AC-Sequencer

Audio Sequencer with Bluetooth

Group 10:

Brandon Marcoux (CpE) **Giani Francis (Cpe)**

Miguel Chavez (Cpe) **Alexis San Javier (Cpe)**

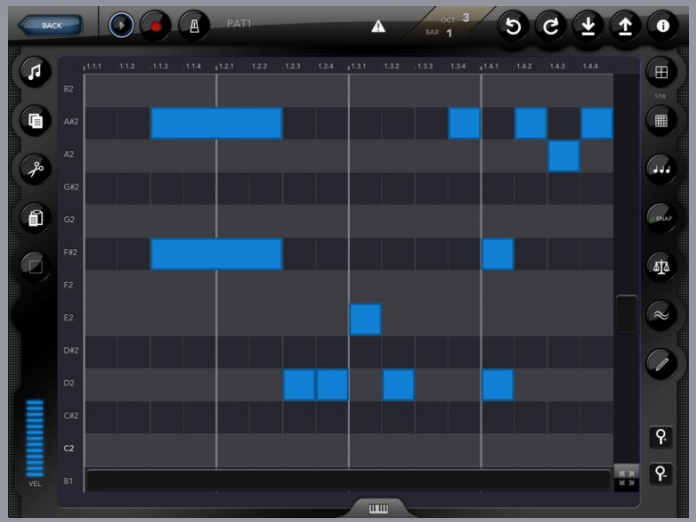
What is an Audio Sequencer?

Project Description

Audio Sequencers

- Program musical sequences.
- Implemented in different format for many different platforms.
- Sounds are assigned to channels, dynamically or statically.
- Output can be a trigger signal or audio.
- Basic components: Start/Stop button, sequencing button array, Tempo and Volume controls.

Audio Sequencers



What do we aim to change?

- Simplify the sequencing workflow
  - Phones in place of computers.
  - Bluetooth sound transfer in place of wired connection or transferable storage medium.
- Improve portability while maintaining the appeal of physical unit.
- Easily upload new sounds
- Lower price point

Goals and Objectives

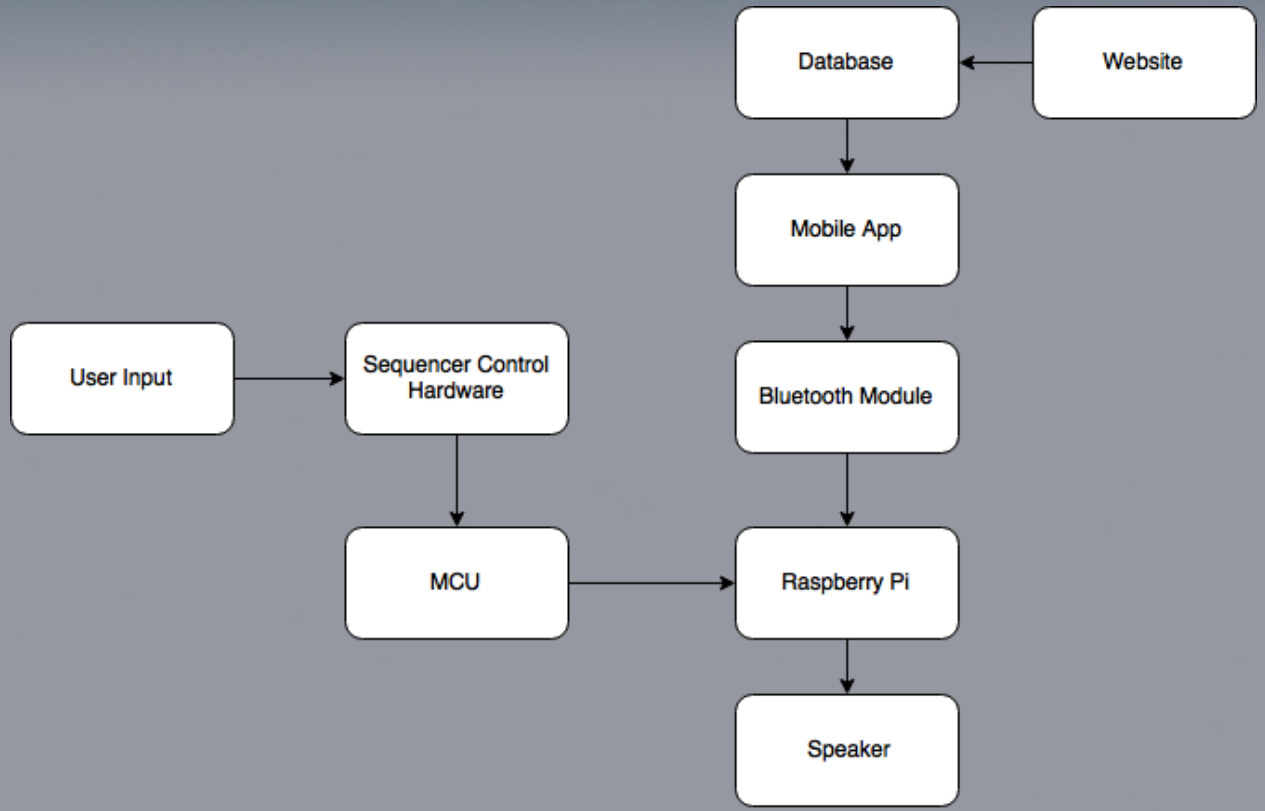
- Main Goal: Create a professional grade product that musicians will prefer to use over other commercially available options.
- Maintain well formatted documentation in order to ensure the project is scaleable and maintainable.
- Design a product that will durable and easy to replicated and build.

System Design

Requirements

- Use Bluetooth to transfer files from Smartphone to Physical Unit
- 8 channels x 16 steps.
- Raspberry Pi which can store and mix 8 sound files simultaneously
- Physical unit with at least one output options.
- Mobile application that allows user to select sound samples from their library and upload them to Physical Unit
- Unit weight of less than 5 pounds.
- Unit can run on battery power or from a power supply (provided with the unit).

System component Diagram



Hardware Design and Implementation

Multiplexing overview

- Necessary due to limited number of pins on MCU.
- Handle outputting signal for LEDs, input signal for buttons.
- Utilizing two 8x8 LED Matrices in an 8x16 array, which makes for 128 effective LEDs.
- Row of 16 red LED’s that shows the steps of the sequence
- 8x16 button array = 128 buttons.
- The MCU does not have 144 GPIO pins.

