TENSE MED DYNAMICS

Software Design Document

February 7th, 2025

Project Sponsor: Jesslynn Armstrong

Ryan Lucero, Cathy Ly, Zach Trusso, Marco Castrita

Version 2

Table of Content

Introduction

In the rapidly evolving medical world, it is crucial to create innovative solutions that are both accessible and impactful; however, the current technology for photobiomodulation therapy are the complete opposite: bulky, inaccessible, awkward, and expensive. Infrared light technology made waves in the beauty/skin care industry for its ability to repair skin, boost new cell growth and enhance skin rejuvenation. Alongside with a variety of other research-supported benefits, light technology has the capabilities of reaching other industries like clinical, emergency, and military uses. Tensegrity Medical's goals are to revolutionize photobiomodulation (PBM) therapy with LightDose, a non-invasive, wireless device designed to empower patients through real-time biometric tracking and personalized health management.

The focus of LightDose is to make PBM therapy more accessible, affordable, and user-friendly. In collaboration with Mechanical Engineering, Electrical Engineering, and Computer Science, LightDose integrates a wearable PBM device that connects to Bluetooth via a Flutter-based mobile app, securely storing data in Firebase, ensuring HIPAA compliance. The app will also incorporate gamification features that will let users track progress on therapy, monitor health biometrics, and provide real-time feedback. LightDose offers cost effective vascular health improvement and pain relief by moving PBM therapy from the clinical setting to remote, self-managed care.

Key functionalities will include real-time biometric tracking, secure data transmission, and multi-device compatibility. The app should be fast, intuitive, and accessible, loading within five seconds and displaying biometric information in less than a minute. It will comply with WCAG 2.1 guidelines, ensuring usability for older adults and people with disabilities. It must maintain a 99% data synchronization success rate with minimal battery power consumption. Security is a top priority, requiring strong encryption, multi-factor authentication, and role based access control. The patients' data will be encrypted using the Advanced Encryption Standard (AES) for storage and transmission to allow the healthcare provider to analyze trends and adjust

treatment plans remotely. AES is a symmetric encryption algorithm that securely encrypts and decrypts data using block ciphers with key sizes of 128, 192, or 256 bits. The firmware of the LightDose device will be written in Arduino, allowing compatibility with most medical hardware and future scalability. The app will keep a stable Bluetooth connection, enabling users to control the device, check battery levels, and receive notifications in real time.

LightDose addresses the shortcomings of in-person PBM therapy, offering more accessible and engaging treatments while reducing healthcare costs. By integrating wearable technology, biometric data tracking, and secure remote monitoring, LightDose represents a leap in personalized healthcare and self managed-therapy.

Implementation Overview

Tensegrity Medical's LightDose project aims to revolutionize PBM therapy by creating a compact, non-invasive, wireless device that connects to the LightDose application in order to track its users' health. LightDose aims to overcome these limitations by utilizing modern technology in order to provide a portable, affordable and user-friendly alternative. LightDose will track advanced biometric data, as well as offer personalized health results in real time. The solution seamlessly utilizes advanced technology into a user-friendly interface. The device's main functionality is built around real-time biometric monitoring, providing users with feedback and health management insights all while ensuring seamless connectivity and secure data management

Key Technologies and Tools Used:

Frontend Development

Flutter: To develop a responsive, interactive, and cross-platform UX that will provide its users with optimal use.

Backend and Cloud Infrastructure

Firebase: The core backend platform will be Firebase, utilizing Firebase's real-time database synchronization and user authentication in order to host dynamic content.

Biometric Data Integration:

Bluetooth Connectivity Bluetooth connectivity from the device will be used in order to track biometric health data like oxygen levels and heart rate.

Security and Data Management

Firebase Authentication (FA:) FA provides secure authentication and authorization for user accounts, making sure that only authorized users can access the biometric data.

Firebase Security Rules: Implements fine-grained access control to protect sensitive biometric data.

By combining these technologies with a thoughtful design approach, LightDose aims to deliver a robust and scalable solution in order to revolutionize PBMtherapy.

