Design Document 5 - Proposed Design

Part 2

Team Name: sdmay23_30

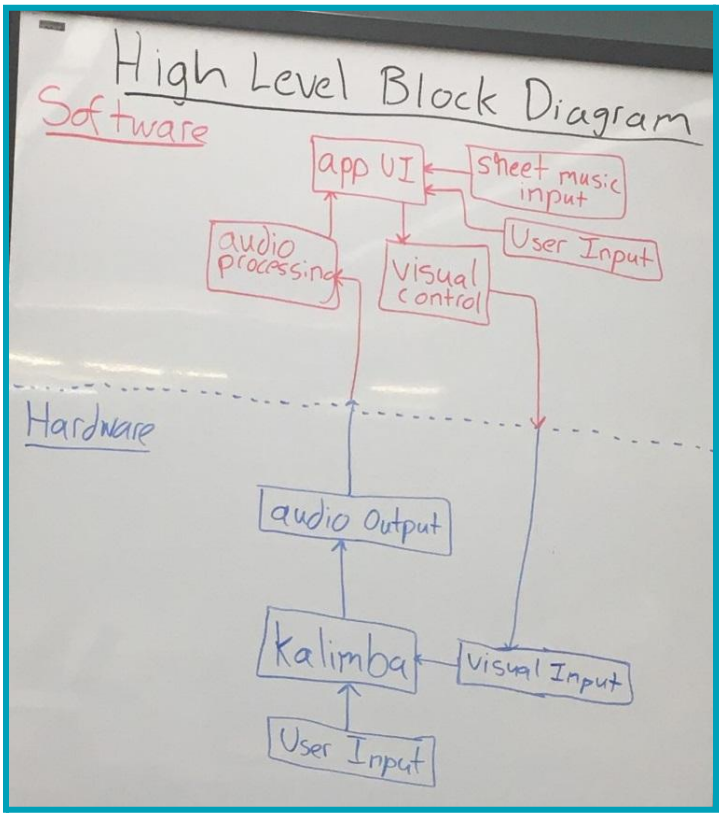
Team Members:

1) Mesa Hassel **2)** Eileen Hillier
3) Kaitlyn Nolting **4)** Stuart Pearson
5) Andrew Adams **6)** Daniel Duerr
7) Isaac Vrba

4.3 Proposed Design

4.3.1 Overview

High Level Functional Block Diagram

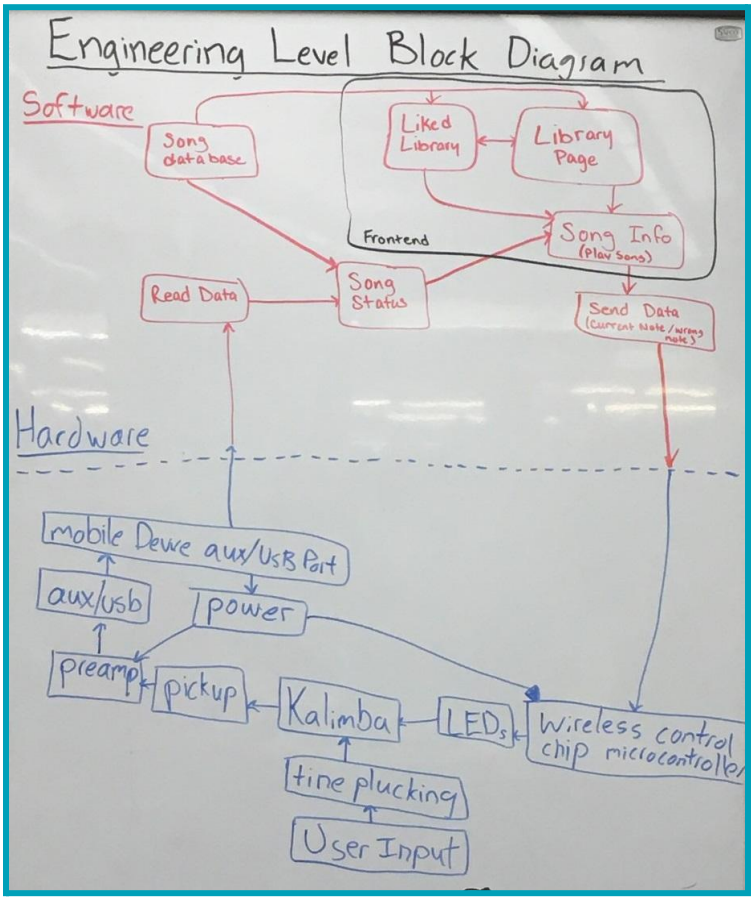


The software side of our design will be a mobile application where a user can add/remove songs to their own liked library. They can add or remove songs from the app's existing library or upload sheet music that will then be processed to be able to be played. They will then be able to play songs from this library, the notes of this song will then be displayed on the device that is connected to the kalimba. There will be a led light over every tab on the kalimba that will light up based on which note the user needs to play next. This device will also be able to process which note the user has just played and whether it was correct or not.

Once this is determined the software will decide if the next note in the song will be played or if the same note will display again.

4.3.2 Detailed Design and Visual(s)

Engineering Specific Block Diagram



The hardware side of the device would consist of inputs of an array of LEDs controlled wirelessly with a microcontroller chip and user inputs from the plucking of the tines on the kalimba. The audio output would be gathered with an audio pickup fed into a preamp in order to make the signal intelligible for software to read it. This will then be fed into an aux/usb port which will connect to the user device (generally a smartphone).