Button Multiplexing

- The best way we could find to multiplex the buttons was to connect them in rows and columns. That way, when we read which button is pressed, we can detect the row and column and figure out which button has been pressed. All buttons are momentary push buttons.
- With the SN74LS148 one can utilize multiple units and cascade them with a multi NAND gate in order to implement a 16:4 encoder.
- Utilizing one SN74LS148 (8:3 encoding) for the columns and two in a 16:4 configuration will accommodate reading all buttons using only 7 GPIO pins.

LED Multiplexing

- Shift Register implementation (74HC595) vs Decade counter implementation (CD4017B) vs LED Display Driver (MAX7219).
- Based on the number needed and efficiency the MAX7219 was the obvious option.

Part	# Needed to Drive Display	# of LEDs controlled	Efficiency
74HC595	5	8	Low
CD4017B	5	19	Low
MAX7219	3	64	Very

MCU

The microcontroller selection was an important decision to make in the design of our project. We chose the ATmega328 for the following reasons -

- high performance
- low power
- large online community for support during development
- 5V operating voltage
- processing speeds of 16MHz.
- cost effective

Bluetooth Module

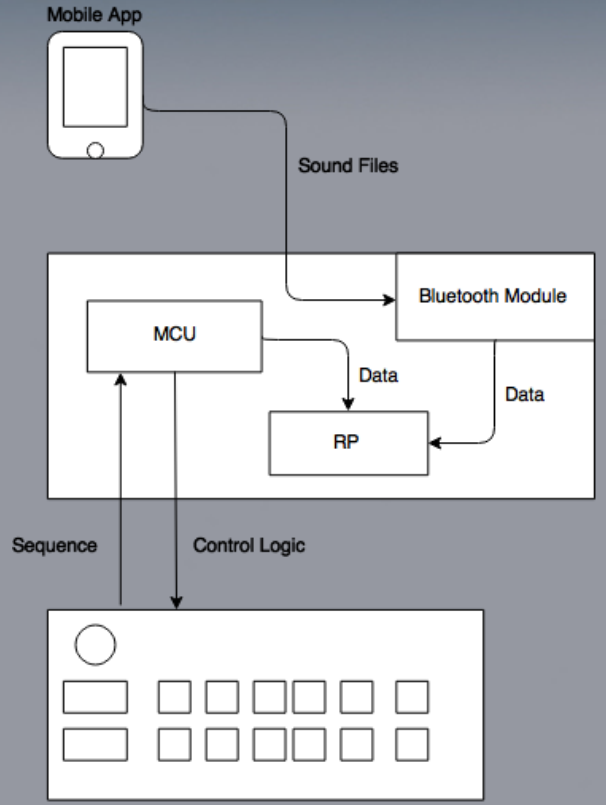
- The Bluefruit UART Friend., Bluetooth 4.0 BLE is required for our wireless design.
- The Bluetooth module will allow us to receive sound files and send sound files wirelessly.
- Permits the transfer of serial data from our mobile application to the Physical Unit

Bluetooth Unit	Operating Voltage (V)	Pins Required	Version	Price ($)
Bluefruit LE	3.3-5	6	4.0 BLE	17
HC-06 RS232 TTL	3.3	2	2.0	7
Bluefruit	3.3	3	4.0 BLE	17

Bluetooth Data Transfer

Data transfer for our project will be done using serial communication. Below is a prototype of the steps taken to control the sequencer from the mobile application.

- Mobile Application will send sounds associated with each row in the button matrix to the Raspberry Pi
- The Raspberry Pi will receive the data via serial communication from the Bluetooth module
- The Raspberry Pi will store the sound files
- The MCU unit will compute the state of the button array
- The MCU will send that button state to the Raspberry Pi which will play the corresponding sounds



i2C

- The MCU will use i2C in order to communicate to the Raspberry Pi
- i2C in our project is responsible for sending the button states to the Raspberry Pi, which tells it the order to play the sounds
- Also, the Raspberry Pi communicates to the MCU via i2C when it is receiving bluetooth data over serial. This tells the MCU to stop sending data while the bluetooth is transferring data from the database to the board.

Playing Multiple Sounds

Requirements needed to test and play multiple sounds are:

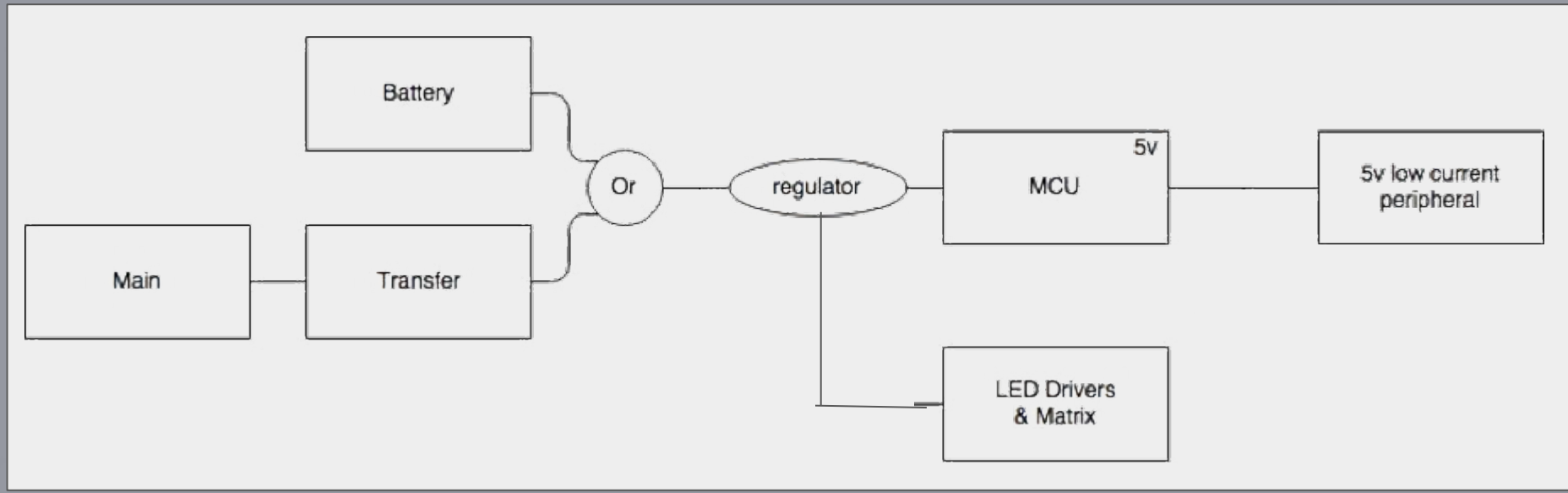
- ATmega328 microcontroller
- Raspberry Pi
- Speaker (at least 8 ohm)
- Summing Solutions
- Python Pygame Mixer library

Enclosure

- Material Property Considerations:
  - Weight
  - Durability
  - Ease of construction
- Materials considered:
  - Wood
  - Plastic (3d printed)

Power Constraints

- All chosen components run on 5 volts.
- A system requirement specifies that we must be able to supply the system from a battery or a wall outlet.

