ECE 445 Project Design Document

Shower Music Controller

Team Members: Shalin Joshi (shalinj2), Amar Patel (amarcp2), Varnith Aleti (valet3)

Date: 25 February 2026

Project #:17

TA: Eric Tang

Professor: Craig Schultz

1. Introduction

1.1 Problem

Many people enjoy listening to music in the shower to boost their mood or relax after a long day. However, bringing a phone into the shower to control the music can cause damage from water and steam. Existing solutions to this problem involve placing the phone into a protective, waterproof case, but these can be inconvenient because they inhibit the use of the touch screen and often are not very durable. Additionally, trying to operate a phone with wet hands is difficult and raises the risk of dropping and damaging it.

1.2 Solution

Our solution is a shower music controller that can be stuck to the wall and connects to an app on a phone via Bluetooth.

The controller will have a screen to display the user's playlists and songs, along with D-pad buttons to navigate the screen, which will provide more reliability than a touch screen. Additionally, the device will have buttons to play, pause, skip, and control volume to allow for full control of the music. The device will also be powered using AA batteries enclosed in a waterproof enclosure to prevent any open ports into the device.

In order to waterproof our device, we will first 3D print an enclosure for the microcontroller, screen, batteries, and buttons. This enclosure will not be waterproof, and instead, we will create a custom plastic container that will be molded to cover the screen and other water-sensitive components. This container will be the primary component that is attached to the shower walls using suction, and our device will be able to be inserted into this container. The buttons will be exposed, but to prevent water damage, they will be covered using a silicone membrane.

The mobile app will connect to the controller using Bluetooth and call the Spotify API to transmit and retrieve all the information necessary to operate the controller. The user can connect their Spotify account in the app to have access to all of their playlists and songs on the controller.

1.3 Visual Aid

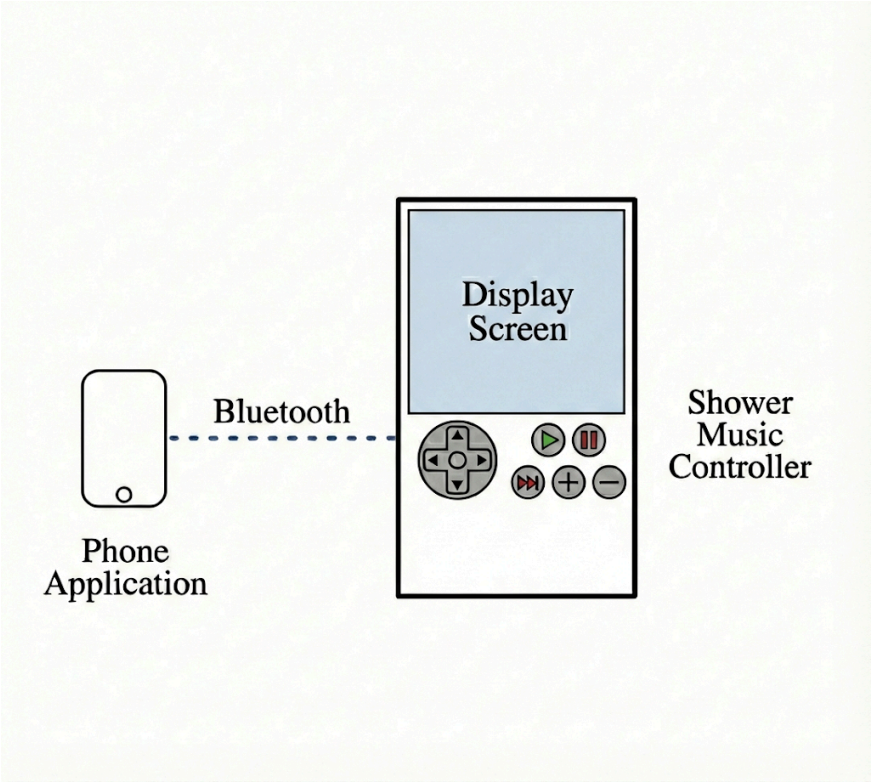

Figure 1. Device Visual Aid

1.4 High-level Requirements List

- The user can successfully perform different playback actions on the controller with a maximum 1 second of delay: Play/Pause, Next Track, Volume Up/Down
- The controller can connect and remain connected through Bluetooth to the phone companion app, and all relevant information from the Spotify API is displayed on the screen
- Controller remains functional after 5 minutes of exposure to shower spray/steam
- Controller operates for at least 2 hours of active use on a full charge

