- *The application should be designed to assist patients with Alzheimer's disease and their caregivers in managing the disease and communicating with doctors.*
- *The application should be capable of analyzing MRI scans and administering MMSE tests to patients with Alzheimer's disease to provide accurate results for both.*
- *The application should be designed to improve the quality of care and support for patients with Alzheimer's disease and their caregivers.*
- *The app should comply with medical regulations and standards and should include information on the latest research and best practices related to Alzheimer's care.*
- *The app should ensure patient privacy and confidentiality.*

2.4.4 Functional Requirements

2.4.4.1 User Functional Requirements.

Caregiver / Doctor can

- *Register an account.*
- *Edit or delete his account.*
- *Login into his account.*
- *Do MMSE test for the patient.*
- *Upload or take a picture of MRI scan.*
- *Chat with each other.*
- *Log out from his account.*

2.4.4.2 System Functional Requirements

- *System should add any new accounts correctly.*
- *System should administer MMSE test to patients as a preliminary test and show its result accurately.*
- *System should allow doctors / caregivers upload MRI scans for analysis and provide accurate diagnosis of Alzheimer disease with determination of its stage.*
- *System should allow caregivers & doctors to securely communicate with each other through a chat feature.*
- *allow caregivers and doctors to securely communicate with each other through a chat feature.*

2.4.5 Non-Functional Requirements

- **Usability:**
The application should be user-friendly and intuitive for both caregivers and doctors to use.
- **Reliability:**
Having a good backend using Firebase we will have 99.95% uptime.
- **Maintainability:**
Our team will consistently test the app and fix any bugs (if found) ASAP.
- **Scalability:**
The App will be developed by adding new features.
- **Security:**
By using Firebase security metrics (App check) the data of our user “doctors or caregivers and their chats” should be protected from any malicious attack and protect patient privacy by complying with medical regulations and standards.
- **Performance:**
The project should have a fast response time and run smoothly & efficiently.
- **Compatibility:**
The application should be working on old and new android (Android 9.0+) and IOS smartphones (IOS 15.0+)

2.5 Advantages of the new system

- 1) Caregiver can upload patient data such as MRI rays and it will be analyzed by using image processing to know if the person has Alzheimer or not via machine learning algorithms.*
- 2) Doctors can make MMSE test to patient on the application to determine which degree of disease the patient in.*
- 3) Caregivers & doctors can also communicate with each other via chat.*

2.6 Risk and Risk Managements

During our application's development, we anticipated facing some issues. We created a list of certain anticipated risks and a risk management strategy to avoid these anticipated threats.

Some of these hazards that we are anticipated to face include the risk of running out of time because of the overload of responsibilities, as well as the risk of having a lot of tests and projects throughout the semester so we attempt to organize our time well to reduce this risk.

The risk associated with deciding which technology to use to develop our application is also there. To ensure that our product ran smoothly across all platforms and operating systems, we had to learn new software like Flutter. We took a chance with this.

Additionally, as we worked on our project, we faced some predicted technical issues that we attempted to properly resolve through research, experimentation, and consultations with experts.

We faced certain unexpected hazards, such as poor communication between team members and between team members and the project supervisor, but we were able to find solutions. Another significant unexpected issue that almost affected the project's development was a managerial oversight during the project's registration, but we are now getting organized and the project was successfully completed.

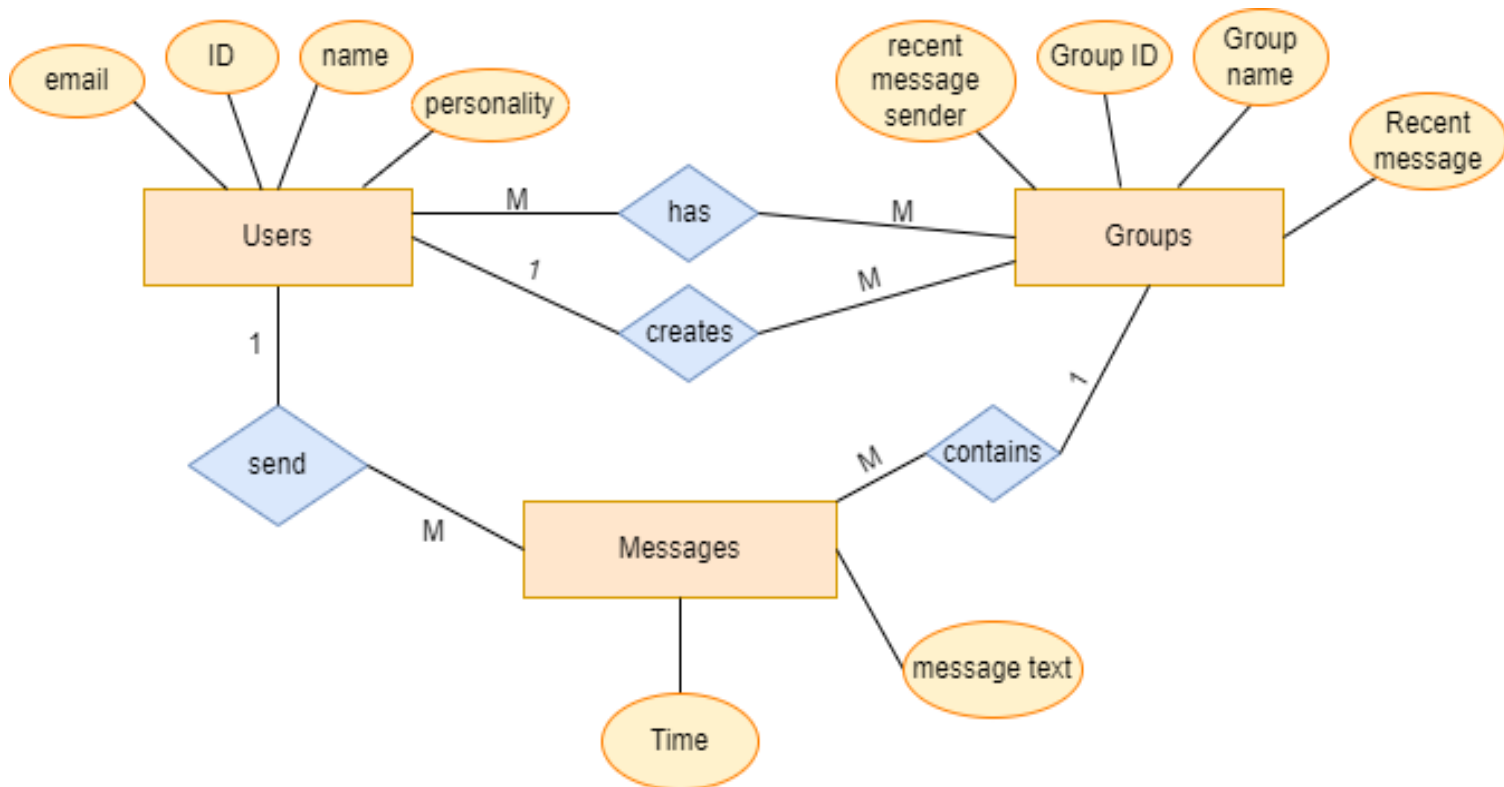
Technical problems:

- Lack of dataset used in ml model however we got the data after researching.*
- Flutter does not support backward compatibility and this lead to great efforts in learning Flutter and running the code.*
- Lack of information about how to connect ml model with Flutter because the library used in flutter (TFLite) was deprecated and most of tutorials are using this library however we make research to find another way to connect ml model.*
- The emulator used in VS code does not work because our laptops do not support it, but we managed to run on real device even it was very slow to build and run on real device as laptops do not have hard SSD to make the compilation faster.*
- Training ml model takes more time to train on data, but we used online python notebook other than local notebook to become faster in training.*