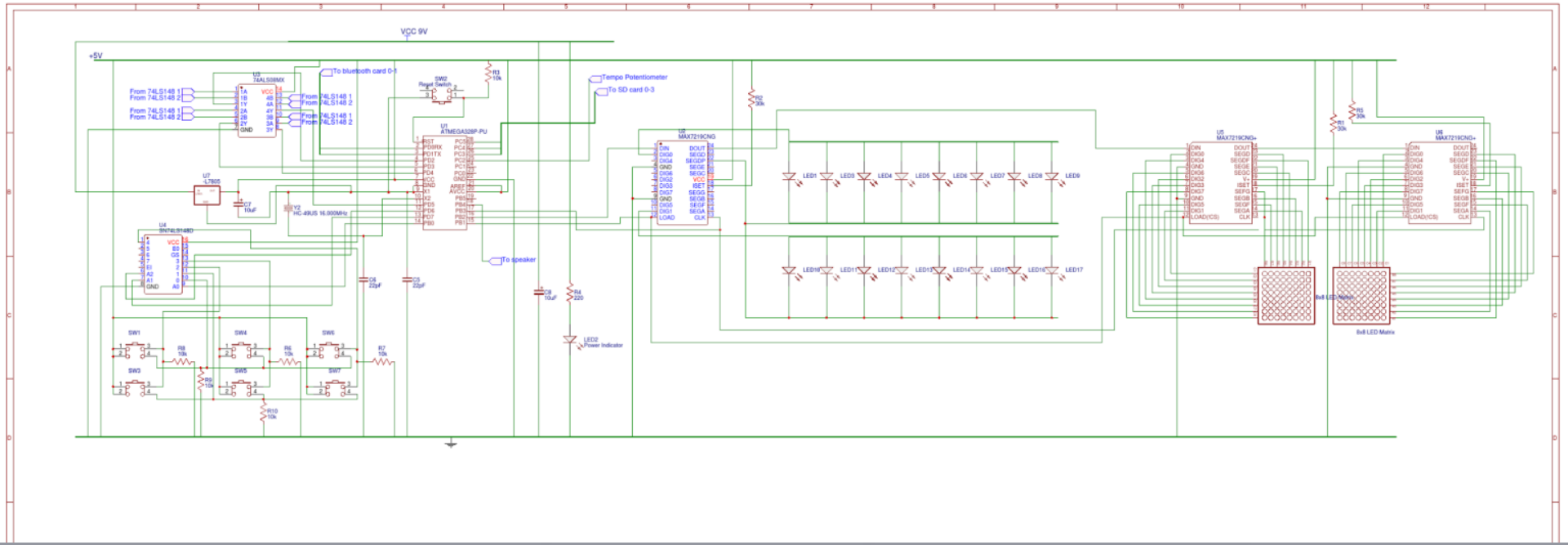


Voltage Regulator Selection

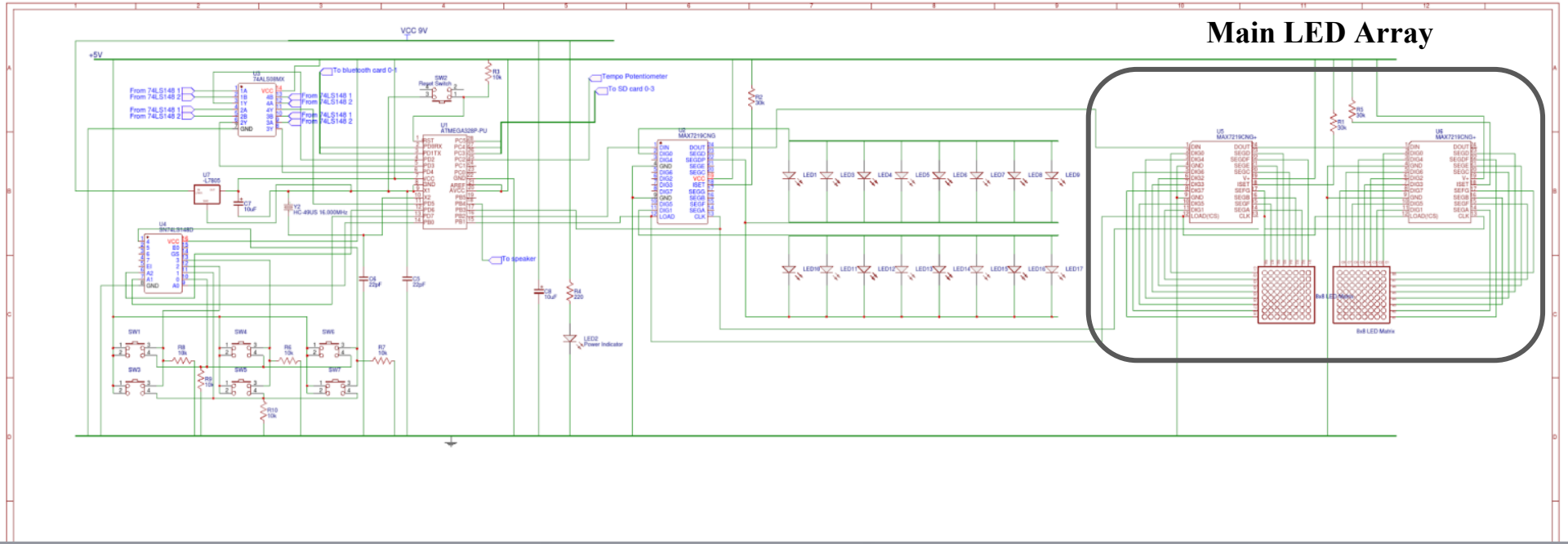
- Decided on L78xx series based on:
  - Price point
  - Consistency
  - We will be able to source the regulator we require from this one series.