2. Design

2.1 Block Diagram

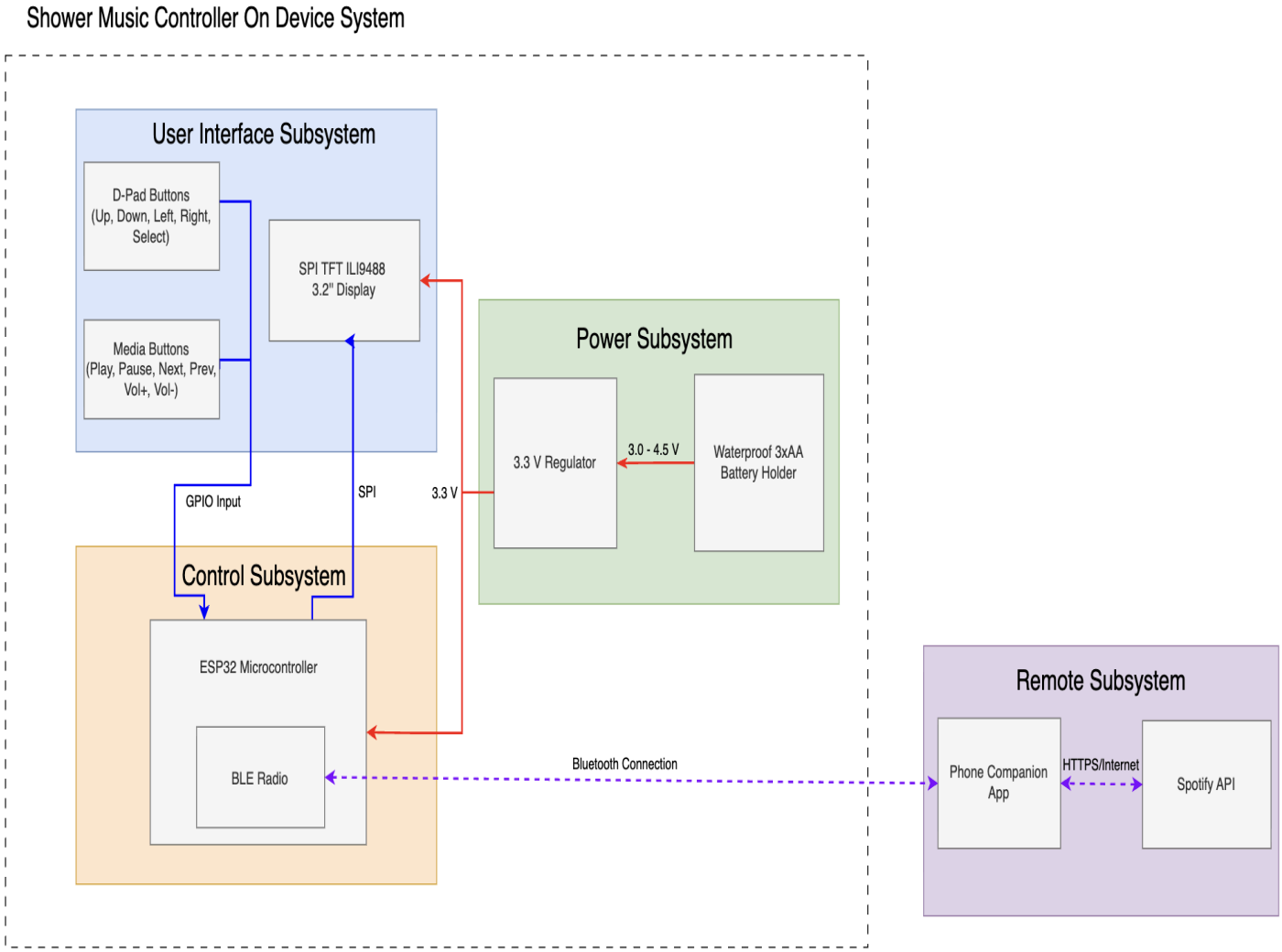

Figure 2. Block Diagram

2.2 Physical Design

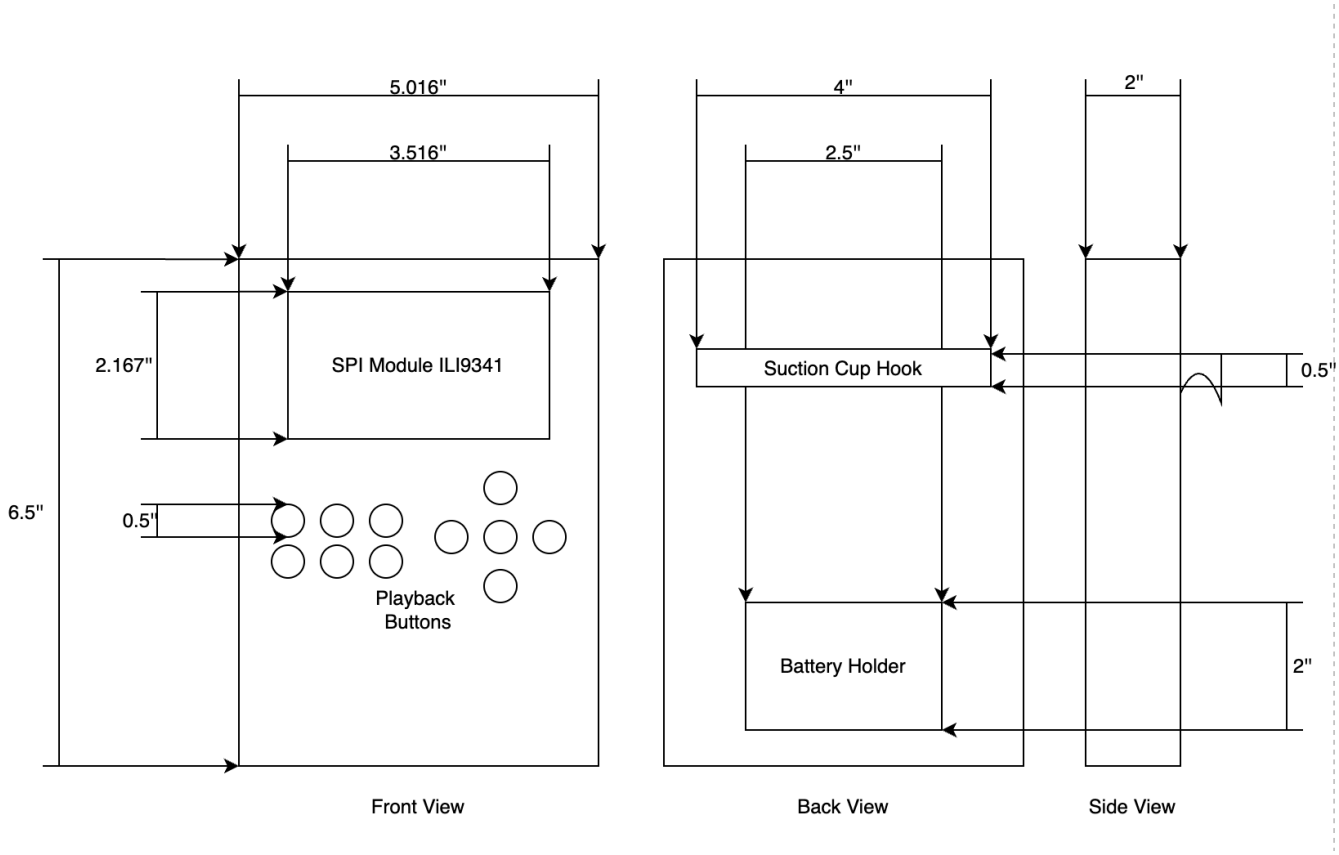

Figure 3. Physical Design Diagram

The physical design of our device is primarily based on a 3D-Printed enclosure, which will hold our different subsystems, like the PCB. The enclosure will be approximately 5 inches wide, 6.5 inches in height, and 2 inch depth. The front side of the device will include the SPI ILI9341 display positioned near the top of the enclosure and an array of playback and menu navigation buttons below it. The screen will be covered with a plastic layer to ensure that it does not sustain any water damage. We will be using waterproof push buttons, which attach to the enclosure using a screw and bolt from the inside to also ensure that no water gets into the enclosure from the shower. The back of our enclosure will include a screwable casing exposing the battery compartment and a U-shaped hook, which will be used to put the device on the shower wall. The device will attach to the wall using suction cups, which include hooks that the enclosure can hook onto.

2.2 Subsystem Overview

Our design is made up of 4 different subsystems, 3 of which are on the physical device, including the power, control, and user interface subsystems, and the last, which is a remote system based on a phone companion app.

2.2.1 Power Subsystem

The power subsystem is responsible for supplying stable electrical power to the electronics in the device. The system uses three AA batteries inside a waterproof battery holder with a built-in power switch as the primary energy source. 3 AA batteries will be able to provide a voltage range of 3-4.5 V, depending on the battery state. The voltage will be regulated to 3.3 V using an LDO 3.3V 600MA regulator, which is the required voltage to power the microcontroller and screen.

Interfaces:

- Input: 3.0-4.5 volts from 3 AA Battery holder
- Output: regulated 3.3V to the control and user interface subsystems at ≥ 500 mA