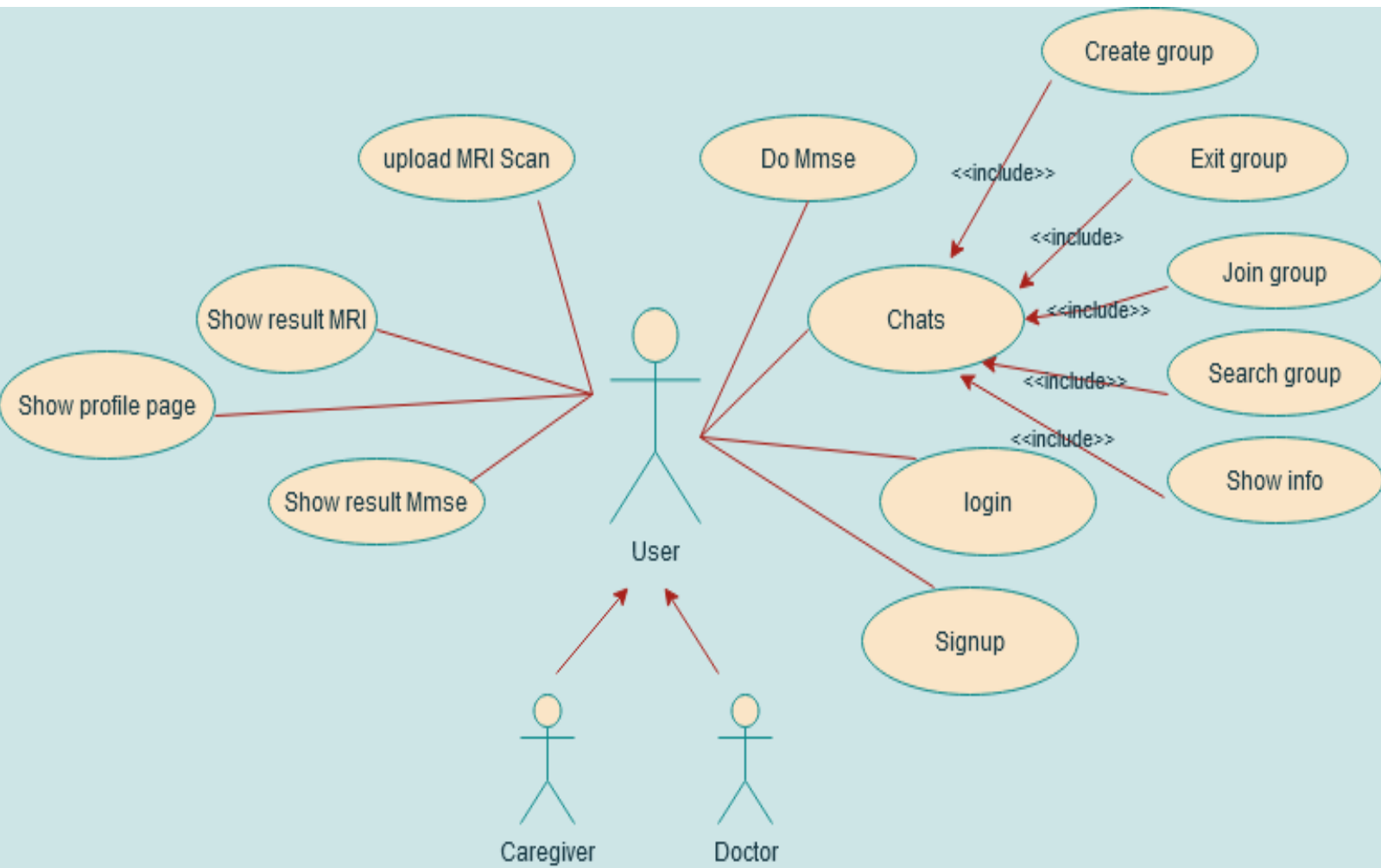
Chapter 3

Software Design

3.1 Design of Database (ERD or Class) Diagram



3.2 Use case Diagram



Use Case Scenario:

Login

| | |
|-----------------------|---|
| <i>initiator</i> | <i>Caregiver/Doctor</i> |
| <i>goals</i> | <i>Open his account on App</i> |
| <i>precondition</i> | <i>He already has an account and enter password and email correctly</i> |
| <i>Post condition</i> | <i>He can enter homepage</i> |
| <i>Main scenario</i> | <i>1-Enter password and email</i> <i>2- press login</i> <i>3-Enter homepage</i> |

Signup

| | |
|-----------------------|--|
| <i>initiator</i> | <i>Caregiver/Doctor</i> |
| <i>goals</i> | <i>Create an account</i> |
| <i>precondition</i> | <i>Enter personal data correctly</i> |
| <i>Post condition</i> | <i>He can create an account and enter home page</i> |
| <i>Main scenario</i> | <i>1-Enter personal data</i> <i>2-press signup</i> <i>3-Enter homepage</i> |

Show MRI result and upload MRI.

| | |
|-----------------------|---|
| <i>initiator</i> | <i>Caregiver/Doctor</i> |
| <i>goals</i> | <i>Known result of MRI scan (mild or very mild or moderate or normal)</i> |
| <i>precondition</i> | <i>He already has an account</i> |
| <i>Post condition</i> | <i>He can upload and know result of MRI scan</i> |
| <i>Main scenario</i> | <i>1-press on MRI scan button</i> <i>2- press Take a photo or pick from gallery and choose photo</i> |

Do MMSE and show the result MMSE.

| | |
|-----------------------|--|
| <i>initiator</i> | <i>Caregiver/Doctor</i> |
| <i>goals</i> | <i>Know result of questions of MMSE</i> |
| <i>precondition</i> | <i>He already has an account</i> |
| <i>Post condition</i> | <i>Know the result and score of MMSE question</i> |
| <i>Main scenario</i> | <i>1-press on MMSE button</i> <i>2- make patient answer all questions</i> |

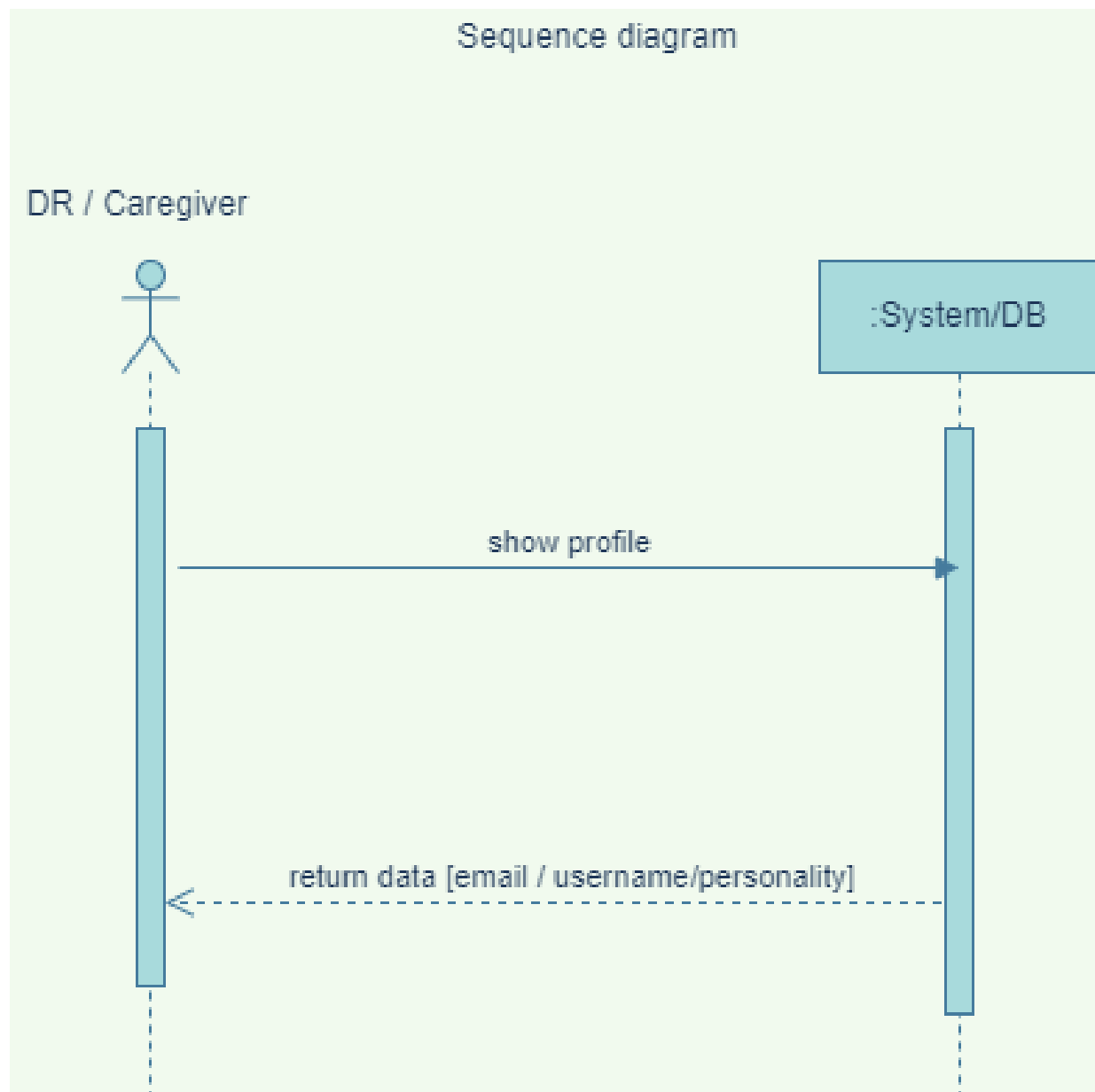
Show profile page.

| | |
|-----------------------|---|
| <i>initiator</i> | <i>Caregiver/Doctor</i> |
| <i>goals</i> | <i>Show profile page</i> |
| <i>precondition</i> | <i>He has an account</i> |
| <i>Post condition</i> | <i>He can enter to his profile page</i> |
| <i>Main scenario</i> | <i>1-Enter password and email correctly or create an account with correct data</i> <i>2- press on profile page</i> |

