CyberJewelry

Saniya Singh, Madi Davis & Shize Che
Department of Electrical and Computer Engineering, Carnegie Mellon University

***Abstract*— Wearable tech has become ubiquitous in the last few years, but most of the widely available options are focused on utility: collecting health data or providing access to smartphone features. These options have extremely limited customization lacking sufficient avenues for self-expression. Our system hopes to provide a solution to this in the form of wireless, customizable digital jewelry.**

***Index Terms*— BLE, DotStar, LCD, STM32**

I. Introduction

In the last decade, wearable technology has become increasingly ubiquitous in our society. Most of the widely available wearable tech options are focused on providing some utility such as collecting health data (ie: Fitbit, Oura Ring) or increasing access to smartphone functions (ie: Apple Watch). These devices all have limited customization options which do not provide sufficient avenues for the wearer to express their individuality. Due to these limitations, there exists a market for digital wearable technology that exists purely for the sake of style and self-expression. These bold electronic accessories used to be reserved for the realms of festival wear and costuming, but recently they have been slowly entering the mainstream fashion consciousness. With our project, we aim to design a set of digital earrings that are wirelessly customizable via Bluetooth and an iOS mobile application. This device would have a variety of use cases from daily jewelry to clubwear for people who want an elevated accessory that provides near infinite customization options.There are very few competing technologies in this area, and no commercially available products that provide the features we want to implement in our system.

II. Use-Case Requirements

From the description of our system and the needs of our user base, we proceed by introducing the use case requirements that will guide our design process.

A. Physical Design

a. Weight

We want our users to be comfortable with the weight of the device that will be hanging from their earlobe. We've decided on a maximum weight constraint of 20 g per earring.

b. Temperature

We want our users to be comfortable with the temperature of the device because it will be in close contact with the user's skin. We've decided on a maximum temperature constraint of 40° C.

c. Material

We want the material that is used for the casing of the device and the earring post to be skin-safe and hypoallergenic. For this, we will be using surgical grade steel for the portions of the earring that will be in direct contact with the skin.

B. Functionality

a. Update Speed

We want the user to be able to quickly modify the pattern on the earring without a lot of lagging after pushing the new design to the earring. Knowing the data transmission speed for the BLE protocol and estimating the maximum size of a single data pattern for the LED matrix and the LCD screen, we have decided that 1500 milliseconds is an appropriate upper bound for update speed.

b. Setup Time

We want users to have a quick and painless way to set up the device. We are using BLE to make the device more comfortable to wear, but this means the user's will need to pair to the device from their iOS device.We would like the general setup time for the device to come in under 90 seconds.

c. Battery Life

We want the device to be usable for general eventwear so we would like to achieve a minimum battery life of 3 hours for the DotStar Matrix and 45 minutes for the LCD screen versions of the device.

III. Architecture and/or Principle of Operation

Our product will consist of three primary subsystems: Hardware Design and Firmware, the mobile application interface and the device enclosure. System composition and Interactions are outlined in Fig 1. Our system accounts for two possible display options, an LCD Screen and an 8x8 DotStar LED Matrix. Each display option will have slight variations for Hardware/Firmware and Mobile interface implementations. The physical component of the project consists of a wearable housing that attaches to the user's ear lobe via a post-stud backing. A custom fabricated PCB containing a controller and display (LED or LCD) will be permanently fitted to the casing. The mobile interface will enable the user to create custom pixel designs by varying the hue, brightness and spatial characteristics of the display. Designs will be forwarded from the application to the device controller through a bluetooth connection. The microcontroller will subsequently communicate design updates to the display via a SPI bus.