Architectural Overview

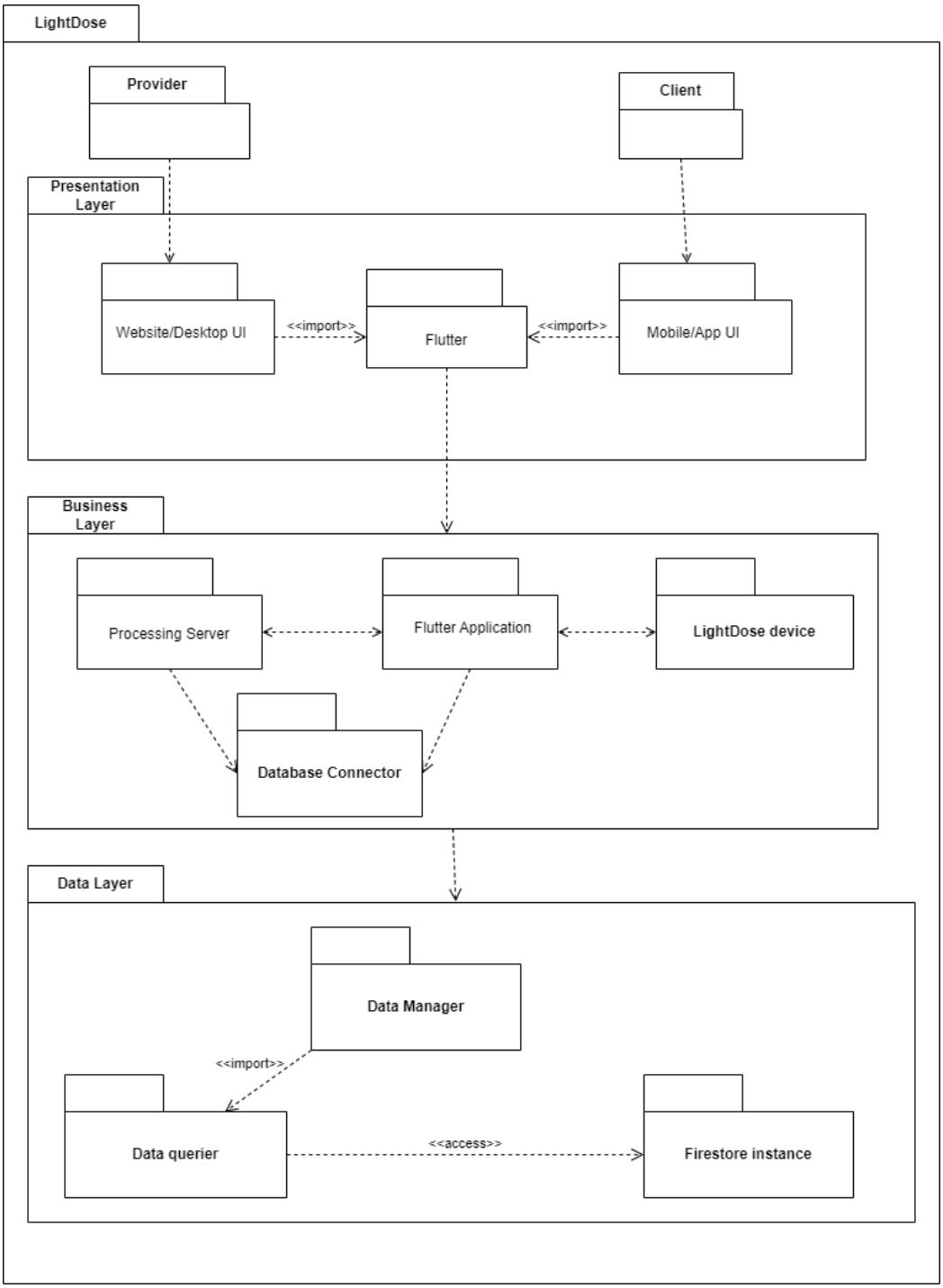

Figure 1: High-Level Layered Architecture Diagram for LightDose Application

Explanation of Components

Component	Description
User Interface	A simple, intuitive design built using a cross-platform approach, allowing it to run on any device without hardware-specific modifications.
Firestore API/Module	Enables remote access to Firestore's NoSQL database, organizing data into collections (tables) and documents (records). The module also includes custom functions and error handling
Bluetooth API/Module	Facilitates communication between the app and the LightDose device, allowing data transfer and command execution without Wi-Fi. It determines when and how to gather or transmit data.