Chats

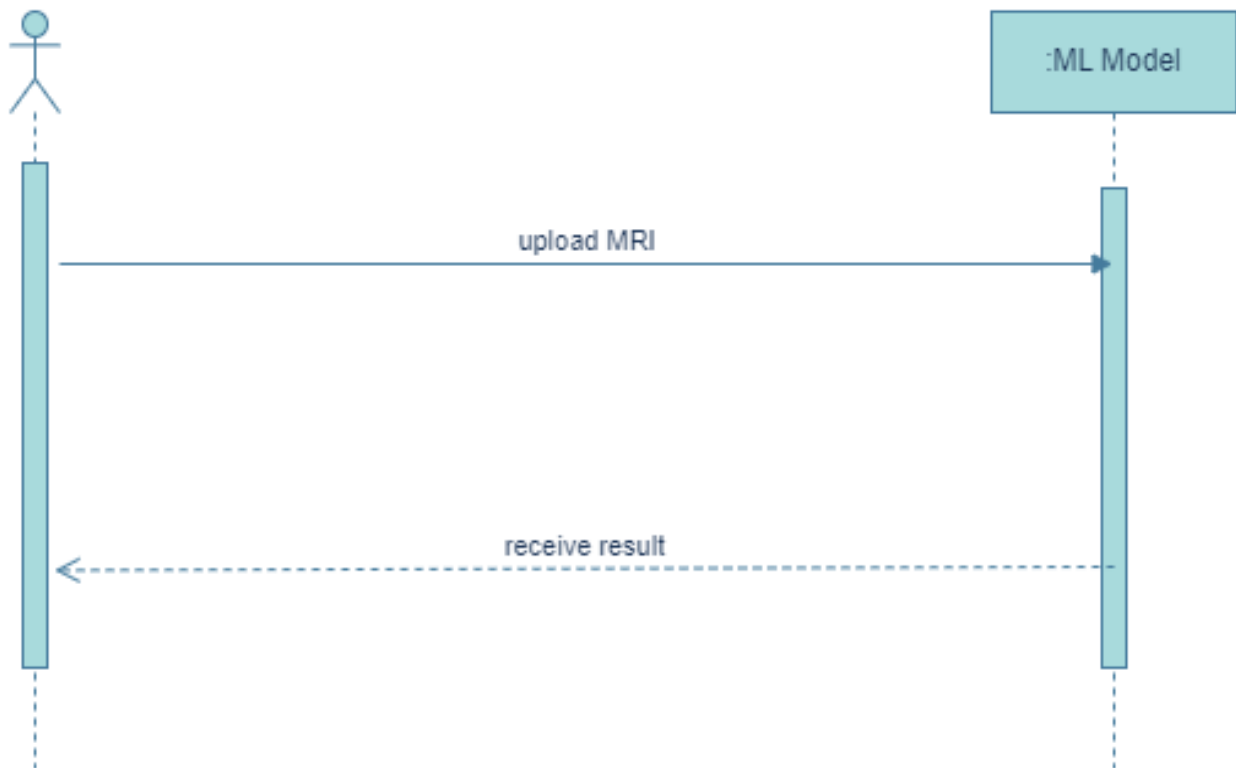
| | |
|-----------------------|---|
| <i>initiator</i> | <i>Caregiver/Doctor</i> |
| <i>goals</i> | <i>Caregiver can communicate with doctors</i> |
| <i>precondition</i> | <i>He has an account on App</i> |
| <i>Post condition</i> | <i>Caregiver can contact doctors</i> |
| <i>Main scenario</i> | <i>1-Caregivers/Doctors can create a group</i> <i>2- Caregivers/Doctors can search in list of chats</i> <i>3- Caregivers/Doctors can exit from group</i> <i>4- Caregivers/Doctors can join a group</i> <i>5- Caregivers/Doctors can show information of group</i> |

