LAMI Full-Stack APP

Project Alpha Prototype Report

Logo

Description automatically generated

**LAMI Team**

Jacob Smith and Jerrel Chapman

11/15/2022

**Assignment Notes**

Length = minimum of 5 pages text + appendices as needed - though, this should be \*MUCH\* longer than 5 pages if you leverage all of your prior documents  
  
Sections that do not count to content for page limit:

* Cover page
* table of contents
* pictures
* tables
* images
* diagrams

Posted as a single self‐contained file (no links to outside resources.)

Posted as a PDF file.

Typed single‐spaced.

Typed with black text.

Typed with #11 font size.

Typed using Arial font or similar font.

Typed with one inch margins on sides, top and bottom.

**Please erase this page in your final document.**

**TABLE OF CONTENTS**

[I. Introduction 6](#_Toc119431159)

[II. Team Members - Bios and Project Roles 6](#_Toc119431160)

[III. Project Requirements 6](#_Toc119431161)

[I. Introduction 6](#_Toc119431162)

[II. System Requirements Specification 6](#_Toc119431163)

[II.1. Use Cases 7](#_Toc119431164)

[II.2. Functional Requirements 7](#_Toc119431165)

[II.2.1. Improvement of the UI 7](#_Toc119431166)

[II.2.2. Website Implementation 7](#_Toc119431167)

[II.2.3. Resident and Caretaker Accounts 8](#_Toc119431168)

[II.2.4. Resident Surveys 8](#_Toc119431169)

[II.3. Non-Functional Requirements 8](#_Toc119431170)

[II.3.1. Must be able to support around 10-15 users 8](#_Toc119431171)

[III. System Evolution 8](#_Toc119431172)

[IV. Solution Approach 9](#_Toc119431173)

[I. Introduction 9](#_Toc119431174)

[II. System Overview 9](#_Toc119431175)

[III. Architecture Design 9](#_Toc119431176)

[III.1. Overview 9](#_Toc119431177)

[III.2. Subsystem Decomposition 9](#_Toc119431178)

[I.1.1. [Flutter Multi-Platform Application Framework] 10](#_Toc119431179)

[a) Description 10](#_Toc119431180)

[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. 10](#_Toc119431181)

[b) Concepts and Algorithms Generated 10](#_Toc119431182)

[c) Interface Description 10](#_Toc119431183)

[Services Provided: 10](#_Toc119431184)

[Service name: Alarm Widget 10](#_Toc119431185)

[Service name: FlutterFire 11](#_Toc119431186)

[I.1.2. [Google Firebase] 11](#_Toc119431187)

[a) Description 11](#_Toc119431188)

[b) Concepts and Algorithms Generated 11](#_Toc119431189)

[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. 11](#_Toc119431190)

[c) Interface Description 11](#_Toc119431191)

[Services Provided: 11](#_Toc119431192)

[Service name: Firebase Cloud Firestore 11](#_Toc119431193)

[Service name: Firebase Authentication 11](#_Toc119431194)

[IV. Data design 12](#_Toc119431195)

[V. User Interface Design 12](#_Toc119431196)

[V. Test Plan 13](#_Toc119431197)

[I. Introduction 13](#_Toc119431198)

[I.1. Project Overview 13](#_Toc119431199)

[I.2. Test Objectives and Schedule 13](#_Toc119431200)

[I.3. Scope 13](#_Toc119431201)

[II. Testing Strategy 13](#_Toc119431202)

[III. Test Plans 14](#_Toc119431203)

[III.1. Unit Testing 14](#_Toc119431204)

[III.2. Integration Testing 14](#_Toc119431205)

[III.3. System Testing 14](#_Toc119431206)

[III.3.1. Functional testing: 14](#_Toc119431207)

[III.3.2. Performance testing: 14](#_Toc119431208)

[III.3.3. User Acceptance Testing: 15](#_Toc119431209)

[IV. Environment Requirements 15](#_Toc119431210)

[VI. Alpha Prototype Description 15](#_Toc119431211)

[VI.1. [Subsystem Name] 16](#_Toc119431212)

[VI.1.1. Functions and Interfaces Implemented 16](#_Toc119431213)

[VI.1.2. Preliminary Tests 16](#_Toc119431214)