Information and Control Flows

Most of the application is broken down into individual pages which act like object oriented programming classes, as they have their own functions, store their own data, and can pass data to other pages. Some pages are self-contained, where anything on that page stays on its page and doesn't make any API calls or pass data to another page.

The login and sign in pages are both pages that interact with the Firebase API, both to request data in order to check if data for a specific

person is already inside the database, or to pass data into the database in the form of adding a new user. The login page then passes the user data to the homepage inorder to know what data to display. The homepage acts as an interface for most other pages as it will display what other pages can be accessed by a user. The Bluetooth page is an example of a self-contained page that stores data, and displays the data to the user, but doesn't pass any of that data to other pages. However, the Bluetooth page does make API calls to both Firestore and Bluetooth in order to transfer the data to the cloud, or the device.

Architectural Style Influences

Client-Server Architecture (Thick Client)

Client-Server architecture is a model where the client—in this case, the LightDose application—requests data from the server, the Bluetooth server of the LightDose Device. The device then sends the requested data to the app and will on request either turn on or off the lights or sensors on the device. The system for this project is considered a thick client version, as the data will be stored on the mobile device running the application.

Layering Architecture

The application uses the layering architecture, as it is broken down into the three main sections. The Presentation Layer which is the user interface (UI) of the application, as it serves as a visual interface for clients to access the application, and connect to the app. The Business Layer which is the backend of the Flutter application, which is broken down into individual modules that each control a different function of the app and is responsible for making API calls, and processing the data returned from the calls. The data layer which is the API interface that connects to the Firestore database to access the data stored in the cloud, and the Bluetooth module which collects data from the LightDose device.

Stream-Based Architecture

The app is in control of the device, through a Bluetooth connection which serves as a stream of data collecting data from the device, and sending signals to the device inorder to trigger turn on/off events for both the lights and the device. Additionally, the data being streamed from the device triggers an update in the real time data display on the app, which will show a visual data display.

Module and Interface Descriptions

Presentation Layer

Mobile Interface

Focuses on the UI and the interaction by using Flutter framework to render the iOS and Android UI components. This takes care of the visual presentation elements such as buttons, forms, and text input fields meanwhile ensuring smooth user interactions. The mobile interface module directly interacts with the user and communicates with the business logic layer and serves as the frontend component of the application.

Serving as the frontend layer and providing an interactive interface to the user, the mobile UI sits in the presentation layer and works as an entry point for users and passes user data and requests to the business layer of processing. When the business layer responds with the necessary data or results, the mobile UI will update accordingly.

A majority of interfaces on the app will consist of buttons and text inputs. Buttons help users navigate like the dashboard will be made up of a variety of different navigation buttons that could take them to their therapy session or their in-app achievements. Text inputs will be used in

forms like creating accounts or changing profile information. In Figure 2, the UML diagram describes the actions of the user interaction features like buttons and text inputs.

The buttons will contain text with a short instruction or simple description of its purpose and will be able to perform an onPress() function, which in certain cases, will navigate to a different screen in the app. Text inputs will contain a placeholder text until the user changes it. Once changed, the text input will perform onChange(). In this example, when the user changes in the text in the settings screen, it will update the changes to the profile.