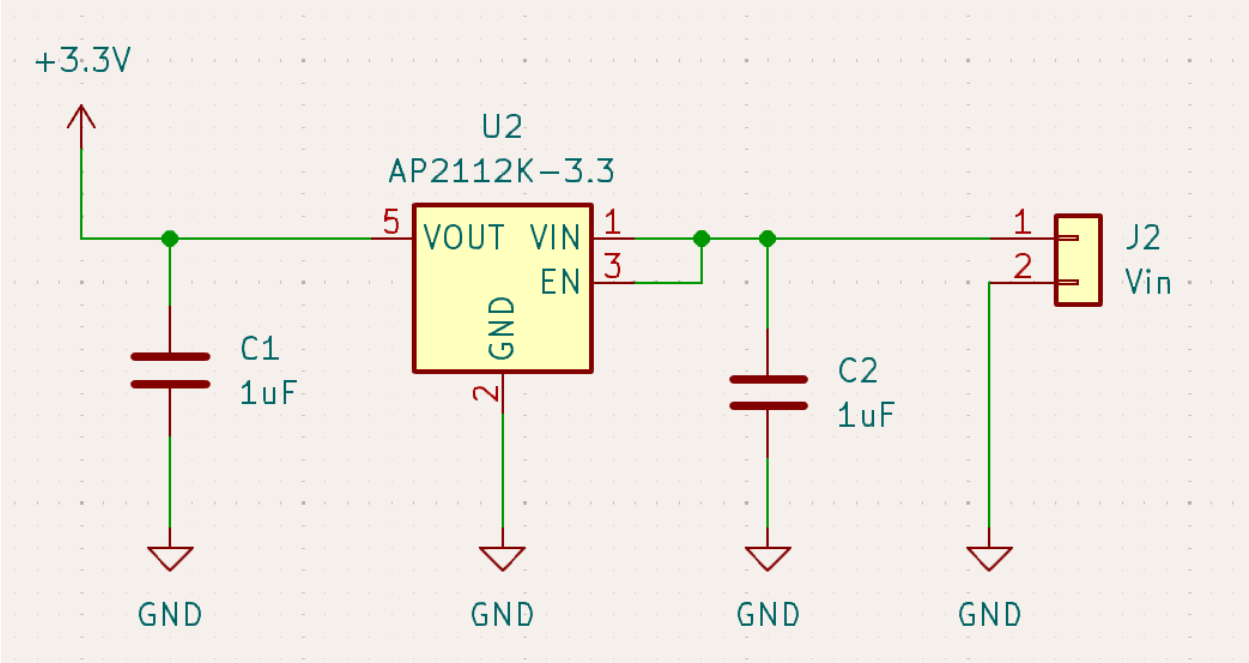

Figure 4. Power Subsystem Circuit

Above is the circuit diagram of our power subsystem. In this diagram, the Vin power supply, which comes from the AA batteries, is fed into the AP2112k-3.3 regulator. This

regulator outputs a regulated 3.3 V rail for the microcontroller and display. 2 1uF capacitors are placed at the input and output terminals of the regulator to provide decoupling.

Requirements	**Verification**
Must accept an input voltage between 3.0 V and 4.5 V	1. Connect a variable DC power supply to the input terminals of the power subsystem. 2. Set the power supply to 3.0 V and then 4.5 V. 3. Use a digital multimeter to measure the voltage at the input pins of the LDO regulator. 4. **To pass:** At each point, the measured value must match the corresponding power supply setting within ±0.05 V
Must continuously supply at least 500 mA at 3.3 V ± 0.2 V to the system	1. Apply a nominal 4.5 V to the input terminals. 2. Connect an electronic load to the 3.3 V output 3. Set the load to draw a constant 500 mA 4. Use a digital multimeter to measure the output voltage every 60 seconds for 5 minutes. 5. **To pass:** All measured output voltages must remain between 3.1 V and 3.5 V for the full duration
Must include an on/off switch	1. Toggle the built-in battery holder switch to the "OFF" position. 2. Measure the voltage at the input of the LDO regulator using a digital multimeter.

	3. Toggle the switch to "ON" and verify voltage presence. 4. Verify the ESP32/Screen powers down completely when "OFF". 5. **To pass:** "OFF" state must measure 0.0 V ± 0.01 V at the regulator input, and "ON" state must measure >3.0 V

2.2.2 Control Subsystem

The control subsystem is the main processing unit of the device, and it is composed primarily of the ESP32-S3-WROOM microcontroller.