[VII. Alpha Prototype Demonstration 16](#_Toc119431215)

[VIII. Future Work 16](#_Toc119431216)

[IX. Glossary 16](#_Toc119431217)

[X. References 16](#_Toc119431218)

[XI. Appendices 17](#_Toc119431219)

# Introduction

Your introduction should describe your project in sufficient detail such that the alpha-prototype description will be easy to follow. This should include material similar (lifted from and edited) from your Project Description document. The sections to focus on will be:

* Introduction - likely needs major editing since the project has evolved
* Background and Related Work
* Project Overview
* Client and Stakeholder Identification and Preferences

Add a brief paragraph to the introduction and explain the purpose of this document as a summary of the project’s progress and technical details about your engineering efforts.

# Team Members - Bios and Project Roles

Include an entry in a narrative form for each of your team members. The goal is to demonstrate the team’s skills and project coverage. This is not just a pasted in resume, but a summary of your involvement in the project, and your technical interests. Feel free to lift from your Team Inventory to include:

* Name
* Degree plan
* Project role - which aspects you’re responsible for
* General areas of experience and technical interests

Example:

Aaron Crandall is a computer science student interested in artificial intelligence, satellite development, and clock making. His prior projects have included smart homes, radio controlled dirigibles, and programming clocks. Aaron’s skills include C/C++, Python, RabbitMQ, Genetic Algorithms, and delinting. For this project, his responsibilities include developing the Gamma Module, leading user experience feedback, and delivering sandwiches.

# Project Requirements

# Introduction

The LAMI team has been tasked with the continued construction of the LAMI app for medication reminders for the residents in the LAMI residence. As stated, this app will be used for the residents of the LAMI house who have mental illness and will serve as a reminder app that will help them take their medication at specified times. There will be a sign-in for both the caretakers and the residents so the residents can have safety in knowing the only people that can view their medications are the caretakers who will act as a secondary reminder (in the case that the resident does not take their medication after being prompted by the app). The end goal of this project is to create an app that is easy and enjoyable to use, making the task of remembering to take medication as easy as possible.

# System Requirements Specification

* Residents of the house should be able to access/run the application on iOS or Android operating systems.
* Residents of the house should be able to access/run the application on iOS, Android, or in a web browser.

## Use Cases

**Story:** Helene Davids is the employee currently overseeing the house and one of the residents, Oliver James, has just been prescribed a new medication. Helene wants to make sure that Oliver has set a reminder to take his new medication, but Helene isn’t scheduled to go back to the house until 3 days from now. She pulls out her phone and logs on to the LAMI app. She selects Oliver’s name from the list of current residents and sees that he has already set a reminder and thinks “I don’t know why I was worried, Oliver is awesome!”.

**Story:** Micah Ferguson is the employee currently overseeing the house and one of the residents, Brandy Hartford, has been acting a little different than usual. Micah wants to make sure that Brandy has been taking all of the medications she is supposed to. Micah logs on to the LAMI website, selects Brandy from the list of current residents, and opens her progress report. He sees that she has not been very consistent with taking her medications, so he makes a plan to talk to her and see how he can help.

## Functional Requirements

### Improvement of the UI

The main function of this project is to create an alarm and the previous team that worked on this project was able to do just that but didn’t have lots of time to make the app look as nice as the clients wanted. Not to say the UI made from the last team was bad, but they thought it was a bit bland and wanted something that would draw more attention toward the application. The clients want a UI that uses calm colors like the darker blues and greens and that has a nice layout; something like Duolingo’s UI in terms of layout. Like having a side bar for all the parts of the application like settings, calendar, and current medications. As this is something that the client has asked for and the main feature of the app is almost complete this would defiantly be at priority level 1. It’s not essential but it is something that can be worked on and improved as we move through the processes of making this app.

Repeat the above for each requirement.

### Website Implementation

The client wants the application to also be online where other employees of the LAMI house can access their app accounts to check up on residents in the LAMI house. This requirement was added since the client wanted the residents and caretakers to have more accessible platforms to choose from for the application. Additionally, the client wanted the caretakers to be able to be able to check on the client’s medication times and if they took them on the house computer in the house. This was one of the main talking points when talking to the client, so the priority level of this would be 0 since part of their specifications included the website implementation.