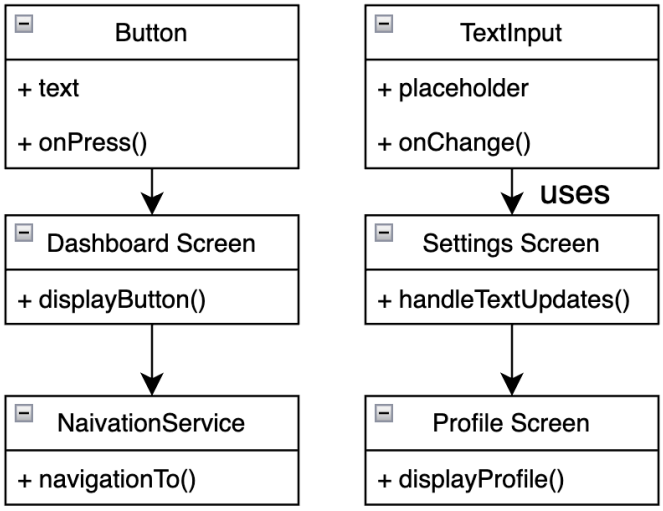

Figure 2: Example UML Diagram of Buttons and Text Input for Mobile Interface Component

Discussion of the Provider's desktop UI will not be discussed due to the project's focus on the mobile application.

Business Layer

Processing Service

The central point for handling the business logic of the whole application, this module will receive input data from the presentation

layer and perform the needed operations, calculations, or transformations that are based on the application's needs. When the processing finishes, it returns data to the mobile UI to update the user interface.

The processing service sits in the business layer and is in between the mobile UI and the data layer. This layer implements the application's core business logic and facilitates all computational tasks.

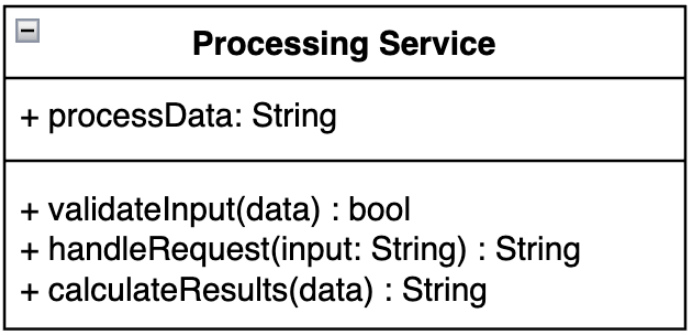

Figure 3: UML Diagram of Processing Service

The simple UML diagram for processing services outlines the tasks that are required for this module of the application. In the mobile UI, when the user fills out a form (i.e. creating an account) the processing service will take the given information and decide what it will need to do next before sending data to the data connector or another module. For example, when the user creates an account, the processing service will ensure the given information is validated and meets the requirements such as email format and password strength.

Database Connector

The connection between the processing service and the Firestore database handles all data retrieval, manipulation, and communication. The database connector's responsibilities are to perform CRUD operations (Create, Read, Update, Delete) and to enforce the integrity and efficiency of the data handling.

While both the Flutter application and the LightDose device are important parts of the business layer, this section of the documentation focuses on the core business logic and its interactions with the database.

Data Layer

Data Manager

Responsibilities of the data manager can be described as coordinating all data-related operations in the application. Tasks such as providing an abstraction layer between the backend and the business layer. The component of the data layer can provide methods to fetch, store, update and delete data from Firebase. All data interactions happen through this component which will allow the application to communicate with the Firestore database.

This component is also responsible for handling the Bluetooth data coming from the LightDose device. It manages the connection to the device and collects the sensor's data, meanwhile ensuring that the data is formatted and processed properly. The data manager will store the data into Firestore. The integration of the data manager provides a seamless communication between the device, the business logic, and the database.

Data Query

Retrieves data from the database based on the parameters or queries, this component defines the needed operations and criteria to fetch certain sets of data like getting user profiles or session records. The data query works by sending a structured request to the database and receives the appropriate results based on the parameters. This can look like filtering conditions, ordering results, or limiting the number of records.

Data query can also be seen as a part of the data manager's responsibilities but it is a tool used by the data manager when it's interacting with the Firestore database to get and manipulate data.