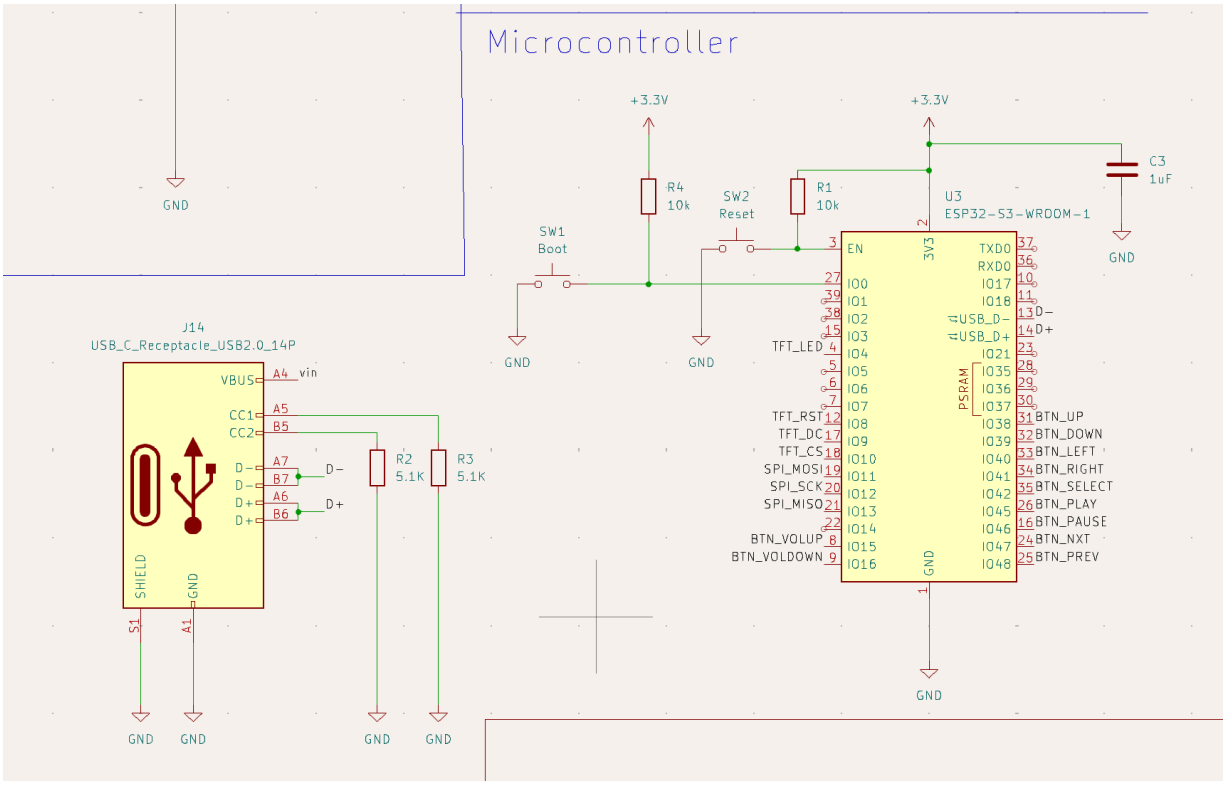

Figure 5. Control Subsystem Circuit

The microcontroller will update the display through the SPI interface, which is connected to the IO pins 8-13 on the circuit diagram. The microcontroller will also read the GPIO signals from the control and playback buttons. We will also need a USB-C input and a

boot and reset button in order to flash code onto the chip. This microcontroller also includes a built-in Bluetooth module, which will be used to wirelessly connect to the remote system phone companion app.

Interfaces:

- Input: 3.3 V from power subsystem, button signals via GPIO
- Output: Display data via SPI
- Wireless BLE link to the phone companion app

Requirements	**Verification**
Must read button inputs and update the display within ≤ 400 ms	1. Connect a 2-channel oscilloscope with the first probe to the D-Pad "Enter" button GPIO pin and the second probe to the SPI chip select (CS) or data (MOSI) line. 2. Trigger the oscilloscope on the falling edge of the first probe. 3. Measure the time between the trigger and the first burst of SPI data sent to the display. 4. Repeat for 5 consecutive presses. 5. **To pass:** All 5 measurements must be ≤ 400 ms
Must maintain a BLE connection to the phone during use	1. Pair the ESP32 with the companion phone app. 2. Initiate a log in the app that pings every 15 seconds. 3. Simulate active use by toggling D-Pad buttons once per minute. 4. Monitor the Arduino serial monitor for "BLE Disconnected" flags or app-side dropouts. 5. **To pass:** 0 disconnected signals over a 15 min period
Must send commands and receive updates within ≤ 2 seconds	1. Press the play button on the device 2. Record the timestamp of the outgoing BLE command packer from the serial monitor

	3. Record the timestamp when the ESP32 receives the corresponding update from the phone and calculate the difference 4. Repeat for different buttons on the device 5. **To pass:** Average of all differences must be less than 2 seconds

2.2.3 User Interface Subsystem

The User Interface Subsystem contains the physical and visual interfaces for the device. It consists of a 3.2inch SPI Module ILI9341 display, which will show menus, playlists, and playback information. The display receives graphical data from the microcontroller via SPI and updates the screen based on the user's button controls.

Figure 6. 3.2inch SPI Module ILI9341

This subsystem will also include a D-pad for navigation and media control buttons for playback. The buttons will allow the user to navigate the menus on the screen and to perform common playback controls while listening to music. We will use 11 specific waterproof push buttons, which are shown in Figure 7. The buttons will be the 4 movement arrows, a select, play, pause, previous song, next song, volume up, and volume down buttons. These buttons will be attached to the PCB board and attached to the enclosure using a screw and a hex nut, as shown in the picture. These buttons can also pose an issue related to debouncing, but we will handle that through software.

Figure 7. Waterproof Push Button

The microcontroller will send signals to the screen in order to show a very simple UI where users can navigate different menus to see what song is being played, their playlists, and their liked songs. Figure 8 shows a very simple sample UI design, which may be changed later.

Figure 8. Sample UI for device

Interfaces:

- Input: 3.3 V from power subsystem, display data from the microcontroller via SPI
- Output: Button signals to the microcontroller via GPIO

Requirements	**Verification**
Must register and transmit unique GPIO signals for each button input	1. Connect a digital multimeter to the GPIO output of the board 2. Press each button one by one 3. Verify that the corresponding pin goes to the correct logic state 4. **To pass:** All buttons must produce the correct logic transition when pressed
Must update display within ≤ 1 second of a button input	1. Use a high-speed camera or smartphone camera to record the screen while pressing an input button

	2. Review the recording and confirm the screen updated within 1 second of the button press 3. Repeat for 5 button presses 4. **To pass:** Screen updates within 1 second for all 5 button inputs

2.3.4 Remote Subsystem

The Remote Subsystem consists of the phone companion app and the Spotify Web API. The phone app acts as the bridge between the device and Spotify, as it handles the user authentication and playback for music control. The app communicates with the device using Bluetooth and translates the button commands into the Spotify API, which is done through HTTPS/Internet. The app will also communicate back to the device so that the display can be updated. Figure 8 shows a basic UI mockup of the app, which will remain very simple and easy to use. The main features of the app will be to connect to the users spotify account, connect to the device via bluetooth and send update messages to the device.