Fig. 1. BLOCK

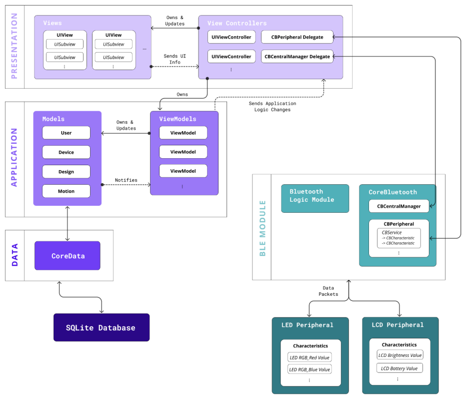


Fig. 2. Detailed Block Diagram of IOS Mobile Application

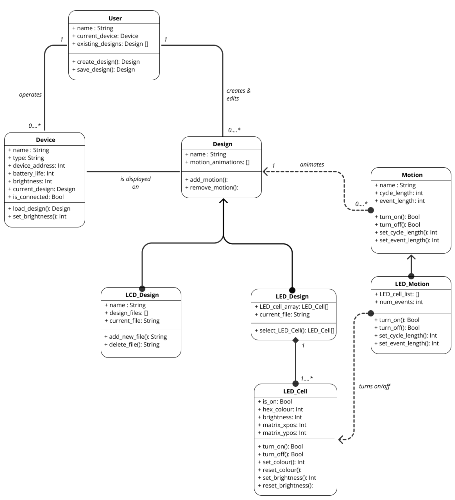


Fig. 3. Mobile Application UML Class Diagram for Data Models

A. HARDWARE DESIGN & FIRMWARE

The hardware consists of two components: microcontroller and display. For the microcontroller, we will use STM32WB55 due to its built-in BLE functionality. With BLE functionality built-in to the microcontroller, no additional BLE supporting hardware is necessary, which simplifies the PCB and contributes to maintaining the size requirement. For the display, we will use an AdaFruit DotStar 8x8 RGB LED matrix for MVP and AFK128128A0 LCD screen for the final. Both displays fulfill our size requirement. A caveat for the MVP from using the AdaFruit DotStar is the display requires 5v operating voltage but the STM32WB55 requires 3.3v operating voltage. Therefore, in order to only use one voltage source, namely a coin battery, we plan to incorporate a voltage regulator to regulate the 5v voltage source to a steady 3.3v voltage supplied to the microcontroller.

The firmware will be developed based on the HAL peripheral driver library. In realizing the Bluetooth data transfer as well as programming the screen using the microcontroller, we need to use the BLE and SPI peripherals. The HAL library provides functions to drive these two peripherals. A working directory with a makefile that automatically compiles and links HAL library is set up in our software stack. We will use these library functions to drive the peripherals and write higher level functions to develop the application loop. The application loop runs on bare metal, and it's a while loop that keeps checking if the user has sent a display reprogramming request. If so, the software receives user RGB data and displays them onto the display. Otherwise, it keeps letting the LEDs/screen display the current pattern.

B. MOBILE APPLICATION INTERFACE

System Architecture

Our User Interface will consist of an IOS 16 application, implemented in Swift using the Model View View-Model Architecture, as detailed in figure 2. The system can be divided into 4 main components, outlined below:

<u>Presentation Layer:</u>
Contains all UI components including layouts, screens, buttons, sliders etc. defined as Swift UIView Objects (for details on UI see Appendix B for Wireframes). View Controllers will be implemented to handle all UIView events, interactions and transitions.

<u>Application Layer:</u>
Contains models to keep track of all data classes and subclasses, as well as ViewModel Objects perform all logic associated with model instances.

<u>Data Layer:</u>
Apple's CoreData Framework will be used to locally store permanent application data.

<u>BLE Module:</u>
Apple's CoreBluetooth will provide an interface for BLE communication with the controller. A Bluetooth logic module will be implemented to format application changes as BLE advertising and data packets.

C. DEVICE ENCLOSURE

The device enclosure requires putting the microcontroller PCB and the display together. To do so, we decided to use PCB connectors.