Name	Manufacturer	Output Voltage	Input Voltage	Max Current
L78xx	ST Microelectronics	5 – 24V Discrete	7-30V	1A
LM317T	Texas Instruments	1.2 – 37V Adjustable	4.2-40V	1.5A

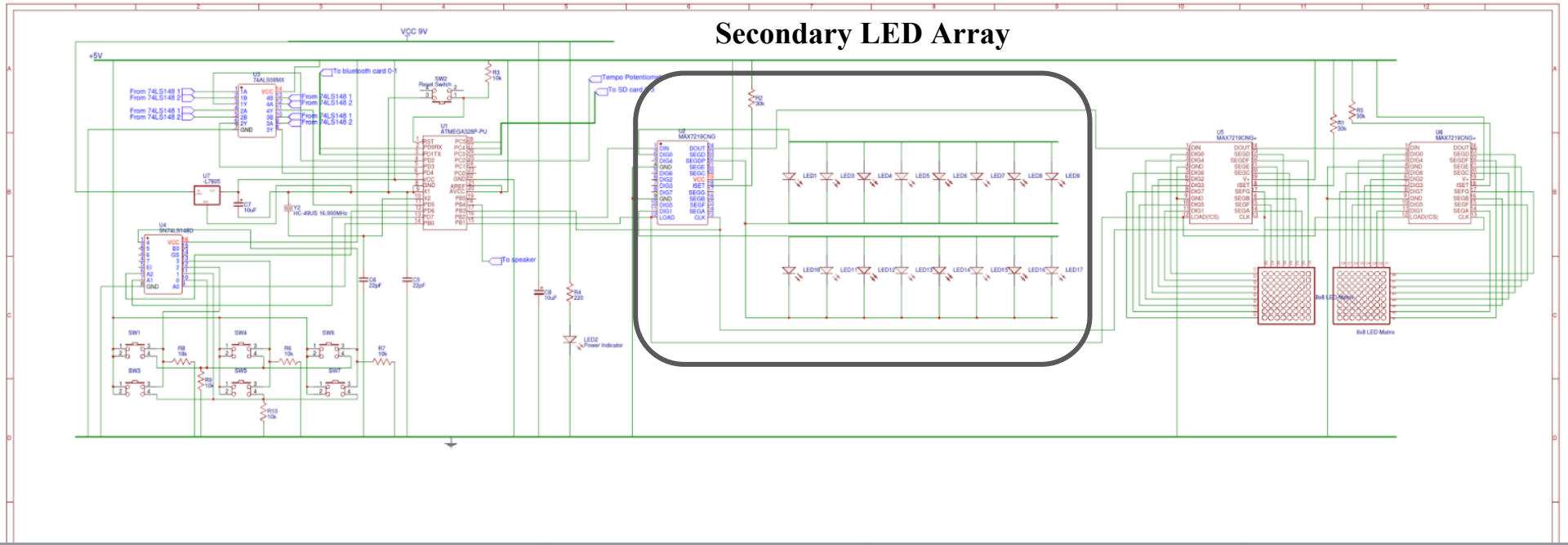
Schematic


VCC 9V
+5V
To bluetooth card 0-1
Tempo Potentiometer
To SD card 0-3
To speaker
LED1
LED2
LED3
LED4
LED5
LED6
LED7
LED8
LED9
LED10
LED11
LED12
LED13
LED14
LED15
LED16
LED17
8x8 LED Matrix

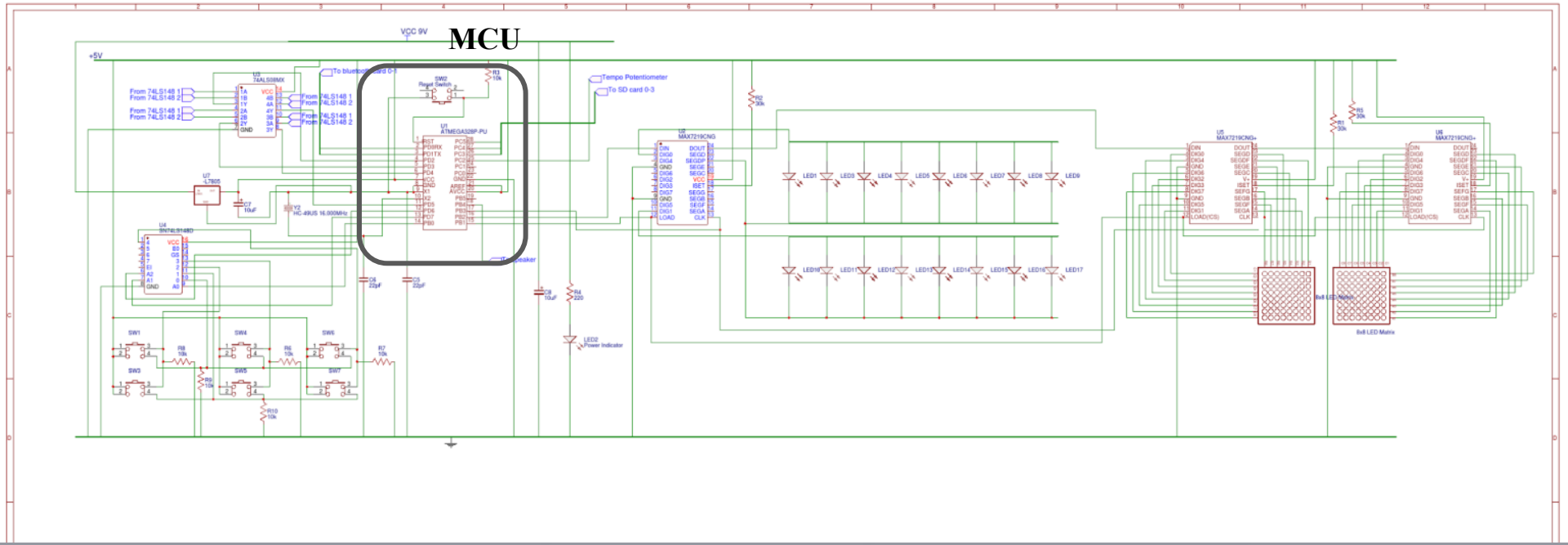
Schematic


Main LED Array
VCC 9V
+5V
To bluetooth card 0-1
Tempo Potentiometer
To SD card 0-3
To speaker
Reset Switch
HC-49US 16.000MHz
8x8 LED Matrix