### Resident and Caretaker Accounts

While its nice to have the timers for the residents the caretakers can’t tell if the medication has been taken or not so the creation of caretaker and resident accounts will allow for different permissions between users. The client asked for this since they wanted an easy way for the caretaker to see if the medication was being taken by the resident. The creation of this feature is priority 0 since allowing the caretaker of the house to see if the residents have taken their medication is something that was talked about a lot with the client.

### Resident Surveys

Another tool that will help the caretakers of the house is after taking medicine periodically a medication survey will come up to ask the resident if the medication is working well and how they like the medication. The client wants these surveys so that they can make sure that the medications that the residents are taking are not negatively impacting them or that maybe they need some new medication that is either stronger or weaker. This seems like it would be a priority 0 program in the application, but the client specified that this was something that we can do if we get around to it, so I would place it at 2.

## Non-Functional Requirements

List the non-functional requirements in this section. Non-functional requirements define system properties (e.g. reliability, response time and storage requirements, etc.) and constraints (e.g. I/O device capability, system representations, etc.)[[1]](#footnote-1)   
  
Generally, non-functional requirements take the form "system shall be <requirement>."

Process requirements may also be listed here (e.g. specifying a particular programming language or development method.) This will include any general testing plans, but there is a later assignment that will go into much greater depth about testing the product.

Please refer to Section 4.4.7 in the book “Object-Oriented Software Engineering” for example categories of non-functional requirements.

You may use the following template for non-functional requirements (Please remove the color formatting in your final document):

### Must be able to support around 10-15 users

The client wants it so that it is possible to have everyone in the house on the server at once so that no one isn’t reminded to take their medications, or a staff member can’t check if a resident has taken their medication.

# System Evolution

As we go on to finish the application for the client we may run into problems or may need to hang things as to help specific clients. So, as we build the application, and we make the web application we may need to make minor alterations for the web application alteration. On top of that adding the different user profiles may cause a problem on the firebase side of things but that will be solved with time and understanding of firebase. As for the UI, it could be changed multiple times depending on how the residents of the house react to it initial designs so constant restructure may be needed.

# Solution Approach

# 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.

# Test Plan

# Introduction

## Project Overview

The software that will be tested is the LAMI full-stack medication reminder application. Some components of this application will be tested like the apps alert component where we will test the ability to alert the user of medication times and see if they can be reminded of multiple medications on the same time. Also, to test the sig in feature to make sure the database is registering people correctly. So that the residents and caretakers can access their accounts.

## Test Objectives and Schedule

We plan to test the soon, however at present we would only be testing things that were tested by last years team that worked on this project. There will be tests, but it will be mostly edge case tests that the previous tests didn’t tackle. The test will most likely be created by some time next month when we get farther into our implementations of the application. When the tests are finished, we will output a code file that will be for testing features and most likely a video displaying some full application tests.

## Scope

The purpose of this document is to give insight into what we are going to test and how we will test those things. Through this document we will discuss testing strategies we will be implementing to test functions and implementation of our code. However, as previously started we will most likely nit be making test since much of the logical code is implemented from last years LAMI team. So, we will be only adding minimal tests since the majority of what we will work on is UI based which is hard to test.

# Testing Strategy

Describe the overall approach to testing and provide the overall flow of the testing process. An example is provided in Appendix A.

Will you be using Continuous Integration (CI) and/or Continuous Delivery (CD) in your testing? If you’re not using CI or CD, make a \*very\* strong case for your decision.

# Test Plans

Describe the plan for testing your project in the context of the following testing activities. You may include additional test activities, if necessary.

For each of the following activity, describe how the testing will be conducted. What would be the sequence of events, and how will the testing activity take place? Please refer to the CptS422 class notes for details on testing strategies.

## Unit Testing

The methods for the following functional requirements for each client will be unit tested using the Flutter test package:

House Residents

* + Log-in to an existing account
  + Sign-in up a new account
  + Add an alarm
  + Turn an alarm on/off
  + Edit an alarm
  + Delete an alarm

Staff

* + View a user’s alarms
  + View all existing user accounts

## Integration Testing

A Flutter package for integration testing exists that allows us to test each widget in the application. With this testing framework, rightfully named integration\_testing, we will test each button by triggering it and checking for the correct response.