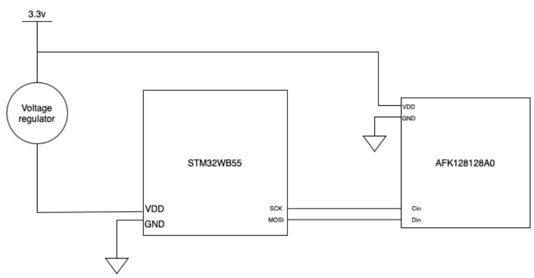


Fig. 4. MVP device enclosure connection

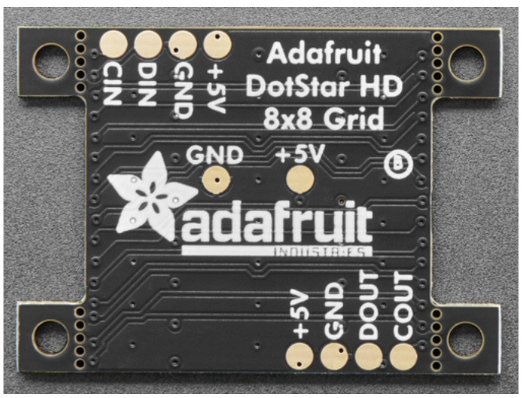


Fig. 5. AdaFruit DotStar LED Matrix PCB

Fig 4 above shows the connections we need to make between the microcontroller ports and the display ports. The Vdd and GND supply the power to the display and the Cin and Din are the clock line and data line of the SPI peripheral. Note that no data enable line is needed to drive the LED matrix. To create these four connections, we will make our microcontroller PCB the same size as the LED matrix and solder four connectors at the same position as the corresponding to the four ports on the top left in Fig 5.

IV. Design Requirements

A. HARDWARE DESIGN & FIRMWARE

Upload Latency

The upload latency of a new display should be no more than 1500ms. Users don't want to wait for more than 1.5s to have the new display uploaded. This performance requirement is important to bring user satisfaction.

Battery Life

The total battery life for our device should be no less than 3hr for the LED Matrix Display and 45 minutes for the LCD display. We want users to expect a sufficient amount of battery life for them to wear the device during social events.

Temperature

We want the operating temperature to be less than 40 degree Celcius, which is an acceptable temperature to the human body,

B. MOBILE APPLICATION INTERFACE

Setup Time

The time for user to setup the bluetooth pairing on their phone should be no more than 90s. Because bluetooth pairing can be unpredictable, we give ourselves a relatively relaxed design requirement to allow flexibility. We believe the 90s requirement is a good gauge of how long users would like to wait for bluetooth pairing. Anything longer than that may cause users to give up.

C. DEVICE ENCLOSURE

Weight

The weight of our device should be no greater than 20g. This requirement is for user comfort. We want our earring to be as light as possible so that users can feel comfortable and can't tell the difference between wearing this relatively bigger earring than wearing normal earrings.

Material

The material of all parts needs to be skin-safe. This requirement is self-explanatory. We don't want our device to cause skin problems.

Sturdiness

We want our device to be sturdy. We can't predict what users will do when wearing the device so we need sufficient guarantee on its sturdiness to make sure the screen doesn't fall from its connection to the microcontroller PCB..

V. Design Trade Studies

A. HARDWARE

When deciding on the hardware components for our device, we made a few tradeoffs to improve the aesthetics and wearability.

PCB

First of all, while it's easier to deploy a pre-made microcontroller board such as the nucleo boards or arduino boards, we decided to design our customized PCB to trade development difficulty for aesthetics and wearability. We want the microcontroller board to be the same size as the display hardware and none of the microcontrollers boards on the market fit our size requirements. As a wearable device for fashion purpose, the aesthetics and wearability of the device are nonnegotiable. Therefore, a customized PCB that has equivalent size as the display is necessary.

Microcontroller
To improve user customizability, we want to give users the power to adjust and design the display of their earring on their phone. So we looked for microcontrollers that come with built-in BLE. Because all group members have development experience with stm32, we decided to use the STM32WB55 as our microcontroller to receive bluetooth signals and drive the display. The programmability given by st-link technology also makes stm32 series a common choice for PCB design. Additionally, we are able to find a full peripheral driver package for STM32WB55 on GitHub, which will make the process of experimenting BLE and SPI peripherals easier.

Display
We prioritize aesthetics when choosing our display technology. To maximize aesthetics, we ideally want a high-resolution screen comparable to a TV display. However, given the time constraint and foreseeable issues such as device temperature and battery life, it is too optimistic to use a high-resolution screen in MVP. We found an alternative approach with aesthetics requirements in mind. The display technology we will use in MVP is an LED matrix. With an 8x8 LED matrix, we can display a sufficient number of patterns and give users flexibility in creating their own designs. Using an LED matrix also makes it easier to develop and address issues related to device temperature and battery life.