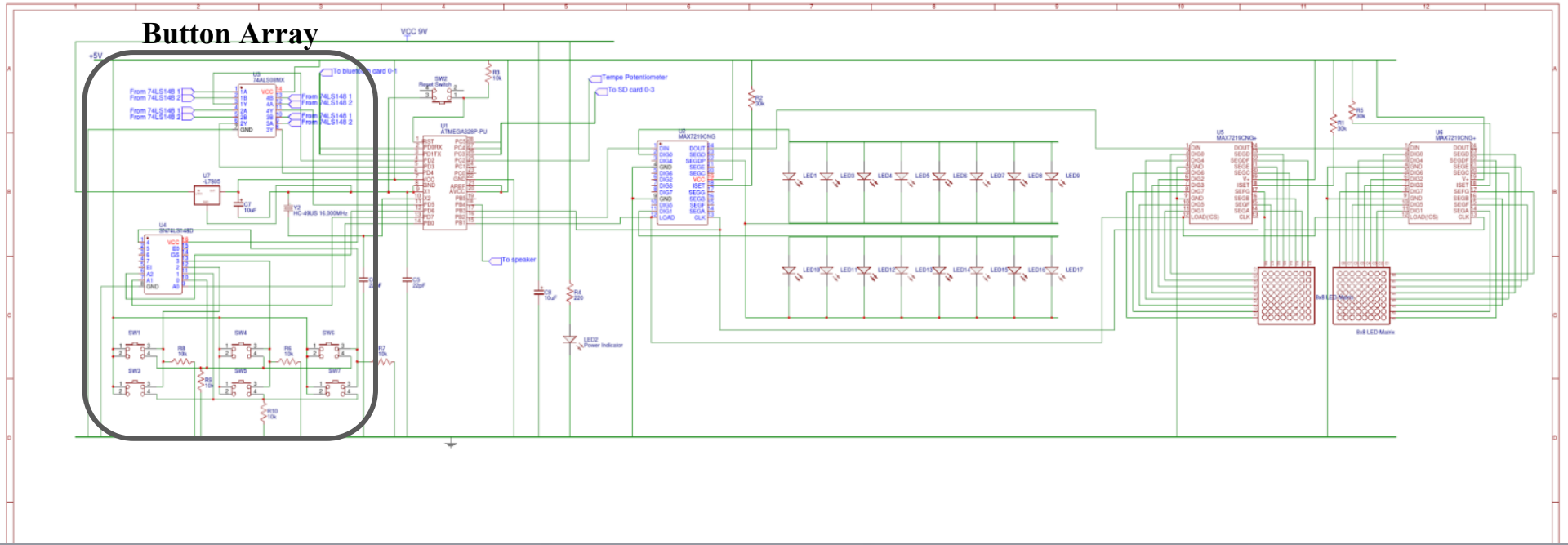
Schematic



Schematic

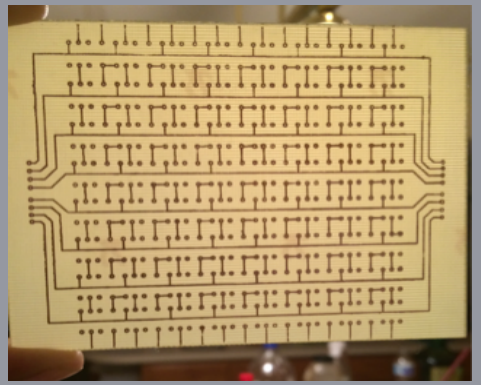

MCU
VCC 9V
+5V
Tempo Potentiometer
To SD card 0-3
ATMEGA328P-PU
MAX7219CNG
8x8 LED Matrix

Schematic


Button Array
VCC 9V
+5V
To bluetooth card 0-1
Tempo Potentiometer
To SD card 0-3
To speaker
LED2
Power Indicator
8x8 LED Matrix

PCB

- Main PCB
  - Houses atmega328 and all chips controlling the LED's, and reading button states
  - Two Layer
  - Purchased from manufacturer
- PCB for Button array
  - Detects a button that is pressed and send a voltage signal
  - Homemade using the Toner method



Software Design and Implementation

System Software Architecture

4 Main Software Components

- Website
- Database
- Mobile Application
- Physical Unit

Database

Google Firebase

SQL vs Firebase

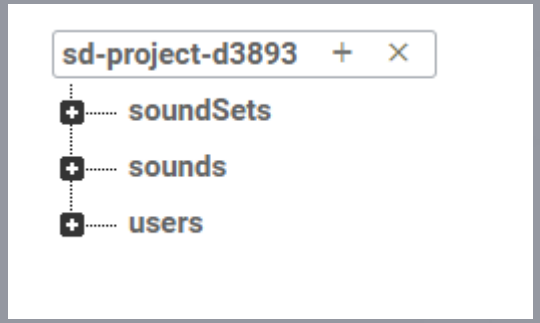
Environment	SQL	Google Firebase
Schema	Static	Dynamic
Team Knowledge	Low	High
Simple Authentication	No	Yes
Server Needed?	Yes	No
Concurrency?	Yes	Yes
Durability?	Yes	Yes