## System Testing

## Functional testing:

As stated previously, the major functional testing requirements for each client are as

follows:

House Residents

* + Sign-up for a new account
  + Log-in to an existing account
  + Add an alarm
  + Turn an alarm on/off
  + Edit an alarm
  + Delete an alarm

Staff

* + View all existing user accounts
  + View a user’s alarms

## Performance testing:

To test the performance of the application, we will register enough accounts to ensure the free Firebase Firestore database plan can handle the quantity of accounts needed for all residents at the house.

## User Acceptance Testing:

We will test the app by using it on our personal phones and computers for an extended period, as well as messing with it to try and cause errors. In addition, we are also able to post the web app early, to let the staff look at and use both the resident and staff sides of the application. We are also able to do this for free, as every Firebase project is provided with a free subdomain on the web.app and firebaseapp.com domains.

# Environment Requirements

Specify both the necessary and desired properties of the test environment. The specification should contain the physical characteristics of the facilities, including the hardware, communications and system software, the mode of usage (for example, stand-alone), and any other software or supplies needed to support the test. Identify special test tools needed.

To do our tests we will only need the phones and the website connected to the application so that we can test our implementation. By having both the android and the iPhone we can test both phones application to make sure there isn’t any problems with one or the other. Then the website will need a computer component to be able to be tested so a laptop or computer will be needed to run these tests for the website application.

# Alpha Prototype Description

\*\*\*This is the first section that is truly new for this document\*\*\*

Describe your alpha prototype implementation. Please format this section according to what you think is the best way to describe your prototype. The following is just a suggestion.

Explain what parts/subsystems of your proposed architecture your team started to implement. Mention the current progress at each part, i.e., how much of the proposed functionality you have completed.

For the parts that you already started to work on, explain whether you have tried to integrate them with the rest of system. In other words, explain which of the interfaces in your architecture have you started to implement.

If you have performed any tests on your alpha prototype (or the subsystems of your prototype), briefly explain your findings.

I recommend to include plenty of images and pictures of the following where appropriate:

- any diagrams/figures that visualize various features of your prototype;

- the screenshots of your user interfaces;

- the screenshots of your test programs;

- pictures of your team testing and debugging the devices, programs, etc.

A well-thought and clear diagram is better than long and descriptive text.

If your document starts to be very long due to screenshots and diagrams, please put at least some of them into an appendix to this document.

For each subsystem that you have implemented in your alpha prototype, you may include the following sub-sections.

## [Subsystem Name]

### Functions and Interfaces Implemented

List and describe the implemented functionality. Explain the remaining work.

### Preliminary Tests

Report any test results for the unit and integration tests that you performed on your prototype. This subsection is a good place to include screenshot images from your tests (if applicable). A notable component here would be to include the results of your CI/CD status. Hopefully master still builds, right?

# Alpha Prototype Demonstration

Summarize the highlight of your prototype demonstration to your mentor. The items to discuss in this section may include the following. (Please include all other necessary details in addition to the following).

1. Summary of what you showed to your mentor.
2. Your mentor’s comments/suggestions on your prototype.
3. Your mentor’s questions to your team and your responses to those questions.

After testing your prototype and demonstrating it to your mentor, you will have a better idea whether the initial design you proposed earlier will work. Additionally your mentor might suggest modifications to your current design. In this section list and explain all design modifications that you plan to make based on your preliminary test results and mentor comments (if applicable).

# Future Work

List the major tasks for the second semester and briefly explain your plan to complete them.

# Glossary

Define technical terms used in the document.

# References

Cite your references here. -- Ensure you’re pulling them from your earlier works!

For the papers you cite give the authors, the title of the article, the journal name, journal volume number, date of publication and inclusive page numbers. Giving only the URL for the journal is not appropriate.

For the websites, give the title, author (if applicable) and the website URL.

Please use either Chicago or IEEE format for your citations

# Appendices

As needed, copy over your appendices for the various sections. You can have as many appendices as required. Normally, they’re numbered with letters:  
Appendix A  
Appendix B  
…  
Appendix *n*

1. https://en.wikipedia.org/wiki/Non-functional\_requirement [↑](#footnote-ref-1)