B. MOBILE APPLICATION INTERFACE

The motivation for our design decisions is the need to develop an intuitive, accessible interface that presents users with an expansive number of device customisation options. We have opted for mobile development over web development to afford the user the convenience and flexibility to alter their design at any time, from any location. We considered multiple avenues for mobile development including Android Studio, Apple's XCode IDE or a cross-platform development kit such as Flutter.
We ultimately chose to build our application for IOS 16 utilizing Swift and the XCode development environment due to the expansive availability of documentation and supporting resources, and our access to a number of IOS devices for testing. Moreover, the iOS SDK provides integrated "Core" frameworks (e.g. CoreBluetooth, CoreData) that are directly applicable to our functional requirements (BLE Communication, Data Storage, Sensing). We settled on native development over cross-platform to simplify UI/UX development and testing.

User Experience / User Interface
To implement our application front-end, Apple's Native SwiftUI declarative framework and UI ToolKit will be employed to efficiently create a UX ecosystem that adheres to Apple's iOS composition and accessibility guidelines.

Data Storage
For the current scope of our project we have decided our mobile interface will be developed as a self-contained application, due to the scope of the capstone project and the limited need for back-end functionalities as a single user app. Application data will be stored locally in an SQLite database that will be interfaced through Apple's CoreData framework.

Peripheral Communication
As outlined above, we have chosen to make use of the iOS SDK CoreBluetooth Framework for BLE communication between our Application and Device. Our initial approach to managing bluetooth communication was developing a simple python script to send and receive data from peripherals. However, although this solution may be beneficial for initial testing, it is not compatible with our decision to pursue mobile development. We also considered other open source ios Bluetooth libraries compatible with SwiftUI/XCode, including BlueSwift and RxBluetoothKit. Whilst these libraries aim to offer a simpler interface than Corebluetooth, our research led us to conclude that working exclusively within the Apple SDK would minimize our risk of compatibility issues.

C. DEVICE ENCLOSURE

As outlined in our design requirements, our earring must be lightweight, ergonomic and constructed of a bio-safe material. Whilst we plan to refine our enclosure design through prototyping and user testing, consideration of these criteria allowed us to develop a list of design guidelines.

Size
The size of our earring enclosure must be as small as possible, in order to minimize the overall weight of the product, thus maximizing user comfort. An analysis of existing jewelry products on popular ECommerce websites (e.g. SSENSE, Farfetch, Nordstrom) revealed current products on the market range from 6.35mm-50.8 mm in width and 6.35mm-127mm in length.
Our Design also must be large enough to contain our display screen and custom PCB. Our Selected LED and LCD displays are 25.4mm x 25.4mm x 2mm and 30mm x 35mm x 2.6mm respectively.

Based this, we have determined the following size guidelines:

- Width = 35 ± 5mm
- Length = 60 ± 20mm
- Depth = 5 ± 2mm

Material
Weight and Skin Reactivity were our primary considerations for material selection. We initially investigated fabricating our enclosure from a biocompatible polymer, such as PLA or Polyethylene. These materials are low-cost, lightweight and are suited for use in medical grade products. However thermoplastics such as PLA are subject to UV degradation [2], compromising the potential durability of the product and limiting the potential environments the product can be exposed. Additionally HDPEs, such as Polyethylene, whilst more stable, are difficult and costly to manufacture on a small batch scale.

As a result of this, we researched the applicability of several metals and alloys. Although many existing earrings on the market are composed of gold, chromium, palladium alloys and nickel, these materials can lead to irritation, allergic reactions and contact dermatitis [3]. We also evaluated the applicability of several body-safe metals used in jewelry manufacturing including titanium, surgical steel and platinum.

TABLE I. EVALUATION OF BODY-SAFE METAL PROPERTIES

Material	**Design Consideration**	
	Density (g/cm3) **[4]**	***Cost ($USD per kg)***
Titanium	4.51	0.77
Surgical Steel	7.48	0.51
Platinum	21.5	31, 249

The data in Table 1 presents surgical steel and titanium is the most viable option due to their relatively low cost and low density. Although the low density of titanium is suited for our product's low weight constraint, titanium's low elasticity, hardness and high melting point limit our fabrication options. Thus, we believe utilizing a surgical steel such as SAE 316 Stainless Steel is the best option.