3.3 Sequence Diagram



Sequence diagram

Doctor / caregiver



Sequence diagram

Doctor / Caregiver

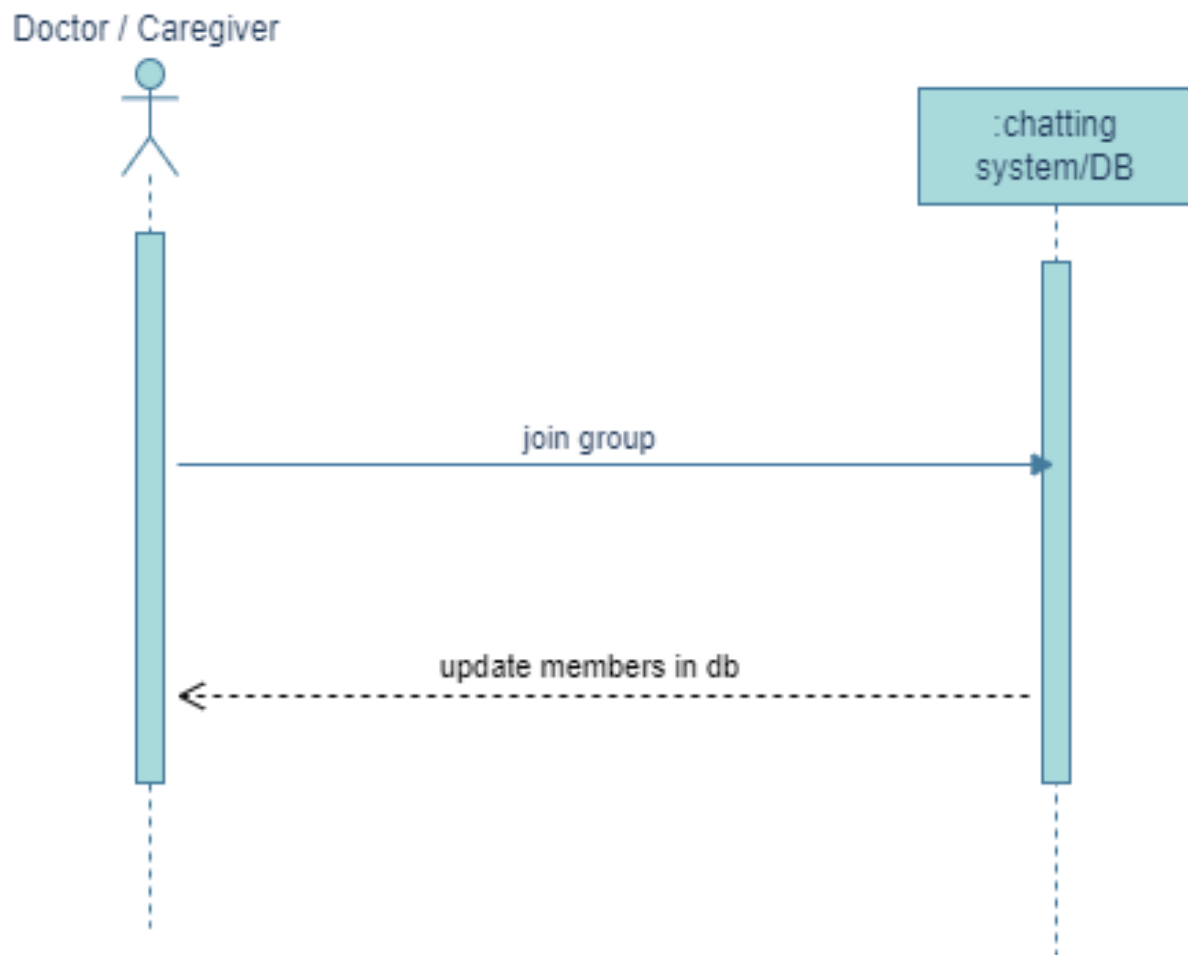


:chatting
system/DB

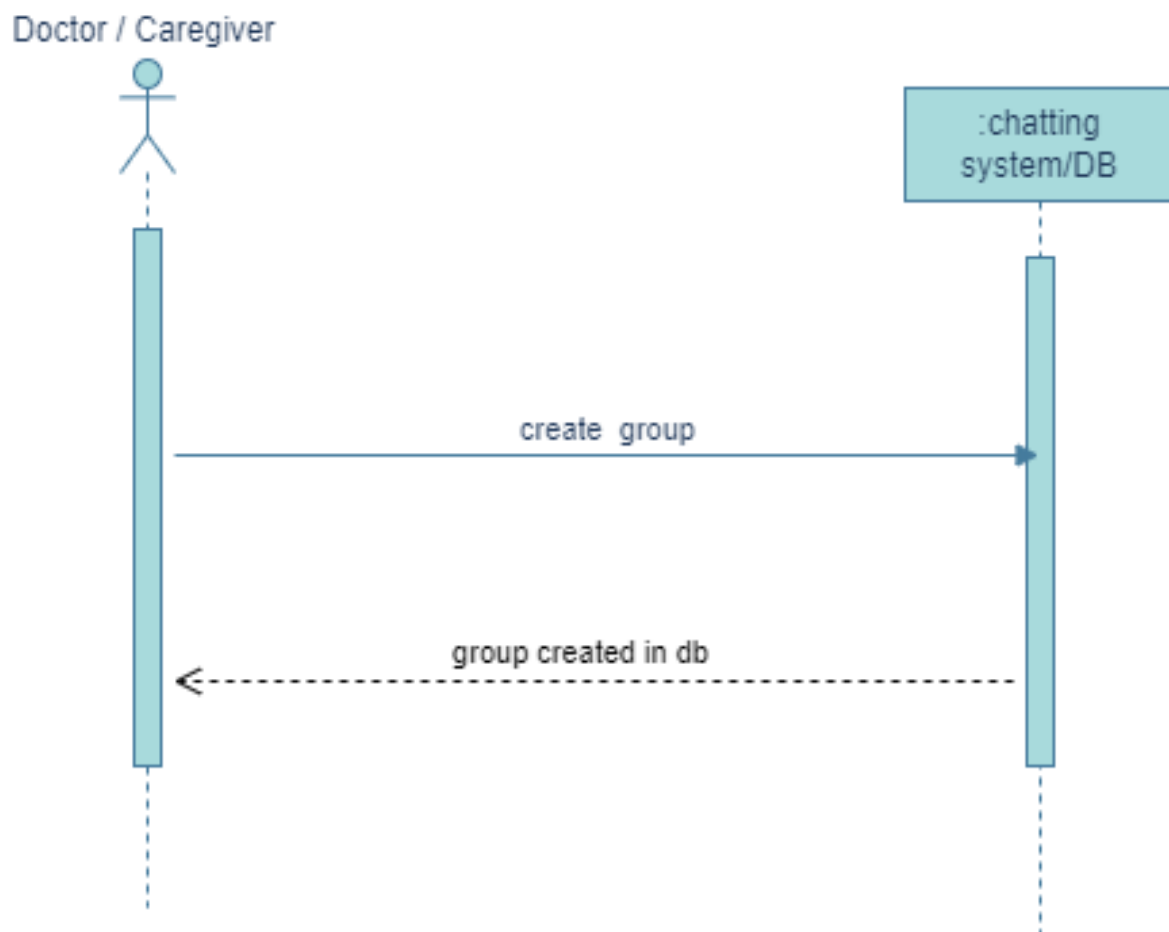
search group

retrieve groups

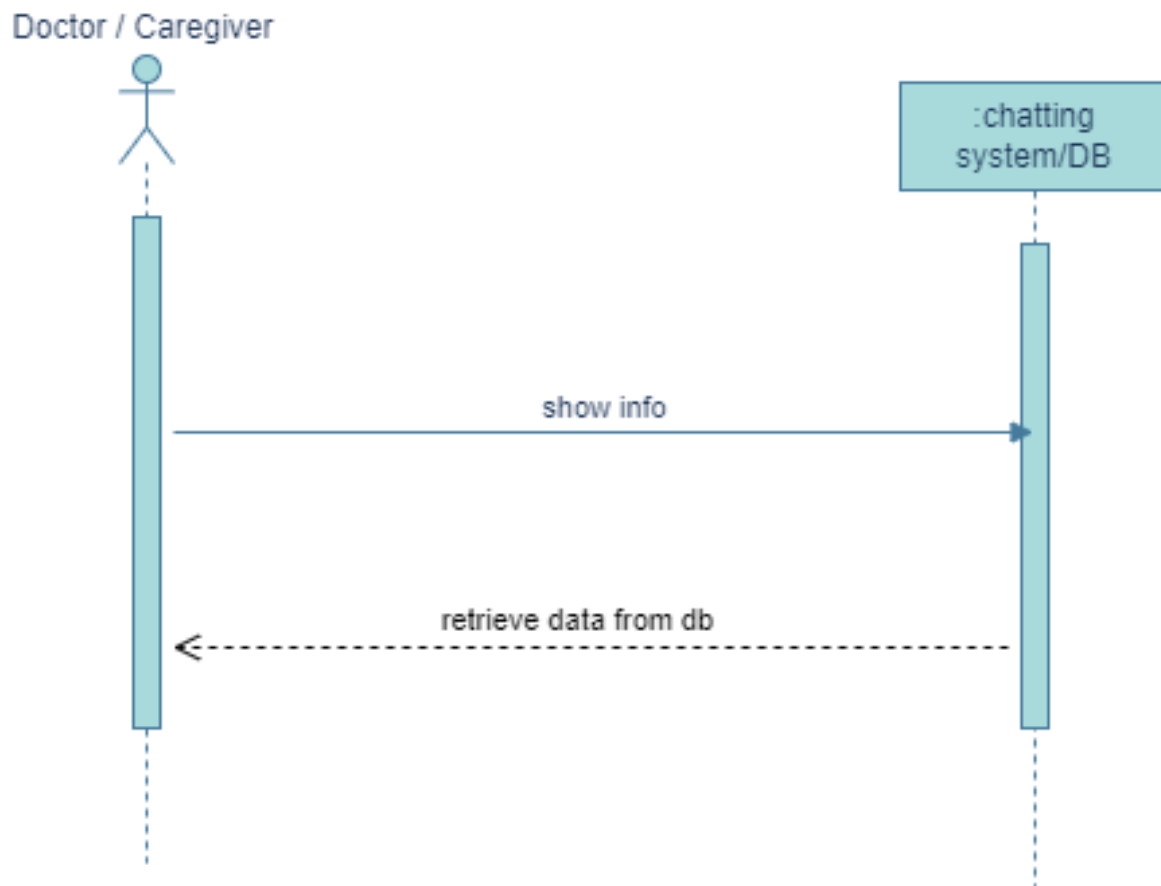
Sequence diagram



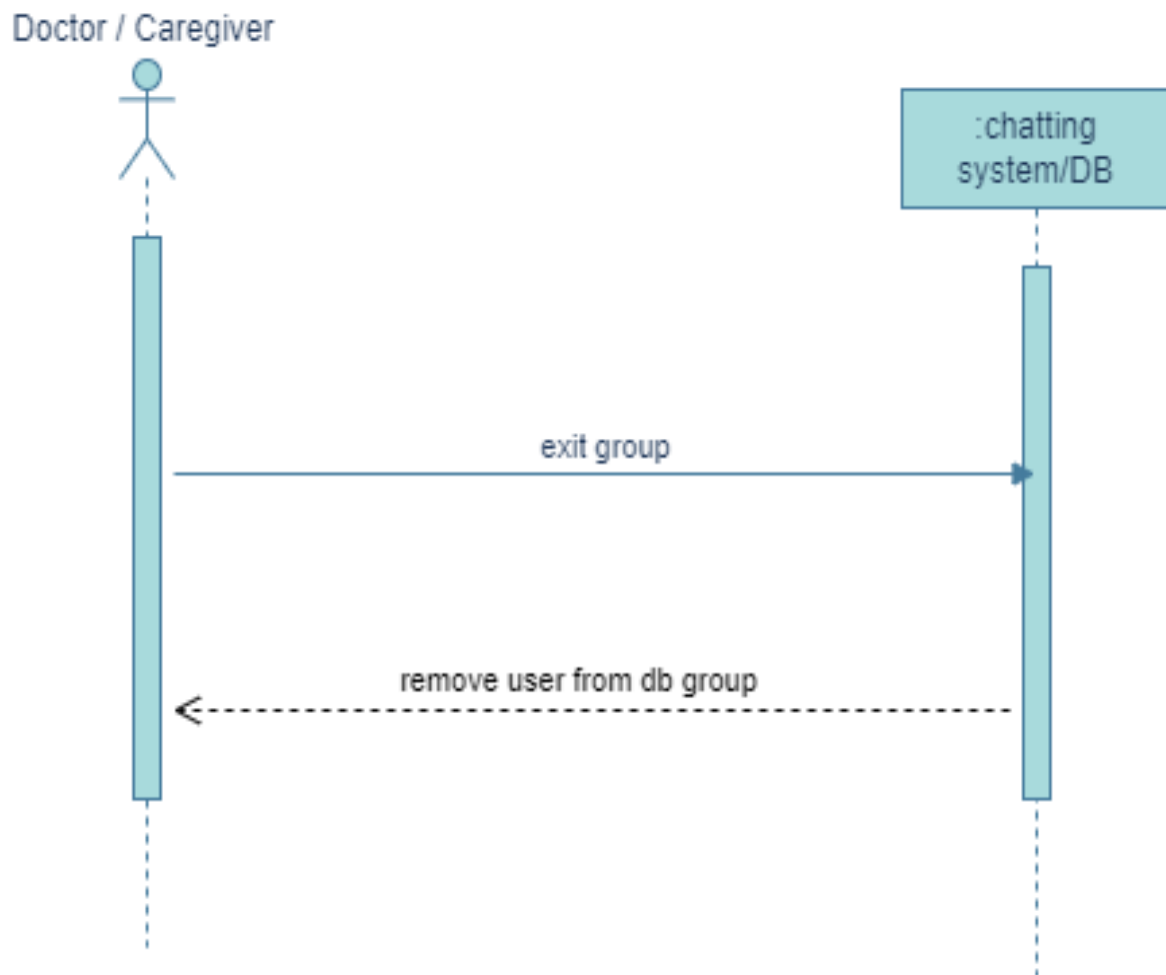
Sequence diagram



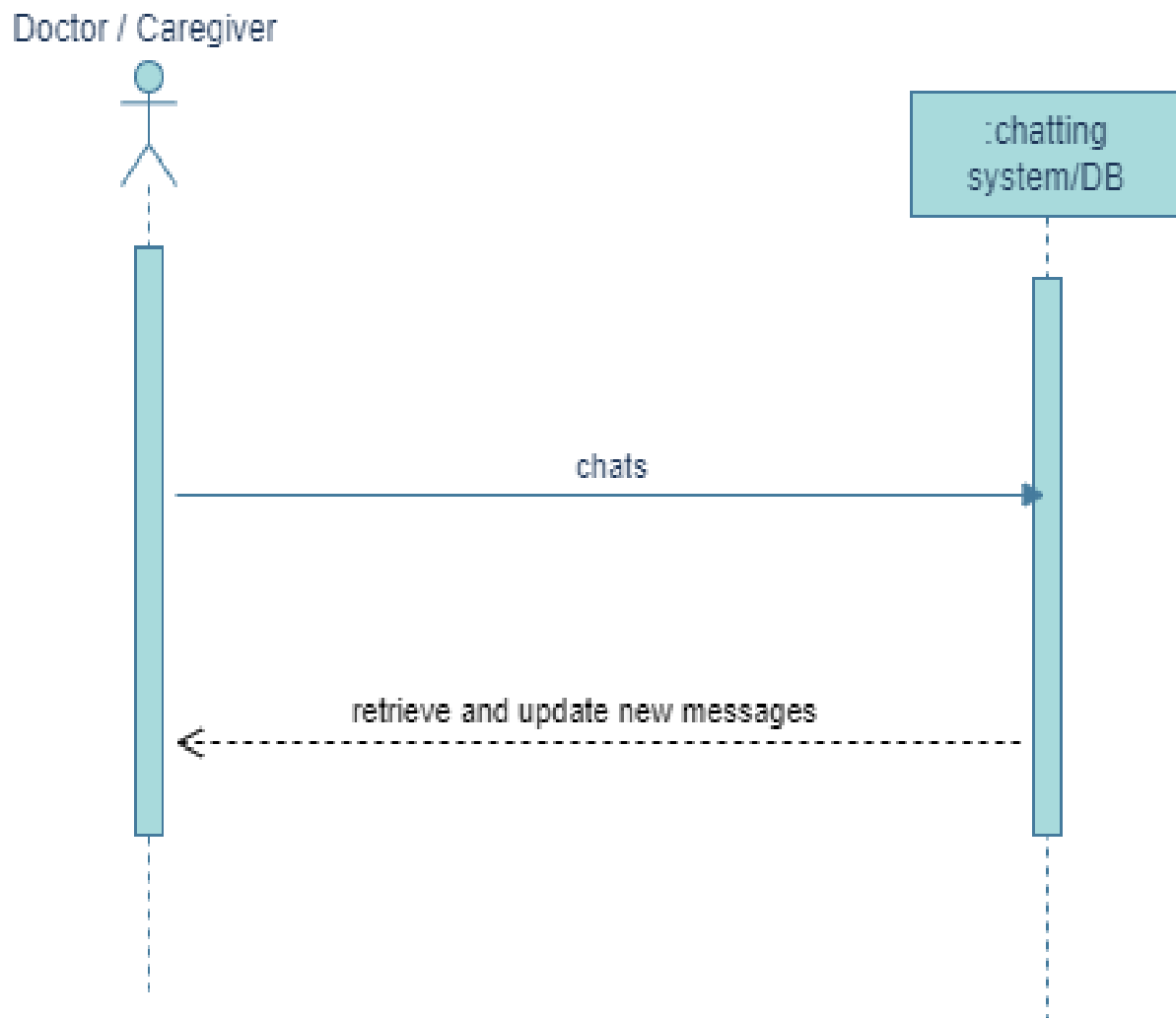
Sequence diagram



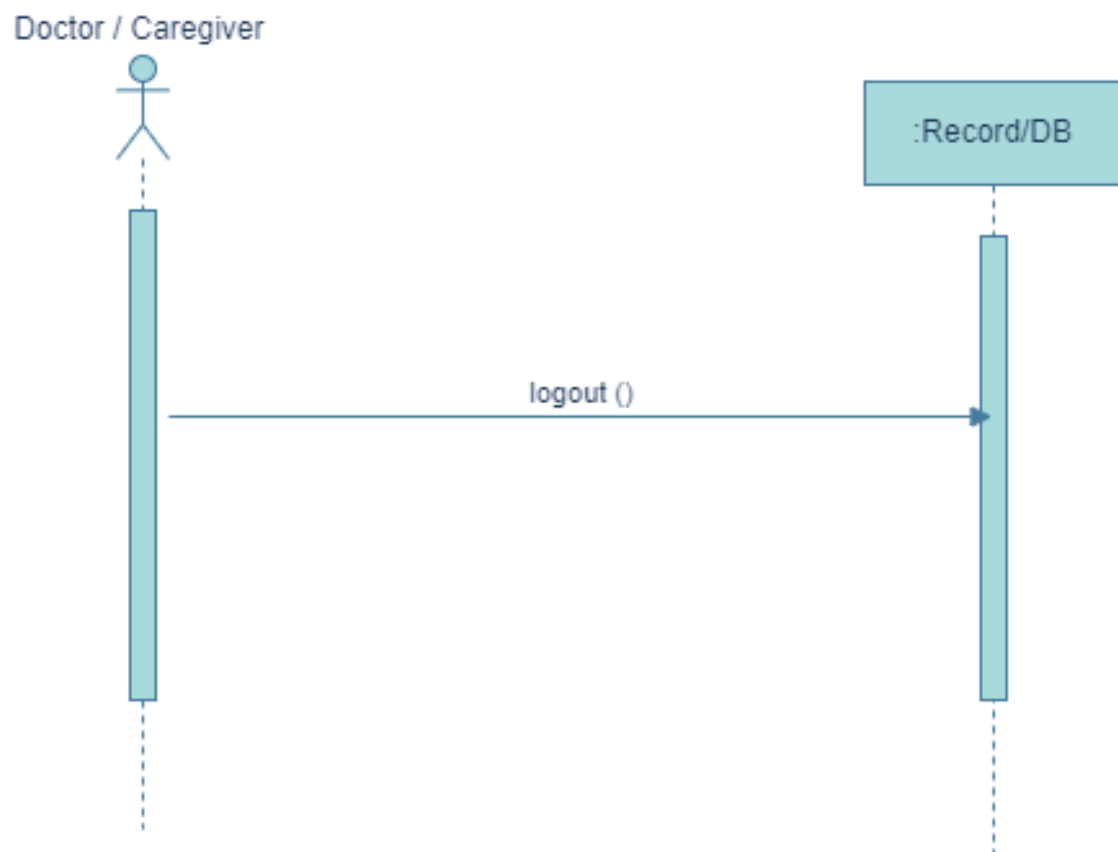
Sequence diagram



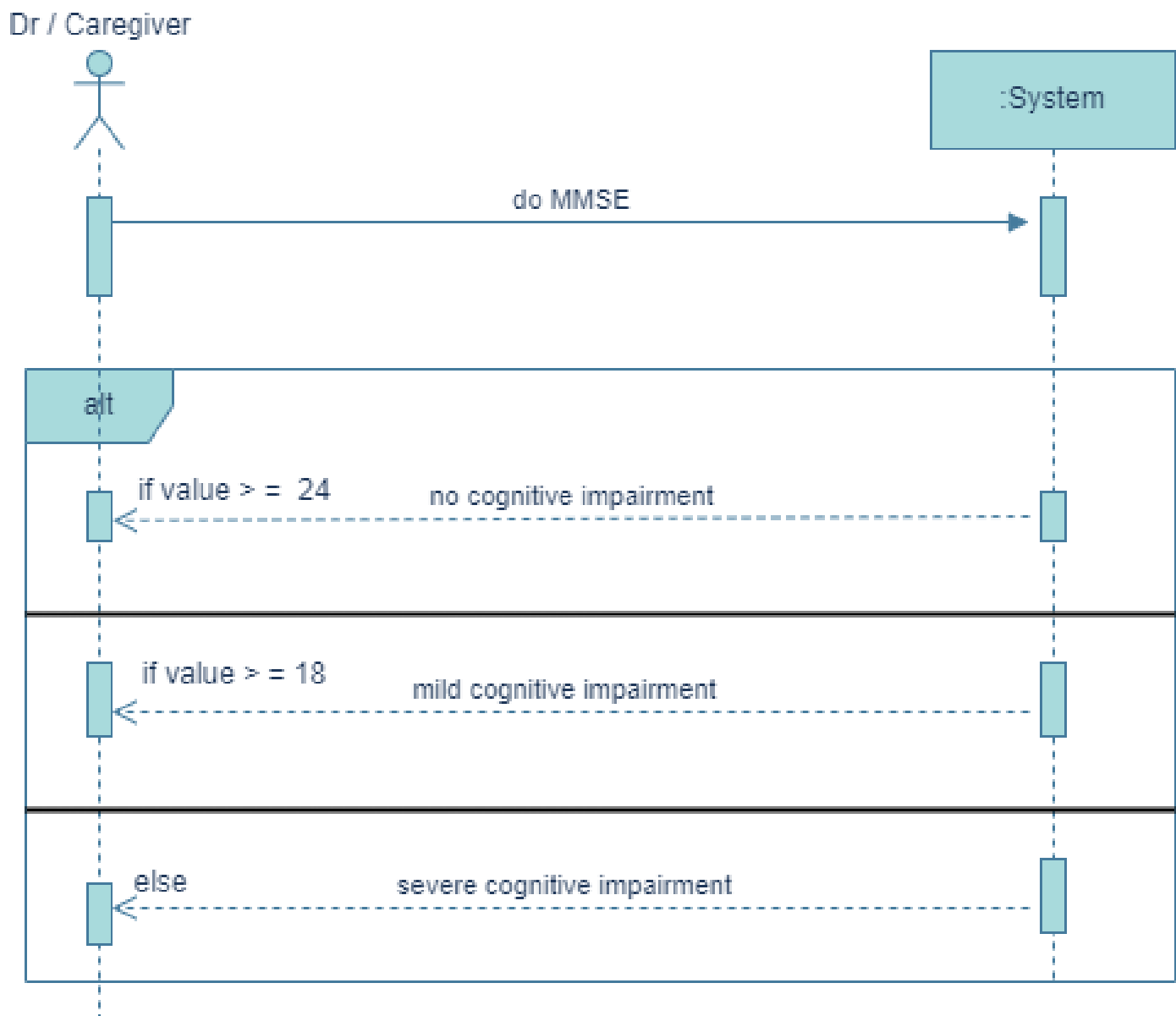
Sequence diagram



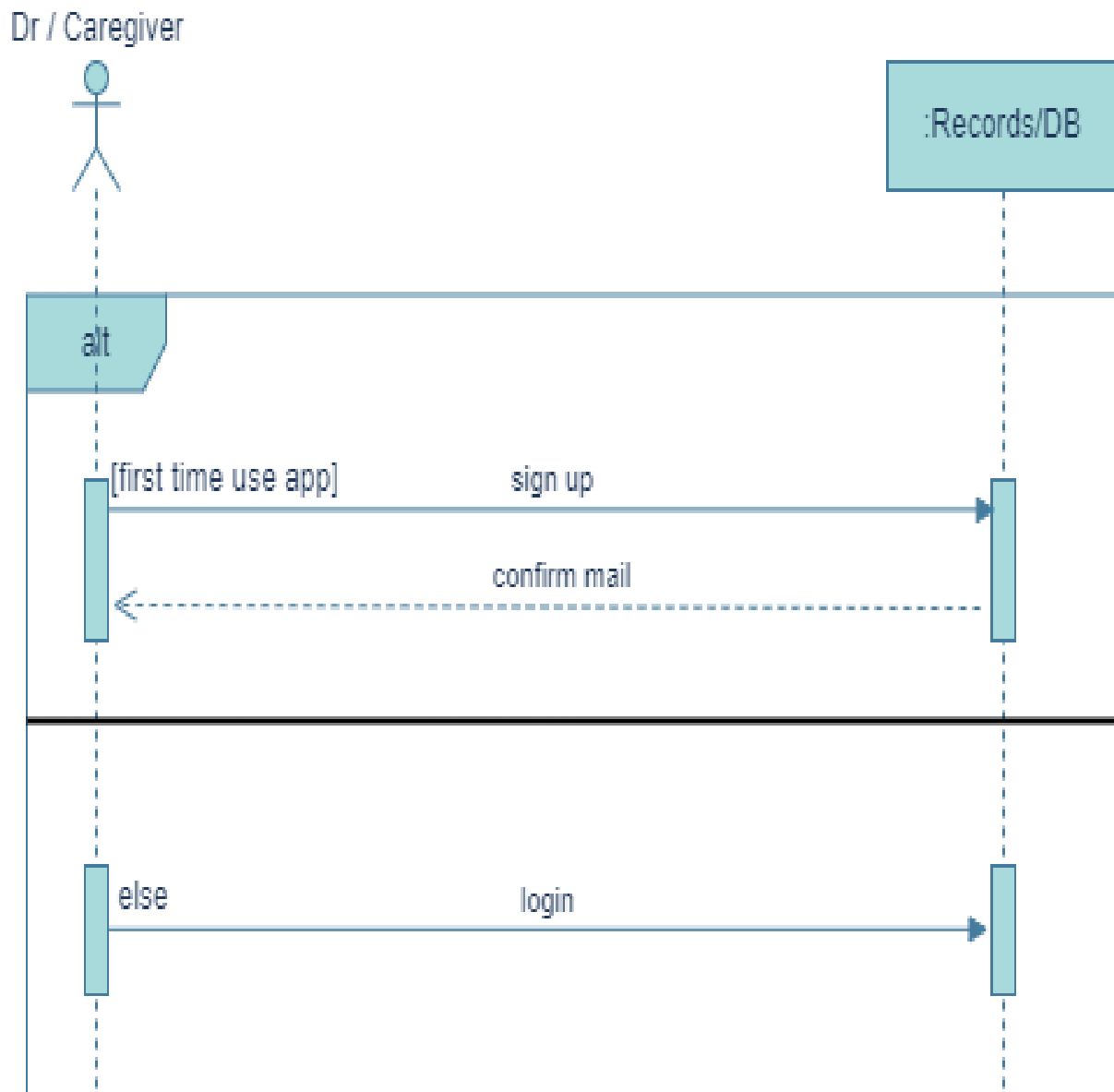
Sequence diagram



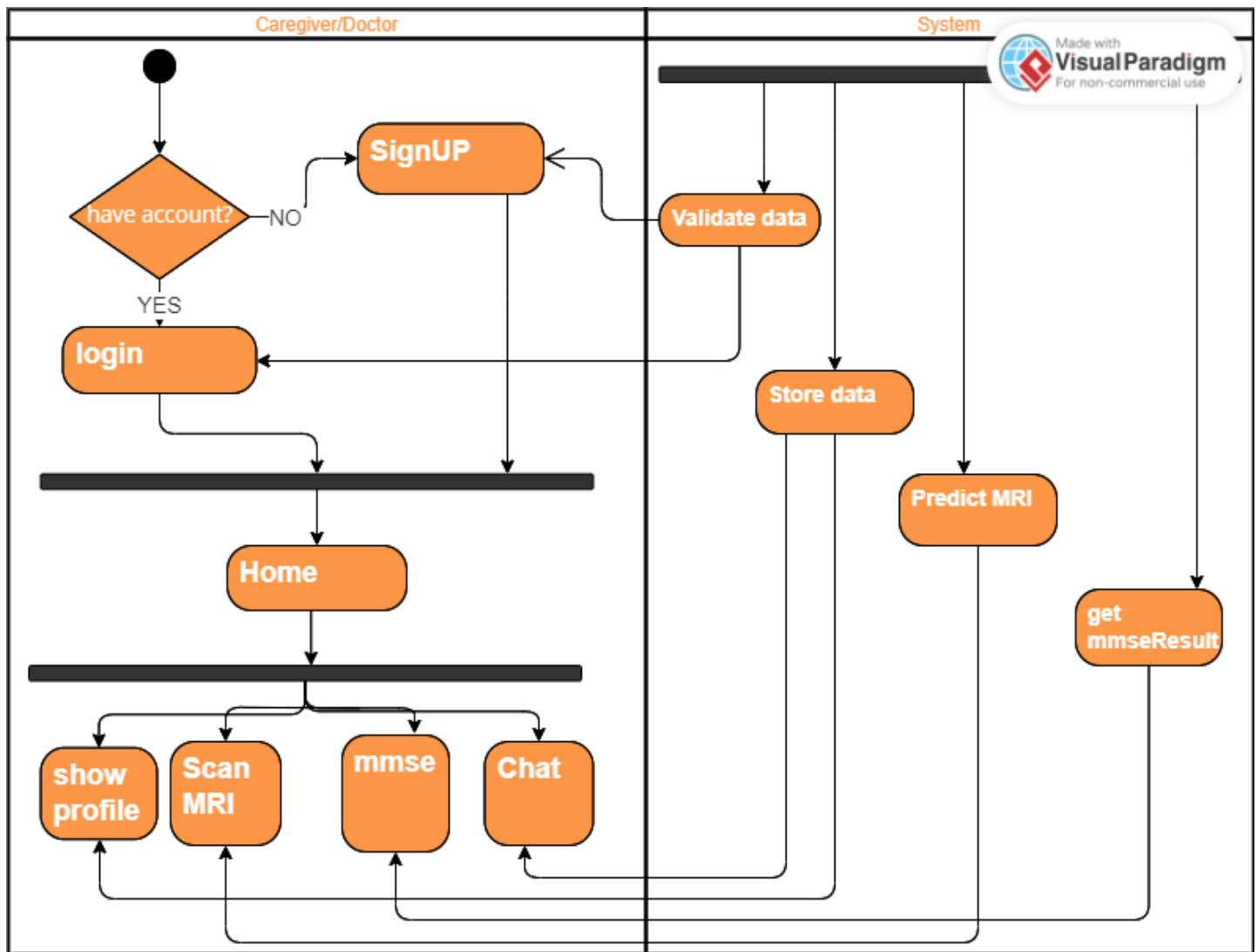
Sequence diagram



Sequence diagram



3.4 Activity Diagram



Chapter 4

Implementation

4.1 Software Architecture

4.1.1 Register

```
// register
Future registerUserWithEmailAndPassword(String fullName, String email,
    String password, String selcteditem) async {
  try {
    User user = (await firebaseAuth.createUserWithEmailAndPassword(
      email: email, password: password))
      .user!;

    if (user != null) {
      // call our database service to update the user data.
      await DatabaseService(uid: user.uid)
        .savingUserData(fullName, email, selcteditem);
      return true;
    }
  } on FirebaseAuthException catch (e) {
    return e.message;
  }
}
```

```
register() async {
  if (formKey.currentState!.validate()) {
    setState(() {
      _isLoading = true;
    });
    await authService
      .registerUserWithEmailAndPassword(
        fullName, email, password, selectedvalue!)
      .then((value) async {
        if (value == true) {
          // saving the shared preference state
          await HelperFunctions.saveUserLoggedInStatus(true);
          await HelperFunctions.saveUserEmailSF(email);
          await HelperFunctions.saveUserNameSF(fullName);
          await HelperFunctions.saveUserPersonalitySF(selectedvalue!);
          nextScreenReplace(context, const HomePage());
        } else {
          showSnackBar(context, Colors.red, value);
          setState(() {
            _isLoading = false;
          });
        }
      });
  }
}
```

4.1.2 login

```
// login
Future loginWithUserNameAndPassword(String email, String password) async {
  try {
    User user = (await firebaseAuth.signInWithEmailAndPassword(