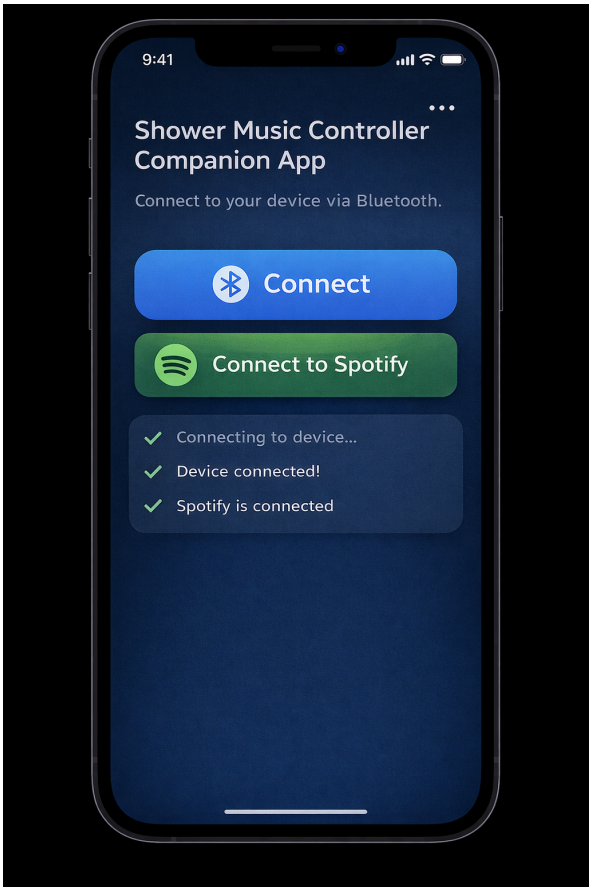

Figure 8. Phone Companion App UI Idea

Interfaces:

- BLE connection to the Control Subsystem
- HTTPS/Internet connection to the Spotify Web API

Requirements	**Verification**
Must translate a BLE command to a Spotify API request and execute the action within 2 seconds.	1. Open the Spotify Desktop app on a PC to visually monitor playback status. 2. Simultaneously, press "Skip" on the ESP32 hardware. 3. Using a stopwatch, measure the time from the physical button press until the track changes on the Spotify Desktop app. 4. Repeat for 10 trials to account for network variability. 5. **To pass:** Average latency must be less than 2 seconds.
Must successfully execute play/pause, skip, volume control, and selection.	1. Trigger each command by pressing corresponding button on device 2. Verify corresponding change on Spotify 3. **To pass:** All functions must execute correctly on Spotify

2.4 Tolerance Analysis

A risk for this device is that it might not be able to have enough battery power to last the required runtime more than 2 hours of operation. We want to make sure that we do not underestimate the total current draw which would lead to a short battery life and burnouts. Below we will highlight the calculations required to figure out the battery runtime for our device.

Known Values:

- Battery Pack
    - Capacity(C): 2850 mAh
    - Nominal Voltage: 1.5 V per AA = 4.5 V total
- Voltage Regulator
    - Output voltage: 3.3 V
    - Max output current: 600 mA
    - Dropout voltage (max): 0.4 V at 600 mA
- ESP32-S3-WROOM Microcontroller
    - Receiving current: 95-97 mA
    - Transmitting current: 285-355 mA
- 3.2’’ SPI TFT (ILI9488) display
    - Working current: 90 mA

Using these known values we can calculate the total system current for 3 different use cases.

Case 1: Showing UI and maintaining wireless connection

- 97 mA from the ESP32 and 90 mA from the display. This means that the total current is 187 mA.

Case 2: ESP32 transmitting at high current with a high power radio state with display running

- 355 mA from the ESP32 and 90 mA from the display. This means that the total current is 445 mA.

Using these total current values, we can estimate the battery runtime of the device using: Battery life (hours) = Battery capacity (mAh) / Current draw (mA)

The battery capacity is 2850 mAh, which represents how much charge the AA battery pack can supply over time.

For Case 1 (normal UI and wireless connection):

- Total current = 187 mA
- Battery life = 2850 mAh / 187 mA = 15.24 hours

For Case 2 (high radio transmit power with display running):

- Total current = 445 mA
- Battery life = 2850 mAh / 445 mA = 6.40 hours

Even in the worst-case scenario, the device can operate for over 6 hours, which is well above the required 2-hour runtime.

Next, we consider the voltage regulation limit. The AP2112K-3.3 regulator requires the input voltage to be at least: 3.3 V + 0.4 V = 3.7 V

Since the battery pack uses 3 AA batteries in series, the minimum average voltage per battery is: 3.7 V / 3 = 1.23 V per battery

If the battery voltage drops near this level, the regulator may no longer maintain 3.3 V, which can cause the ESP32 to reset or the display to become unstable. This shows that the main risk is voltage drop under high current, not total battery capacity. However, since the calculated runtime exceeds the requirement, the design is feasible for the intended use.