VI. SYSTEM IMPLEMENTATION

A. HARDWARE

The implementation of hardware includes two phases: prototyping and PCB manufacturing. We want to develop the prototype of the hardware on a breadboard first. For developing the system on the breadboard, we will use the STM32WB55-Nucleo as the microcontroller board. The nucleo board is easier to program and is therefore a good candidate for prototyping. The circuit connections will be done using jumper wires on the breadboard.

The firmware code will be compiled from the HAL library and loaded into the microcontroller flash memory using linker script. On top of the firmware code, higher level application functions that take in display data and configure the display will be developed also in the prototyping phase.

Once we can successfully drive the screen in the prototype, we will be confident that our circuit design can be turned into a successful PCB design. The PCB design will be created using fusion360 and once we are confident about its functionality, we will place the PCB order.

Because the BLE communication only involves the microcontroller and the user phone, no additional PCB components need to be added except a PCB antenna. Therefore, the development of BLE communication will be done after PCB manufacture.

If we can get the MVP hardware done before its due, we will look into the design of using LCD screens as the display technology. The LCD screen requires more energy and a more complex SPI peripheral. We will think about how to minimize the energy consumption if we end up designing the final screen version.

B. MOBILE APPLICATION INTERFACE

The Web Application interface can be segmented into three user flows, as documented in Appendix A: User Setup and Design Selection. Design Editing and Design Upload & Save.

User Flow 1: User Setup - After Launching the application, users will be directed to a "Bluetooth Connection" page. Landing on this page will direct an instance of CoreBluetooth's CBCentralManager to scan for nearby peripheral devices. Detected devices will appear on the bottom of the screen as a list of card style buttons detailing the device name and type (LED or LCD). When a user selects their desired device from the available options, "Device Home" will be loaded in the application.
This Home Screen will display a summary of device information including an image of the device, device name, the current battery life, the current uploaded design and a slider control top adjust the overall display brightness. A button in the top right hand corner of the screen will enable the user to toggle the device's bluetooth on and off. The home page will be the same for both types of device. Using the toolbar at the bottom of the screen allows the user to Navigate to the "Design Library" where they will be able to choose an existing design to upload to the device from a scrollable list of options, via a search bar. Alternatively Users may choose to create a new design by selecting the button in the top right corner of the screen view. Both

options will prompt the application to navigate to the "Design Edit" page.

User Flow 2: Design Creation - The type of peripheral device connected will dictate the "Design Edit" display and subsequent functionalities.
If the device features an LCD Screen Display, the "Design Edit" page will consist of a preview of the design rendered on stylised representation of the device. A control menu at the bottom of the screen will provide users with two options to customize their design. Selecting the "Visual" Tab of the menu will allow the user to upload files in the format of .jpg, .png and .gif. Selecting multiple files will display each visual as a slideshow, with a default display time of 10 seconds per image and a default transition length of 500ms. Controls in the "Motion" tab will allow the user to adjust slideshow parameters. The"Transition Length"slider will change the transition time between visuals (from 0 to 1000ms) and the "Event Length" slider will alter the amount of time each visual is displayed (from 0 to 30 seconds).
If the device has an LED Screen Display, the "Design Edit" Screen will feature an touch-interactive model of the 8x8 matrix in the center of the page. The user can tap on one or more of the 64 graphical representations to edit display properties of the corresponding LED units on the device. Selecting the "Hue" tab from the control menu allows the color and brightness of the LED(s) to be adjusted using 2 separate slider controls. Selecting the "Motion" tab allows the user to animate a single or group of LED(s). Animation can be customized by selecting custom preset animations (E.g. blink, fade, pulse etc.) or setting custom values for the Event Length (length of time an LED is on/off) and Transition length via slider widgets.
Selecting the Reset button in the top left hand corner of both views will; reset the design back to its last saved state.

User Flow 3: Design Upload and Save
Selecting the button in the top right corner of the "Design Edit Page" will save the current design and upload the device to the design.

C. DEVICE ENCLOSURE

The physical enclosure for our device will be modeled using Blendr, to aid us the flexibility to design a custom product that meet our design and user requirements. The CAD model will be designed as a modular system to enable the hardware to be integrated without additional fabrication steps. We plan to have our final product manufactured from surgical steel using an external lost-wax casting or CNC service.