      email: email, password: password))
      .user!;

    if (user != null) {
      return true;
    }
  } on FirebaseAuthException catch (e) {
    return e.message;
  }
}
```

```
login() async {
  if (formKey.currentState!.validate()) {
    setState(() {
      _isLoading = true;
    });
    await authService
      .loginWithUserNameAndPassword(email, password)
      .then((value) async {
        if (value == true) {
          QuerySnapshot snapshot =
            await DatabaseService(uid: FirebaseAuth.instance.currentUser!.uid)
              .gettingUserData(email);
          // saving the values to our shared preferences
          await HelperFunctions.saveUserLoggedInStatus(true);
          await HelperFunctions.saveUserEmailSF(email);
          await HelperFunctions.saveUserNameSF(snapshot.docs[0]['fullName']);

          await HelperFunctions.saveUserPersonalitySF(
            snapshot.docs[0]['personality']);

          nextScreenReplace(context, const HomePage());
        } else {
          showSnackBar(context, Colors.red, value);
          setState(() {
            _isLoading = false;
          });
        }
      });
  }
}
```

4.1.3 *logout*

```
// signout
Future signOut() async {
  try {
    await HelperFunctions.saveUserLoggedInStatus(false);
    await HelperFunctions.saveUserEmailSF("");
    await HelperFunctions.saveUserNameSF("");
    await firebaseAuth.signOut();
  } catch (e) {
    return null;
  }
}
```


4.1.4 MMSE result

```
String get resultPhrase {  
    String resultText = '';  
    if (widget.resultScore >= 24) {  
        resultText = 'No cognitive impairment';  
    } else if (widget.resultScore >= 18) {  
        resultText = 'Mild cognitive impairment';  
    } else {  
        resultText = 'Severe cognitive impairment';  
    }  
  
    return resultText;  
}
```

4.1.5 *pick image to be loaded in ml model.*

```
void _onPickPhoto(ImageSource source) async {  
    final pickedFile = await picker.pickImage(source: source);  
  
    if (pickedFile == null) {  
        return;  
    }  
  
    final imageFile = File(pickedFile.path);  
    setState(() {  
        _selectedImageFile = imageFile;  
    });  
  
    _analyzeImage(imageFile);  
}
```

4.1.6 *load classifier model*

```
Future<void> _loadClassifier() async {  
  debugPrint(  
    'Start loading of Classifier with '  