3. Cost and Schedule

3.1 Cost Analysis

The total cost for this project can be split into how much labor and parts cost individually. The total cost of labor is estimated using the average annual earnings of an Electrical and Computer Engineering graduate from UIUC which is around $105,000 per year which comes out to be around $50.48 per hour. As each team member works an average of 10 hours per week for 16 weeks, and applying the standard 2.5x multiplier for overhead and benefits our cost of labor works out to a total of $60,576.00 for all three team members and $20,192.00 per member :

$$\$50.48 * 2.5 * (16 \text{ weeks} * 10 \text{ hours/week}) = \$20,192.00$$

We have 3 members in our group:

$$3 \text{ members} * \$20,192 = \$60,576.00$$

Then for the parts, covering all electronic components, custom PCB fabrication, 3D printing, and mechanical hardware required to build the Shower Music Controller. The parts will come out to be around $81.22

3.1.1 Labor

Name	$/hr	Hours/week	Weeks	Total Cost
Shalin Joshi	$50.48	10	16	$20,192.00
Amar Patel	$50.48	10	16	$20,192.00
Varnith Aleti	$50.48	10	16	$20,192.00
Total				**$60,576.00**

3.1.2 Parts

Description	Manufacturer	Part Number	Qty	Unit Cost	Total Cost
ESP32-S3-WROOM	Espressif	ESP32-S3-WROOM-1-N16	1	$6.50	$6.50
1uF 0603 Capacitor	YAGEO	C0603C105K9PAC7867	3	$0.10	$0.30
10KΩ 1% Resistor	Stackpole	RMCF0805JG10K0	1	$0.10	$0.10
5.1KΩ Resistor	YAGEO	CFR-25JT-52-5K1	2	$0.10	$0.20
3.2” SPI TFT Display (ILI9341)	Hosyond	ILI9341-3.2-SPI	1	$16.99	$16.99
Tactile Push Buttons (D-Pad & media)	Twidec	PBS-33B-BK-X	2	$9.99	$19.98
AP2112k-3.3 Voltage Regulator	Diodes Inc.	AP2112K-3.3TRG1	1	$0.22	$0.22
AA Battery Holder (3xAA) Waterproof, On/Off Switch	Jex Electronics	BHWAA3	1	$4.99	$4.99
Suction Cup Mounts	KPPTYTY	N/A	1	$6.59	$6.59
3D Printing Filament (PLA)	OVERTURE	N/A	1	$13.98	$13.98
AA Batteries (Energizer, 3-Pack)	Energizer	E91BP-3	2	$4.99	$9.98
Boot/Reset Button	C&K	PTS645SL43SMTR92 LFS	2	$0.36	$0.72

USB-C Port	GCT	USB4085-GF-A	1	$0.87	$0.87
Total					**$81.22**

3.1.2 Sum of Costs

Category	Cost
Total Labor	$60,576.00
Total Parts	$81.22
Total	**$60,657.22**

3.2 Schedule

Week of	Task	Person
1/19	Come up with project idea and problem it solves	All
	Initial concept for shower music controller	Shalin
1/26	Draft high level requirements and architecture layout	Varnith
2/2	Finalize system architecture and subsystem interfaces	Shalin
	High level requirements and performance targets	All
2/9	Select components (ESP32, display, buttons, battery, ect.)	All
2/16	Complete full schematic	Varnith
	Begin PCB layout	Shalin
	Begin enclosure concept and waterproofing strategy	Amar
2/23	Finish PCB layout	Shalin
	Finalize parts to order	Everyone
3/2	Revise PCB based on design review feedback	Amar
	Submit second PCB order (if needed)	Everyone
	Breadboard prototype: ESP32 + TFT display (SPI)	Shalin
	Research for Companion app: Spotify API OAuth requirements & BLE communication	Amar

3/9	Breadboard Demo with Instructor and TA	Everyone
	Submit 3rd Round PCBway Order (if needed)	Varnith
	Button GPIO + D-Pad menu navigation on display	Shalin
	Companion app: Spotify playback API integration	Amar
3/16	SPRING BREAK	
3/23	Submit 4th Round PCBway Order	Varnith
	Test PCB when received, address any hardware issues	Shalin
	BLE <-> app command protocol finalized	Shalin + Amar
3/30	Individual progress reports	Everyone
	Integrate BLE: device <-> app <-> Spotify full flow	Shalin + Amar
	3D print enclosure v1, fit-check all components	Varnith + Shalin
4/6	Progress Demo with Instructor and TA	Everyone
	UI polish: playlists, now playing, volume screen	Varnith + Amar
	App stability improvements, background BLE service	Amar
4/13	Waterproofing: seal enclosure	Varnith
	End-to-end latency testing (target <=1 sec)	Shalin + Amar
	Water/steam exposure test (5 min shower spray test)	Everyone
4/20	[DUE MON-FRI] Mock Demo during weekly TA meeting	Everyone
	[DUE THU-FRI] Mock Presentation with Comm and ECE TAs	Everyone
	Battery life stress test (target >=2 hrs)	Varnith
	Bug fixes and final integration polish	Everyone
4/27	[DUE MON-WED] Final Demo with Instructor and TAs (8:00a-6:00p)	Everyone
	[DUE THU-FRI] Final Presentation with Instructor and TAs (8:00a-6:00p)	Everyone
5/4	[DUE WED] Final papers due 11:59p	Everyone
	[DUE THU] Lab checkout 3:00p-4:30p with TA; Lab Notebook due 11:59p	Everyone
	Best Projects Judging and Award Ceremony (Mon 2:00p, location TBD)	Everyone

3. Ethics and Safety

3.1 IEEE Code of Ethics I.1: Safety, Health, and Welfare

Operating electronics in a high-humidity and wet environment poses a risk of electrical shock and device failure. To prioritize user safety, we will use a non-conductive 3D printed enclosure with waterproof physical buttons. Our design will aim to reach an IP64 rating, which will make sure that the device is protected from any water splashing in any direction. We will be using 3 AA batteries and will ensure to implement proper battery protection circuitry, including overcurrent, overvoltage, and thermal protection. Then we will also make sure the battery holder is waterproof to prevent damage to the batteries. During our development process, we will conduct these waterproofing tests in a controlled environment using simulated environmental conditions before actually being used in an actual shower. The device will include clear user warnings about its limitations that it is designed for shower “splashes” and steam exposure.