VII. Test, Verification and Validation

To verify that our device meets our use case requirements for physical design features such as weight, temperature and material, we will use a few simple measures. For the weight requirement, we will verify this by weighing the earring with the casing, microcontroller, screen and battery source fully assembled. To verify that the device temperature meets our specifications, we will utilize the STM32's onboard temperature sensor. Finally, to ensure that our device material is appropriate for contact with skin, we will only use piercing/surgical grade metals for those parts of the device and verify user comfort through the surveys administered to our "beta testers", which will be described in greater detail below. We will divide the rest of the testing into two groups:

A. Tests for Setup Time and Update Speed

This group will be brought in and provided with instructions to set up and program the device with a pattern. We will time them as they walk through this process and time how long is spent on each individual stage. This will allow us to see how consistent connectivity is across many trials and if there are parts in the UI that can be streamlined to improve user experience, we can ascertain that with an exit survey for this set of trials. For this trial, we would like to test with at least 10 people, ideally a few more, to ensure the consistent functionality of our mobile application and BLE connectivity.

B. Tests for Battery Life and Comfort

This group will be asked to wear the device for several hours to test the battery life. They will be provided with fully charged earrings at the start of the trial and will report the time at which the battery runs out. The software will include tracking of the battery status to verify this as well. This test serves multiple purposes in that these users will also be able to provide feedback on the comfort and wearability of the device, which we will collect as an exit survey. This group of testers will be significantly smaller than the previous testing group, maybe 4-5 people.

VIII. Project Management

A. Schedule

See Schedule in Appendix C

B. Team Member Responsibilities

Shize Che is primarily responsible for the Hardware Design and firmware subsection of the project. This will involve designing, prototyping and testing circuitry, as well as PCB development with Fusion360. Additionally he will work on designing and developing the firmware libraries for the display peripherals, SPI and BLE Communication.

Madi Davis is primarily responsible for the Mobile Application Interface section of the project. This will entail designing the application functionality and UI/UX, front-end development and implementing modules for data storage and BLE Communication.

Saniya Singh will be working across both the Hardware/Firmware and Mobile Application subsections. She will collaborate with other team members to design and test circuitry, firmware libraries, UI/UX and ios development. Having a thorough understanding of all working parts of the design system will enable Saniya to oversee system integration at the back-end of the project.

All Team Members will be involved in the device enclosure subsection. This will involve researching and ideating potential earring designs, prototyping & user testing, CAD Modelling with Blendr and Assembly of the final product.

C. Bill of Materials and Budget

Part	Manufacturer	#	Cost
Nucleo Dev Board for STM32WB55	Digikey	2	$83
LCD Screen	Digikey	1	$38
DotStar LED Pixel Matrix	Adafruit	2	$26

D. Risk Mitigation Plans

There are two main risk factors involved in the completion of the project. The first is the PCB not working as expected after being manufactured. To avoid potential PCB error, we will develop the hardware prototype in parallel with the PCB design. We believe the development of hardware will guide us to a correct PCB layout. The second is the BLE communication between the microcontroller and user phone. To address this risk, we will consult the example code provided by the HAL library. Numerous BLE example applications can be found in the library.

IX. Related Work

There's a handful of other projects we found that provided similar capabilities to what we want to achieve with our device. The first and the simplest is the HALO-90 open-source earrings project. The HALO-90 earrings use a ring of single-color LEDs on a custom PCB for their design. The PCB contains a STM8L series microcontroller and an onboard battery. The device must be programmed over a wired connection and the earrings react to music. There is one more open source project that we found relevant to our work as it uses the same LED matrix component for a similar application. The project is titled DotStar Fortune Necklace: it is a digital necklace that uses another Adafruit component for Bluetooth connectivity and a large 3.7 V Lithium Polymer battery that are encased in a large rectangle that hangs at the back of the neck while the matrices acts as the necklace pendant and hangs at the front connected by a wire.