    'labels at $_labelsFileName, '  
    'model at $_modelFileName',  
  );  
  
  final classifier = await Classifier.loadWith(  
    labelsFileName: _labelsFileName,  
    modelFileName: _modelFileName,  
  );  
  _classifier = classifier;  
}
```

4.1.7 Analyze image.

```
void _analyzeImage(File image) {
    _setAnalyzing(true);

    final imageInput = img.decodeImage(image.readAsBytesSync());

    final resultCategory = _classifier!.predict(imageInput);

    final result = resultCategory.score >= 0.8
        ? _ResultStatus.found
        : _ResultStatus.notFound;
    final MRILabel = resultCategory.label;
    final accuracy = resultCategory.score;

    _setAnalyzing(false);

    setState(() {
        _resultStatus = result;
        _MRILabel = MRILabel;
        _accuracy = accuracy;
    });
}
```

4.1.8 *send messages.*

```
// send message
sendMessage(String groupId, Map<String, dynamic> chatMessageData) async {
  groupCollection.doc(groupId).collection("messages").add(chatMessageData);
  groupCollection.doc(groupId).update({
    "recentMessage": chatMessageData['message'],
    "recentMessageSender": chatMessageData['sender'],
    "recentMessageTime": chatMessageData['time'].toString(),
  });
}
```

4.1.9 *get chats.*

```
getChats(String groupId) async {
  return groupCollection
    .doc(groupId)
    .collection("messages")
    .orderBy("time")
    .snapshots();
}
```

Chapter 5

Testing

Software Testing:

Is a method to check whether the actual software product matches expected requirements and to ensure that software product is defect free.

It involves execution of software/system components using manual or automated tools to evaluate one or more properties of interest. The purpose of software testing is to identify errors, gaps, or missing requirements in contrast to actual requirements.

Functional Testing:

5.1- Unit testing:

Is a type of software testing where individual units or components of a software are tested. The purpose is to validate that each unit of the software code performs as expected. Unit Testing is done during the development (coding phase) of an application by the developers. Unit Tests isolate a section of code and verify its correctness. A unit may be an individual function, method, procedure, module, or object.