Firebase

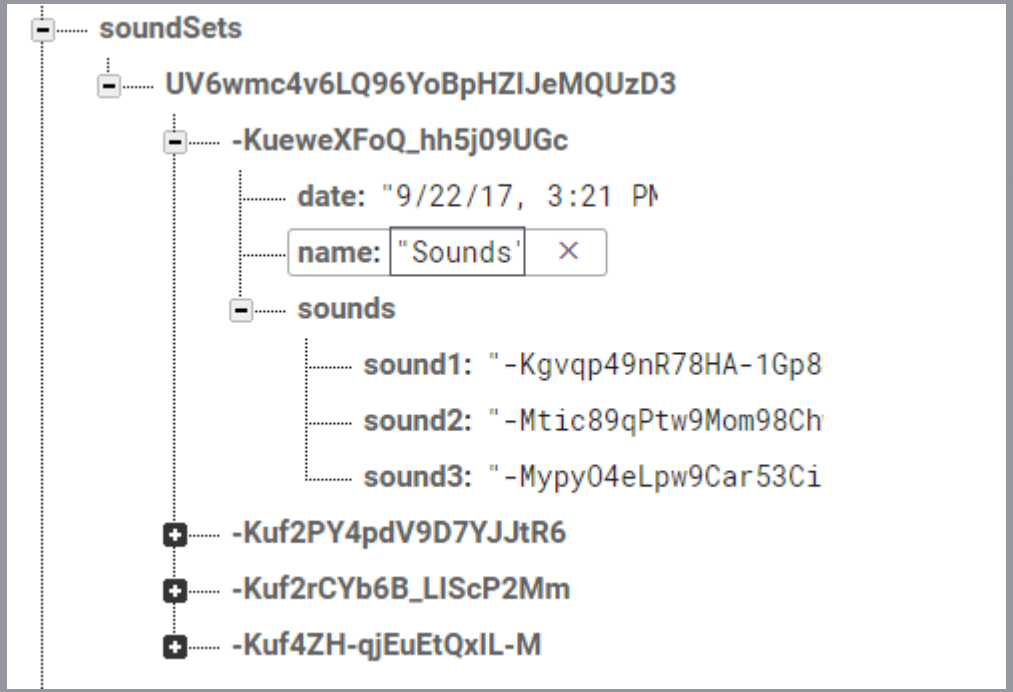
- Simple to use
- API for web and mobile
- Servies
  - Database
  - Authentication
  - Cloud Storage
  - Google Analytics

Firebase layout

- Main Node



- Sound Sets



Firebase layout cont.