X. Summary

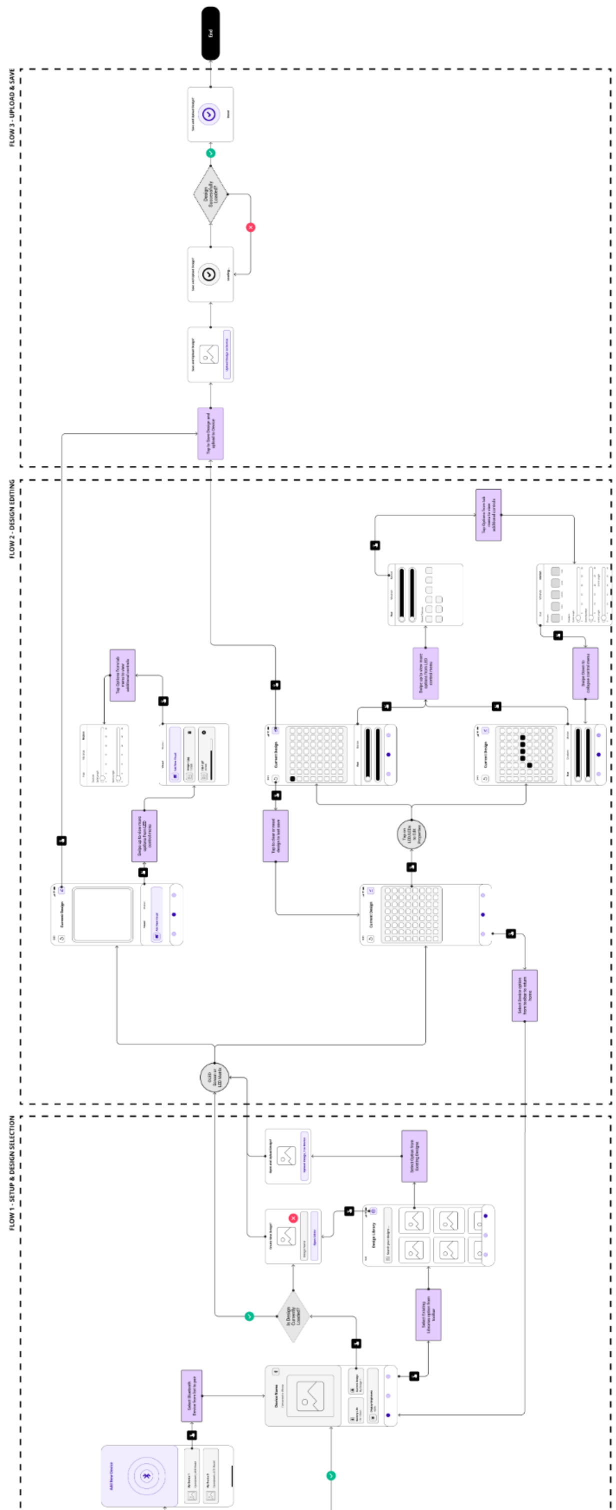
CyberJewelry is a digital earring that aims to be developed into a highly customizable and aesthetic wearable for fashion forward individuals. Our design requirements ensure aesthetics, customizability, and wearability of our device. The implementation plan effectively addresses the design problems to maximally guarantee the proposed design requirements. And the plan for project management will ensure the successful completion of the project as a team.