Firestore Instance

This component refers to the actual connection and the client interface to the Firestore database. Firestore SDK is initialized and the interaction with the Firestore cloud database takes place in this component. Providing the essential functionality for authentication, reading, writing, updating, and deleting data, Firestore instance gives the services to perform any operations.

Implementation Plan

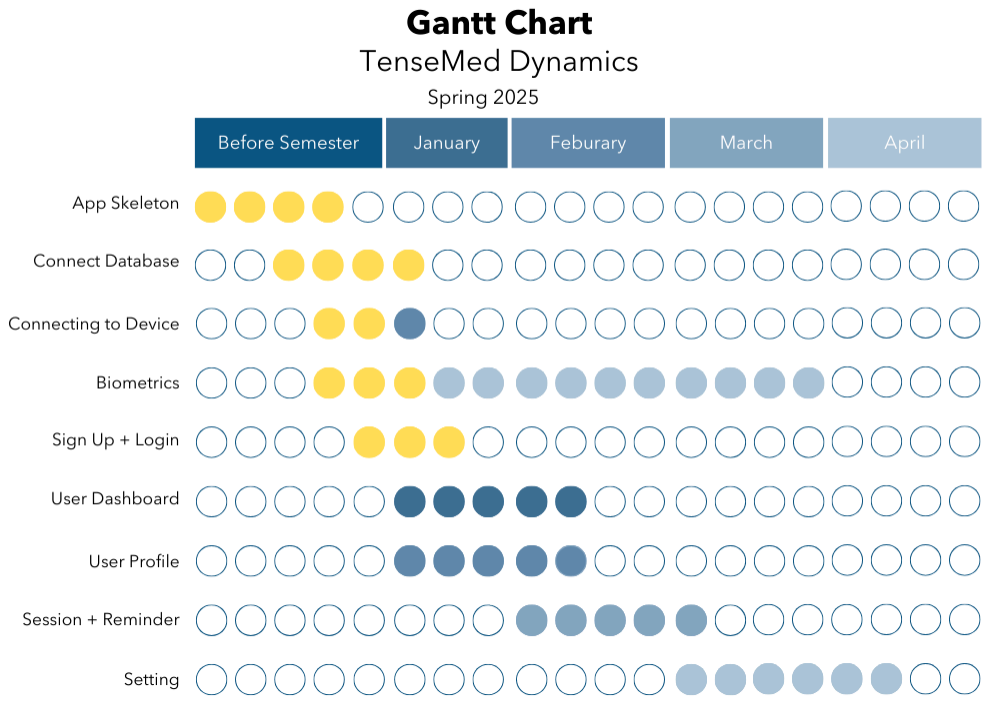

Figure 4: TenseMed Dynamic's Gantt Chart for Spring 2025

Above showcases TenseMed Dynamic's Gantt Chart for Spring 2025 semester. The yellow indicates the progress of the team and the work that has been completed.

The circles within the Gantt chart show a week of work. Each row represents the minimal viable product (MVP) features of the project and is organized in case there are unexpected problems. In the chart, biometrics last longer than one month due to its importance and the way it connects to the other features of the app.

Before the spring semester started, the app's skeleton and the database had been set up and accounts could be created within the app. Tracking biometrics properly in an app is currently in development and tested since last semester. In January, the team began developing user interfaces such as the dashboard, the profile,and reminders. Each month, each member will focus on implementing a new feature for the app while ensuring that security is considered throughout the process.

Conclusion

LightDose aspires to transform PBM therapy by making it compact, non-invasive, and user-friendly for health management. The app provides secure, personalized care with the use of advanced technologies such as Flutter, Firebase, Bluetooth connectivity, and real-time biometric tracking. The architecture has been carefully designed with scalability in mind, featuring a client-server model and layered approach for multi-device compatibility. This will not only overcome the challenges of existing PBM therapy devices, but also make advanced health management both accessible and affordable to patients. LightDose places a strong emphasis on security and usability to change how people manage their health remotely and contribute to a more efficient and cost-effective healthcare system.