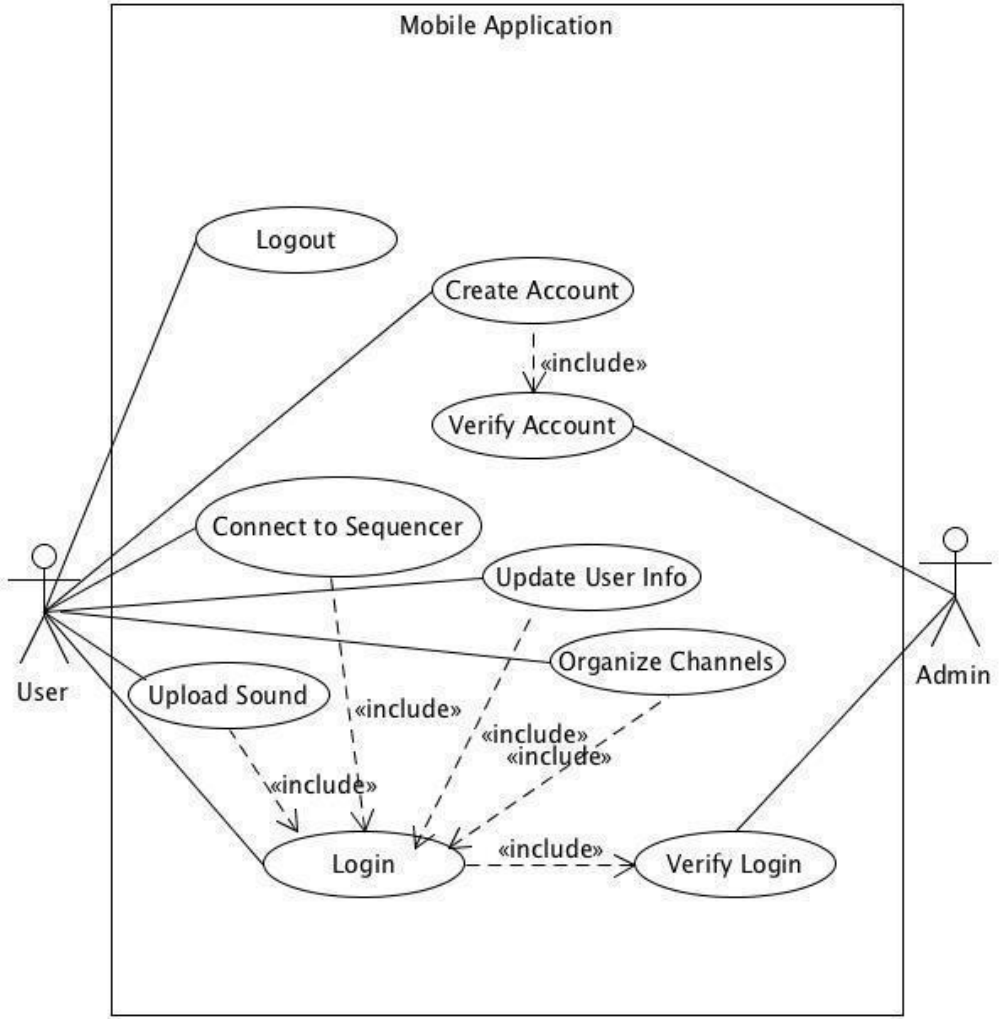
- Sounds

- Users

Mobile Application

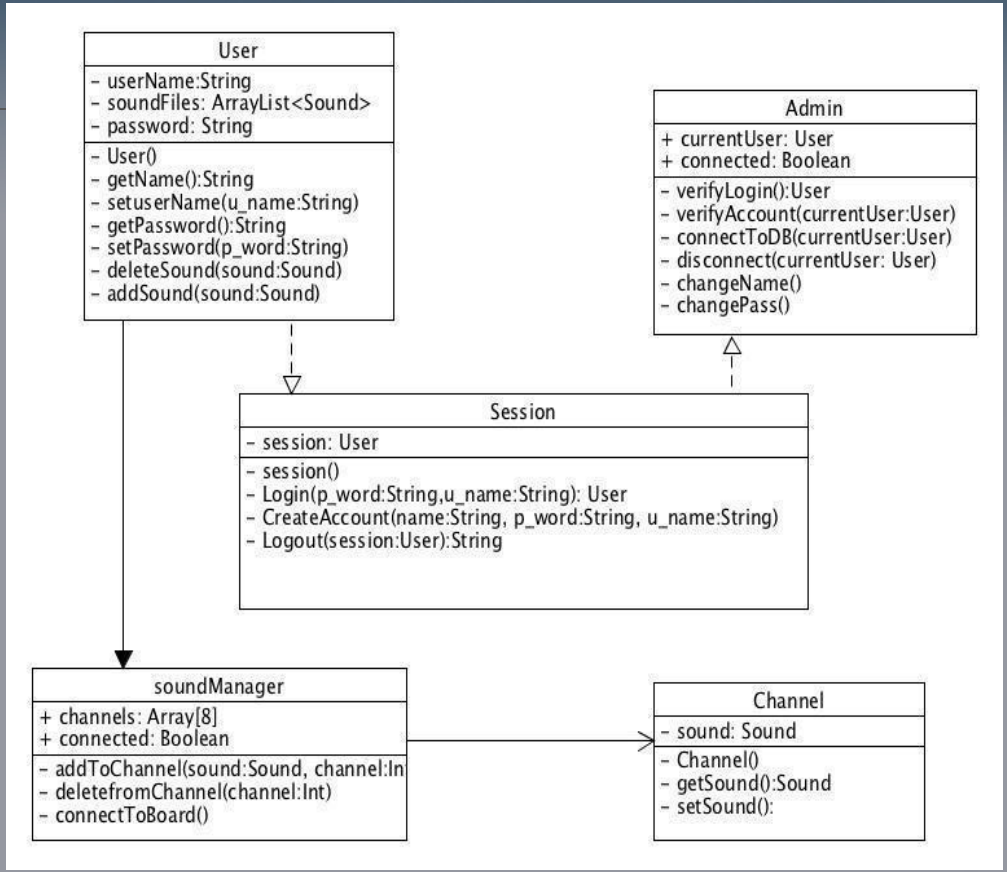
Class Diagram

- User
  - Login
  - Upload Sounds to board
  - Organize Sound channels
- Admin (Firebase)
  - Verifies Account
  - Verifies Login



UML Diagram

- Admin
  - Firebase
- Session
- User
- Sound Manager
- Channel



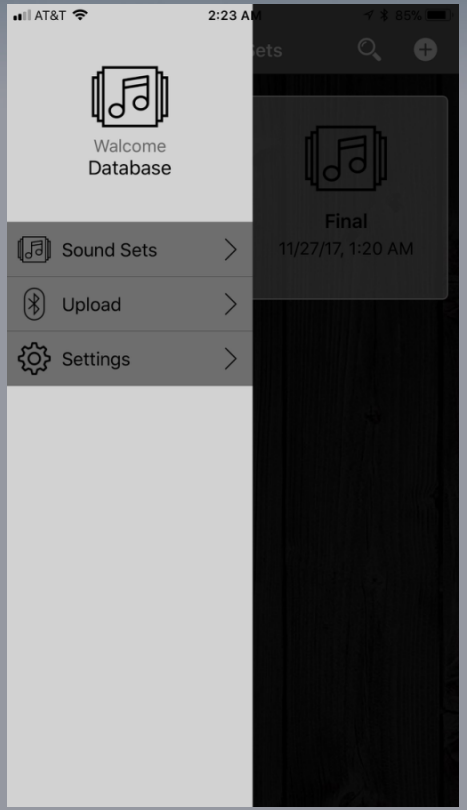
iOS Application Design

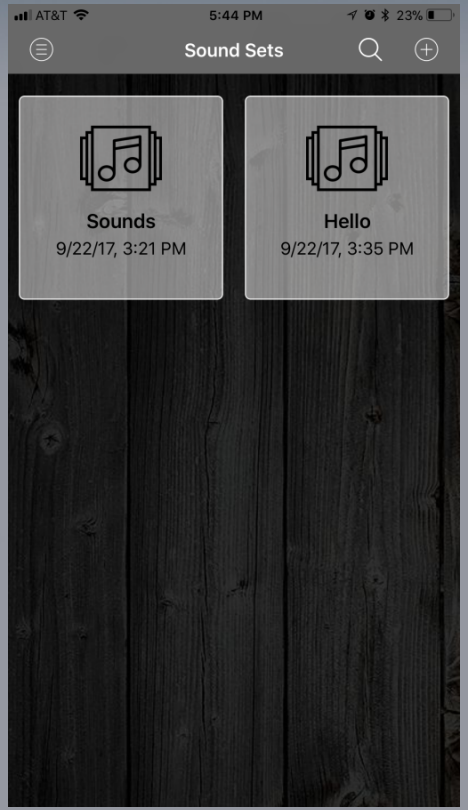
- Application Layer
  - MVC
- Service Layer Architecture
  - Bluetooth
- Data Layer Architecture
  - Repository Pattern

- Libraries
  - UIkit
    - UIViewControllers
    - UICollectionView
  - Lottie
    - Animations
  - Firebase
    - Database
    - Authentication
    - CloudStorage

iOS Application UI

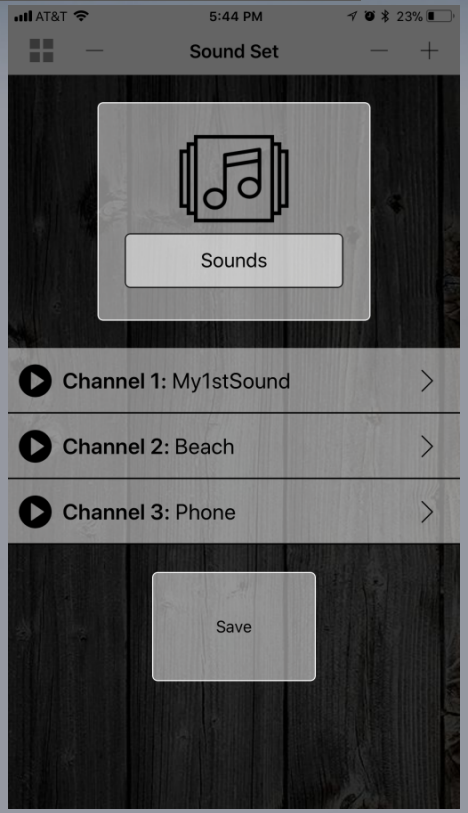
- Menu
  - 3 Sections
    - Sound Sets
    - Upload
    - Settings
- Sound Sets
  - Search through Sound Sets
    - UISearchController
  - Add new Sound Sets
  - Select a Sound Set to Edit
    - UICollectionView