Glossary of Acronyms

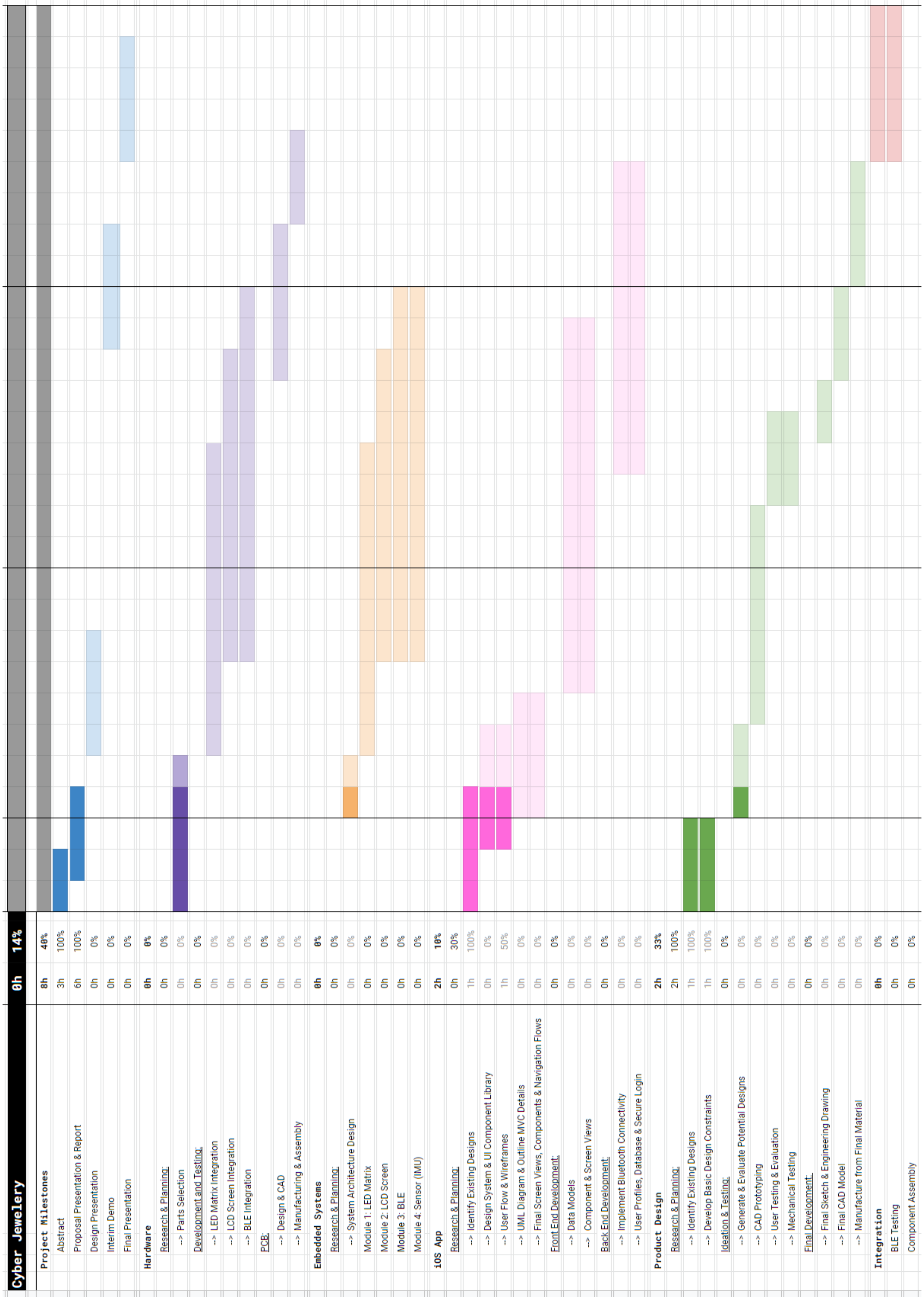
MQTT – Message Queuing Telemetry Transport
OBD – On-Board Diagnostics
RPi – Raspberry Pi
CB - iOS CoreBluetooth Framework
PLA - Polylactic Acid
PCB - Printed Circuit Board
STM - ST Microcontroller

References

[1] G, Charlyn. "DotStar Fortune Necklace with Bluetooth and Touch." Adafruit Learning System, Adafruit, https://learn.adafruit.com/dotstar-fortune-necklace.
[2] Joanna Izdebska,"1 - Printing on Polymers: Theory and Practice", Editor(s): Joanna Izdebska, Sabu Thomas, Printing on Polymers, William Andrew Publishing, 2016, Pages 1-20, ISBN 9780323374682, https://doi.org/10.1016/B978-0-323-37468-2.00001-4. (https://www.sciencedirect.com/science/article/pii/B9780323374682000014)
[3] Thyssen, Jacob P, and Torkil Menné. "Metal allergy--a review on exposures, penetration, genetics, prevalence, and clinical implications." *Chemical research in toxicology* vol. 23,2 (2010): 309-18. doi:10.1021/tx9002726x, pp. xxx-xxx, Abbrev. Month, year, doi: 10.1109.XXX.1234567.
[4] "Densities of Metals and Elements Table," Materials, Engineer's Edge, https://www.engineersedge.com/materials/densities_of_metals_and_elements_table_13976.html



Appendix A: Application Wireflow Diagram



Appendix B: Gantt Chart