3.2 IEEE Code of Ethics I.5: Honest Disclosure

We will be transparent and realistic with our users by stating the limitations of our device. As we are aiming for an IP64 rating, we will inform the users that the device is intended only for water splashing and steam, and not for total submersion or direct water jets. We will make sure to communicate potential performance variances, such as certain waterproofing limitations or Bluetooth signal connections in certain scenarios, to the user in a technical manual.

3.3 ACM Code of Ethics 1.6: Respecting Privacy

As our device will interact with the Spotify Web API, which requires access to user account information, we will collect minimal data. In order to have high standards of privacy, our device will only request the minimum information needed to perform the functions that we described our device would be able to perform, such as control playback and retrieve playlist data. We will also have secure authentication and make sure that the user’s Spotify password is never stored or any other personal information. The companion app will clearly request only needed permissions and provide the users with transparent details of what data is accessed and how it will be used.

3.4 Societal and Accessibility Impact

Our project will aim to contribute to a quality of life improvement for users who enjoy listening to music but face the difficulty of controlling the playback of the music while staying in the shower, with wet hands, and non-waterproof phones. Our device will provide a more accessible interface rather than attempting to use a smartphone in wet conditions for which it was not made, which can cause inconveniences. The device will be able to provide a better interface for these conditions with the physical buttons and dedicated controls required for the playback. By creating a dedicated peripheral for humid and wet environments, we help users avoid the risk of damaging expensive primary smartphones and provide a more convenient option rather than stepping out in the middle of a shower.

4. References

[1] Espressif Systems, ESP32-S3-WROOM-1/WROOM-1U Datasheet, Espressif, 2023. [Online]. Available: https://documentation.espressif.com/esp32-s3-wroom-1_wroom-1u_datasheet_en.pdf

[2] LCD Wiki, 3.2inch SPI Module ILI9341 SKU: MSP3218, LCD Wiki. [Online]. Available: https://www.lcdwiki.com/3.2inch_SPI_Module_ILI9341_SKU:MSP3218

[3] Twidec, Waterproof Momentary Pre-soldered Pushbutton Switch PBS-33B-BK-X, Amazon. [Online]. Available: https://www.amazon.com/Twidec-Waterproof-Momentary-Pre-soldered-PBS-33B-BK-X/dp/B08JHW8BPV

[4] IEEE, IEEE Code of Ethics, IEEE, 2020. [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8

[5] Association for Computing Machinery, ACM Code of Ethics and Professional Conduct, ACM, 2018. [Online]. Available: https://www.acm.org/code-of-ethics

[6] Nature's Generator, AA Battery Voltage: What You Should Know, Nature's Generator Blog. [Online]. Available: https://naturesgenerator.com/blogs/news/aa-battery-voltage

[7] Diodes Incorporated, *AP2112K-3.3 Low Dropout Regulator Datasheet*, 2021. [Online]. Available: https://www.diodes.com/assets/Datasheets/AP2112K.pdf.

[8] Yageo, *CFR-25JT-52-5K1 5.1kΩ Through Hole Resistor*, Digi-Key Electronics. [Online]. Available: https://www.digikey.com/en/products/detail/yageo/CFR-25JT-52-5K1/16674047

[9] GCT (Global Connector Technology), *USB4085-GF-A USB Type-C Receptacle Connector*, Digi-Key Electronics. [Online]. Available: https://www.digikey.com/en/products/detail/gct/USB4085-GF-A/9859662

[10] C&K, *PTS645SL43SMTR92-LFS Tactile Switch*, Digi-Key Electronics. [Online]. Available: https://www.digikey.com/en/products/detail/c-k/PTS645SL43SMTR92-LFS/3861373

[11] Twidec, *Waterproof Momentary Pushbutton Switch PBS-33B-BK-X*, Amazon. [Online]. Available: https://www.amazon.com/Twidec-Waterproof-Momentary-Pre-soldered-PBS-33B-BK-X/dp/B08JHW8BPV

[12] Jex Electronics, *Waterproof 3x AA Battery Holder with Switch*, Amazon. [Online]. Available: https://www.amazon.com/Jex-Electronics-Waterproof-Battery-Holder/dp/B0CXJZD4YY

[13] HiLetgo, *3.2" TFT LCD Display Module (ILI9488 SPI)*, Amazon. [Online]. Available: https://www.amazon.com/dp/B0B1M9S9V6

[14] Generic Manufacturer, *Shower Caddy Suction Cup Replacement Connectors*, Amazon. [Online]. Available: https://www.amazon.com/Shower-Caddy-Connectors-Replacement-Compatible/dp/B0DGT6X5TX