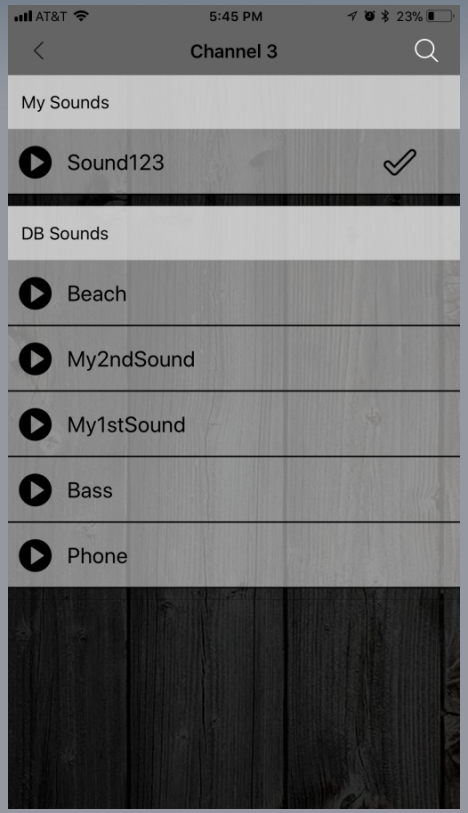




iOS Application UI

- Edit Sound Set
  - Delete Sound set
  - Add /Delete Channels
  - Play sound on Channel
  - Edit Title
  - Select a Sound to Edit
    - UICollectionView
- Select Sound for Channel
  - Search through sounds
    - UISearchController
  - Add new Sound Sets





Website

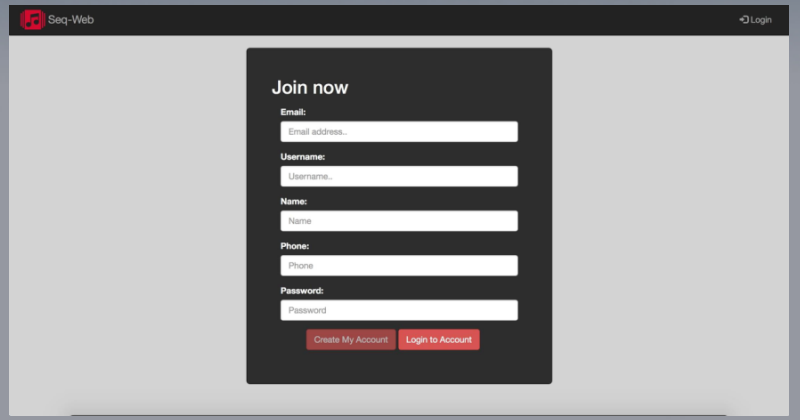
Website

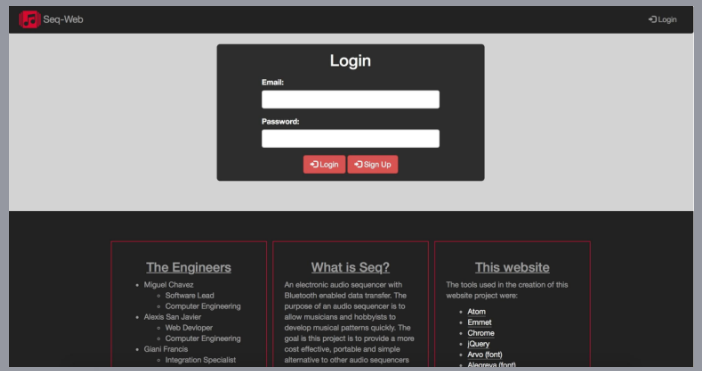
- Database
  - Firebase
- Frameworks
  - Angular
  - Bootstrap
- Functionality
  - Create Account
  - Login
  - Edit Sounds

Bootstrap

Website UI

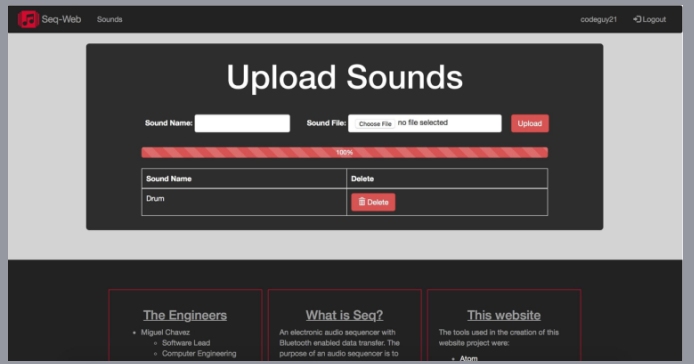
- Register
  - Email
  - Username
  - Name
  - Phone Number
  - Password
- Login
  - Email
  - Password





Website UI

- Home
  - Sounds
  - Logout
- Upload Sounds
  - List of sounds
  - Upload file
  - Delete file



Administrative

Budget

Item	Cost
5Pcs MAX7219 DIP-24 8-Bit LED Drivers Chips for LED display/ Instrument	$10.90
20PCS SN74HC00N SN74HC00 74HC00N 74HC00 DIP14 IC#20236	$6.99
2 of Chanzon 100 pcs 5mm Red & Yellow-Green LED Diode Lights	$6.47
Raspberry Pi	$35.99
Adafruit Bluefruit UART Friend	$19.99
Major Brands 74LS148 ICs and Semiconductors, 8 to 3 Line Priority Encoder (Pack of 4)	$11.11
Wood parts for enclosure	$15.99
3M 9000NA-20-CC Sandpaper Aluminum Obyide, 9-Inch by 11-Inch, Very Fine	$5.07
Hydrogen Peroxide Antiseptic Solution 16 Oz	$5.88
Super Nail Pure Acetone Polish Remover, 8 oz	$6.42
Single Sided Copper Clad Laminate PCB Circuit Board 4X3 (10pcs)	$10.99
Green Envy Muriatic Acid	$10.99
Total	$146.79

Work Distribution

	Mobile App	Web Application	Bluetooth Communication/Data Transfer	System Design/Hardware
Miguel Chavez	Primary	Secondary	Secondary	
Giani Francis	Secondary		Primary	Secondary
Brandon Marcoux				Primary
Alexis San Javier		Primary		Secondary

Questions?