LAMI Full-Stack Application

Project Solution Approach

LAMI

Logo

Description automatically generated

**LAMI Team**

Jacob Smith and Jerrel Chapmen

10/4/2022

**TABLE OF CONTENTS**

[I. Introduction 2](#_Toc82521031)

[II. System Overview 2](#_Toc82521032)

[III. Architecture Design 2](#_Toc82521033)

[III.1. Overview 2](#_Toc82521034)

[III.2. Subsystem Decomposition 2](#_Toc82521035)

[I.1.1. [Subsystem Name] 3](#_Toc82521036)

[a) Description 3](#_Toc82521037)

[b) Concepts and Algorithms Generated 3](#_Toc82521038)

[c) Interface Description 3](#_Toc82521039)

[I.1.2. [Include sections III.2, III.3, etc., for other subsystems] 4](#_Toc82521040)

[IV. Data design 5](#_Toc82521041)

[V. User Interface Design 5](#_Toc82521042)

[VI. Glossary 6](#_Toc82521043)

[VII. References 6](#_Toc82521044)

[VIII. Appendices 6](#_Toc82521045)

# Introduction

This document provides details on the structural and technological decisions made in the development of the application. The application serves as a medication reminder for residents of the Alliance House, all of whom are people with schizophrenia. Many people with schizophrenia struggle to adhere to their prescribed medications and the goal of this application is to address and work towards providing a solution to that problem.

# System Overview

The LAMI project is to create and add on to the already existing code and improve upon it, also to implement a website that can access the same things as the app. The coding will be done using the flutter language since it can be used to program apps and websites. Also, the database we will be using to keep track of profiles and personal info will be firebase. With these the users will be able to communicate through a server to add medications, sleep times for tracking when to take medicine, and surveys so the caretakers can check the medications effectiveness.

# Architecture Design

## Overview

The architectural design of the existing application is structured in accordance with the *Client*-*Server* network architecture. With regards to this application, the *Server* refers to a central database and the *Client* refers to two distinct groups, the employees of LAMI and the residents of The Alliance House. Both groups interact with the same server, albeit in slightly different ways. When broken down to its simplest form, the employee group can view and modify any entry in the database through a web or mobile application, whereas the resident group can only view and modify entries associated with their account through the mobile application.

## Subsystem Decomposition

In the application, the *Client* subsystem is implemented with the Flutter application framework and the *Server* subsystem is implemented with Google Firebase. These two technological choices were made because both Flutter and Firebase are two very popular application development tools and are often used in conjunction. Since they are so widely used, they are constantly being updated and improved. In addition, there is an abundance of online documentation and resources for application development with these subsystems.

Diagram

Description automatically generated

Frontend

Backend

### [Flutter Multi-Platform Application Framework]

#### Description

#### The Flutter framework provides mechanisms to create user interfaces for both mobile and web applications. A consistent framework for the development of all three UI’s (iOS, Android, Web app) increases readability, cohesion, and structure throughout the entire project. One of the most important services Flutter provides is the ability to communicate with the operating system of the phone to send a push notification outside of the application.

#### Concepts and Algorithms Generated

Flutter makes use of a concept called *widgets* to form and manage each component of an application. Essentially, each widget represents a different aspect of the app, whether it be a UI element, styling choice, or object state.

#### Interface Description

#### Services Provided:

#### Service name: Alarm Widget

*Service provided to:* Flutter

*Description:* The service allows users to add a medication to a list and set a reminder for when the medication should be taken. It takes user input (when a user adds a new alarm/medication to the list) then contacts the backend to save the new information to the users full list of medications.

#### Service name: FlutterFire

*Service provided to:* Firebase

*Description:* FlutterFire is a group of plug-ins for Flutter that provide connectivity between Flutter applications and Google Firebase backend services. The specific functionalities utilized from Firebase are detailed later.

*Services Required:*

Flutter, Firebase, FlutterFire

### [Google Firebase]

#### Description

Google Firebase is a backend application development software that provides storage capabilities. To integrate Firebase with Flutter applications, plugins called FlutterFire have been created to easily accomplish this task. In addition to storage, Firebase also provides other services such as authentication, usage analytics, and monitoring tools.

#### Concepts and Algorithms Generated

#### Cloud Firestore is a NoSQL document database. A document database is different than a relational database in that all information about an object (in this case a resident of the Alliance house) is stored in its own document.

#### Interface Description

#### Services Provided:

#### Service name: Firebase Cloud Firestore

*Service provided to:* Flutter

*Description:* Firebase Cloud Firestore provides a database for the storage of user information, medications, and medication reminders. On login, a user’s locally stored data is synced with the data from Cloud Firestore, and any necessary changes are applied.

#### Service name: Firebase Authentication

*Service provided to:* Flutter

*Description:* Firebase Authentication allows the addition, authentication, and deletion of users from the application. Users can also choose to reset their password and Firebase Authentication provides the backend services, allowing a user to reset a password through email.

*Services Required:*

Flutter, Firebase Authentication, Cloud Firestore, FlutterFire

# Data design

As stated previously we will be using firebase to keep track of user profiles and information since the last team to work on the application used firebase. In addition to firebase already being implemented into the code it is an easy-to-use database storage that will be easy to insert new users if there ever happen to be new people added to the LAMI house.

# User Interface Design

Graphical user interface, application

Description automatically generated Table

Description automatically generated Table

Description automatically generated

In the above screen shots are rough looks at what the LAMI application will look like; at least the login and beginning pages for both the resident and caretaker. For the resident it is the middle picture where it will display the calendar for the week of what they must take and below that, it shows them their daily schedule for medicine. Then for the caretaker screen on the right it shows all the patients accounts and below that it shows the weekly schedule with all the patients appointed medication times. For both those screens there is a menu button on the top left which will show setting and for the residents will have a medication tab where they can change times it needs to be taken and change medication. For the caretaker it will show medications used in the house and who is using that medication. Now this is for the application and the website, however the only difference there will be is that the calendar will be bigger to plan out the month of medication.

# Glossary

# References

# VIII. Appendices