Unit testing is the first level of testing done before integration testing.

5.2- Integration Testing:

Is defined as a type of testing where software modules are integrated logically and tested as a group. A typical software project consists of multiple software modules, coded by different programmers. The purpose of this level of testing is to expose defects in the interaction between these software modules when they are integrated.

System testing:

In this method, your software is compiled as a whole and then tested as a whole. This testing strategy checks the functionality, security, portability, amongst others.

Non-Functional Testing:

non-functional parameters such as stress testing, load testing, recovery testing, volume, security, accountability

Types:

1. Performance Testing:

examines the functionality of software components. These tests uncover flaws in software architecture and design.

2. Usability Testing:

Non-functional testing includes usability testing as one of the testing methods. Usability testing is all about ensuring that the software is simple to use for consumers.

3. Compatibility Testing:

Customer satisfaction is ensured by compatibility testing, which is also non-functional testing. Its purpose is to see if your software program or product can run in a variety of browsers, databases, hardware, operating systems, mobile devices, and networks.

5.3 Additional testing:

Registration

Test case scenario:

try to register with valid and invalid data.



| <u>Test case</u> | <u>Test steps</u> | <u>Test data</u> | <u>Result</u> |
|---------------------------------------|---|--|---|
| Register with complete and valid data | 1-press sign up 2-enter data 3-press register | Personal data(name –email- password-country) | 1-account created Successfully And enter to home screen |



| Test case | Test steps | Test data | Result |
|--|---|---|--|
| Register with incomplete or invalid data | 1-press sign up 2-enter data 3-press register | Personal data(name –email-password-country) | 1-display error message to complete empty fields |

Emergency calls only 7:13 AM

Full Name

Name cannot be empty

Email

Please enter a valid email

Password

Password must be at least 6 characters

Select your personality

Register

Login

Test case scenario: try to login with valid or invalid email and password.

Emergency calls only 7:28 AM

AlzheimerCare

Email

Password

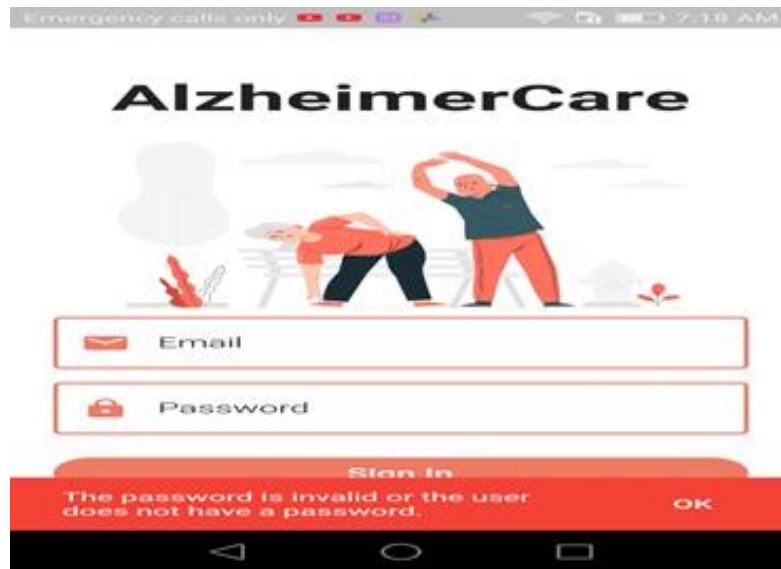
Sign In

Don't have an account? [Register here](#)

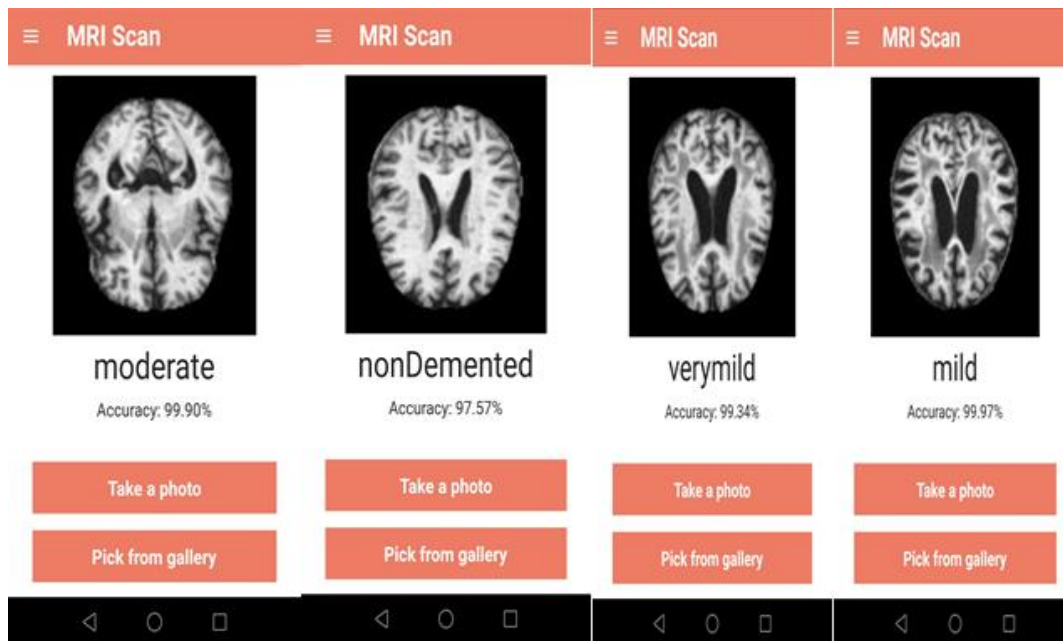
| <u>Test case</u> | <u>Test steps</u> | <u>Test data</u> | <u>Result</u> |
|---|--|-------------------------------|--|
| login with valid email and valid password | 1-enter email 2-enter password 3-press login | Valid email Valid password | 1-logged successfully and enter to home screen |



| <u>Test case</u> | <u>Test steps</u> | <u>Test data</u> | <u>Result</u> |
|--|--|--|---|
| login with invalid email or invalid password | 1-Enter email 2-Enter password 3-press Login | Valid email And invalid password or invalid email And valid password Or invalid password and email | 1-Error message Password invalid or email does not exit |



MRI:



MRI Scan can diagnosis if patient have Alzheimer's disease or not and which level of Alzheimer's disease the patient has and it has high accuracy.

MMSE

| MMSE | MMSE |
|-------------------|-----------------|
| what is the month | Day of the week |
| True | True |
| False | False |

A Mini-Mental State Examination (MMSE) is a set of questions that doctors and caregivers commonly use to check for cognitive impairment (problems with thinking, communication, understanding and memory).

Chapter 6

Results & Discussion

6.1 Result

6.1.1 Expected Results

- *The application is expected to improve the quality of care and support for patients with Alzheimer's disease and their caregivers.*
- *The application is expected to accurately diagnose Alzheimer's disease and determine its stage using image processing algorithms and machine learning algorithms.*
- *The MMSE test is expected to help accurately diagnose Alzheimer's disease and determine its stage.*
- *The application is expected to improve communication between caregivers and doctors and facilitate secure information sharing.*

6.1.2 Actual Results

- *The actual results of the application were evaluated by comparing them to the expected results.*
- *The performance of the application was evaluated by monitoring its compliance with the specified requirements and the level of satisfaction among users.*
- *The accuracy of diagnosing Alzheimer's disease and determining its stage using the application was evaluated by comparing it to the standard medical diagnosis.*
- *Any problems or errors in the performance of the application were analyzed and attempts were made to correct them.*

The only problem with the application was that it did not have an instant response and needed a few seconds to evaluate accurately and precisely.

- *The ease of use and navigation of the application was also evaluated.*

In the end, the actual results & the expected results are not so different from each other.

6.2 Discussion

As mentioned, the actual results of the application were evaluated by comparing them to the expected results and identifying any differences between them. Any problems or errors in the application's performance were also analyzed and attempts were made to correct them. We encountered some challenges during the implementation, such as:

- Gathering a large amount of MRI scan data for analysis and comparing the accuracy of the analysis results to the actual MRI scan results.*
- Selecting the appropriate machine learning algorithms that provide the highest accuracy.*

In addition, we need to discuss any potential improvements to the application in the future, as well as how to develop and improve its performance. We have decided to add some features soon, such as:

- Adding a recommendation system to the application that recommends various activities for the patient to improve their concentration level.*
- Adding a tracking feature for the patient's location to ensure their safety.*
- Adding effective features for doctors, such as the ability for each doctor to add their own patients for continuous monitoring of their condition.*

Furthermore, we aim to improve the performance of the application.

Chapter 7

Conclusion

An Alzheimer's care project with Flutter that includes the use of MMSE tests, MRI scans with machine learning algorithms, and a chat feature with doctors has the potential to significantly improve the quality of care for Alzheimer's patients and their caregivers.

By leveraging Flutter's advanced customization options and machine learning capabilities, developers can create a mobile application that accurately diagnoses Alzheimer's disease using MMSE tests and MRI scans.

Machine learning algorithms can analyze patient data, such as brain scans and cognitive test results, to provide accurate and personalized diagnoses.

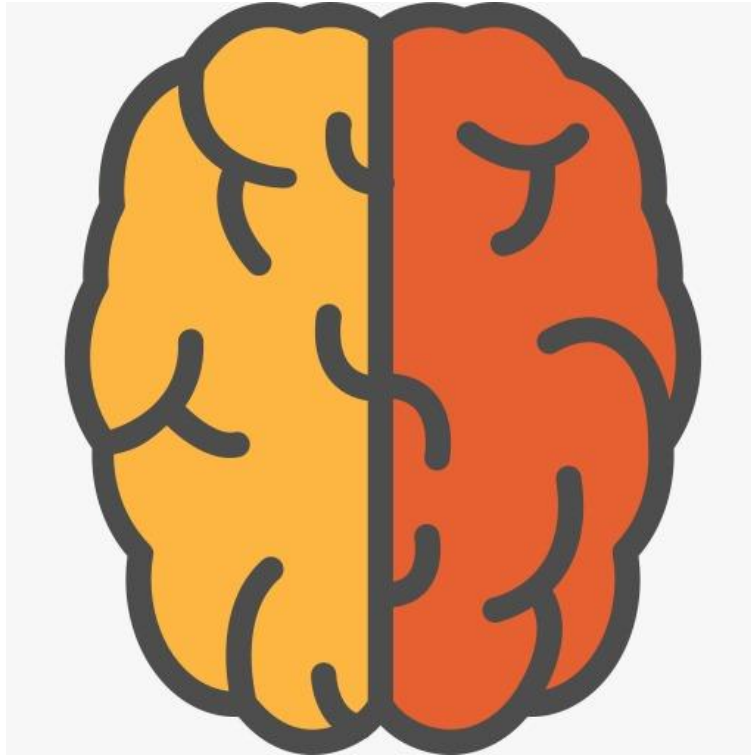
The chat feature with doctors can help patients and caregivers communicate with medical professionals and receive guidance on care plans. This can be particularly helpful for patients who may have difficulty visiting doctors in person.

Overall, this Alzheimer's care project with Flutter has the potential to improve the diagnosis and treatment of Alzheimer's disease and provide valuable support for patients and caregivers. By leveraging the platform's advanced features and machine learning capabilities, developers can create a mobile application that is tailored to the specific needs of each patient, helping to improve patient outcomes and quality of life.

Chapter 8

Future Work

- 1. We plan to put recommendation system in our application that the doctor can recommend the activities and the drugs that suits the patient.*
- 2. We plan to track the patient's place in case he goes out alone and can't come back again.*
- 3. We plan to put reservation system that the caregiver can reserve an appointment with the doctor.*
- 4. We plan to put feedback and rating the doctor in our application so that the caregiver can rate the doctor with the positive or negative opinion.*



AI-ZHIEMER CARE

THANK YOU