For the backend side of the software, the mobile application will have a database that will store songs. A user will be able to upload sheet music that can be read by the software. The software will parse through the song and each note will be assigned a variable. The variables for each song will be stored in an array or another way in order to tell the physical device which light to turn on. Once the note that the user played is determined the backend will take in that information and compare it with the note we are on in the array and determine whether the next note should be lit or if we stay on the current note. (A tine will stay lit until the user correctly plucks that tine)

For the frontend side of the software, the mobile application will have a page that will have a user's library for songs they have added to it as well as songs that we added, a page/function where they can upload sheet music and a way to start a song on the physical device.

4.3.3 Functionality

Step 1: The user mounts the device to the kalimba.

Step 2: The user applies a pick up if one is not already installed

Step 3: The user connects the pick up to the preamp.

Step 4: The user connects the preamp to the appropriate port on the mobile device.

Step 5: The user connects power to the microcontroller via the appropriate port on the mobile device

Step 6: The user opens the app on the phone.

- **Step 6.1:** The user can upload sheet music to add songs to their library.
- **Step 6.2:** The user can add songs to their liked library from the larger library
- **Step 6.3:** The user can view the details of a song

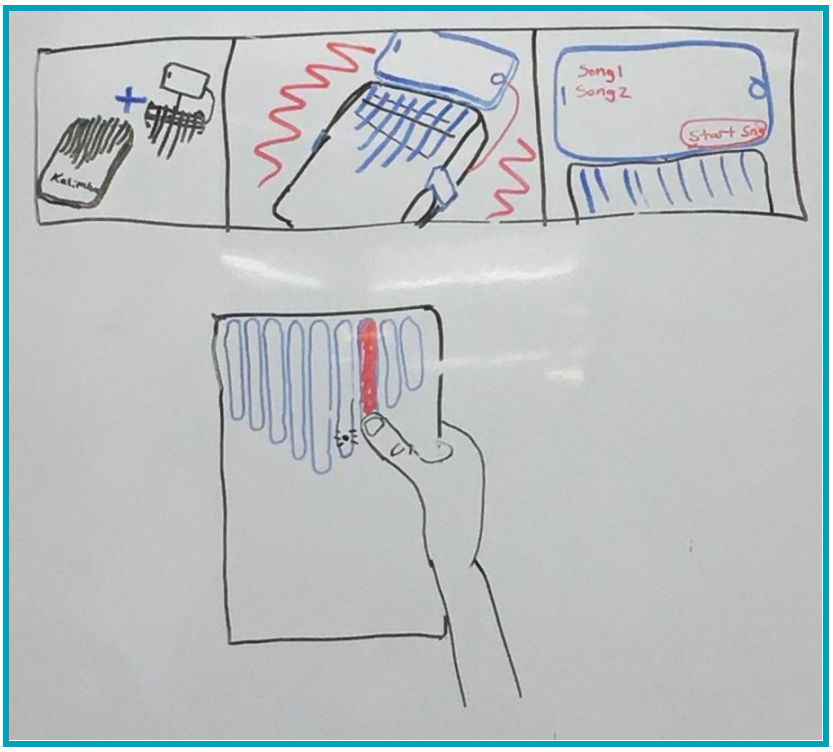
Step 7: The user selects a song to play on the application.

Step 8: The application sends the note data to the kalimba to light up the tines

Step 9: The user plucks the lit up tines in the order they are presented based on the note order of the preselected song.

Step 10: Once the user has finished a song, they can select another song to play or disconnect the power to the device and unplug the preamp.

Storyboard of the User - Our project with a Kalimba by Isaac Vrba



Tickling the Tines - Isaac Vrba

4.3.4 Areas of Concern and Development

How well does/will the current design satisfy requirements and meet user needs?

We have tried to keep user needs a top priority when creating our current design. The main requirement our user needs is to be able to easily learn how to play songs on the kalimba. Our apparatus does this very well at a high level by easily allowing users to be able to choose certain songs, and then correctly lighting up the notes with the LEDs so that the user gets feedback when they play correct or incorrect notes.

Based on your current design, what are your primary concerns for delivering a product/system that addresses requirements and meets user and client needs?

Our primary concerns for our current design include the method of attachment, ease of user set up, and the ability for users to upload their own songs. We have not tested a method of attachment of the device to the kalimba and have not settled on a concrete idea to begin testing. The device and user set up should be designed to maximize ease of use, our current set up has not been streamlined yet for this capacity. We would like for the user to be able to upload their own songs to our app, this could present difficulties with reading the music and correctly formatting the song to play on the kalimba.

What are your immediate plans for developing the solution to address those concerns? What questions do you have for clients, TAs, and faculty advisers?

As far as the mounting attachment goes, we need to wait until we decide on what hardware that we are going to use (LEDS, amps, and microcontrollers). We are currently still working on how the design will come together, and we specifically have set up meetings with professors that are experts with analog electronics and microcontrollers to gain some insight on how to best meet the needs of the users with the hardware that we want to implement. These questions in the meetings will mostly be centered around microcontroller and amplifier selection. When this is all done, we will be able to work on the mount.

4.4 Technology Considerations

Describe the distinct technologies you are using in your design. Highlight the strengths, weaknesses, and trade-offs made in technology available. Discuss possible solutions and design alternatives.

Hardware technology

- LED
    - LEDs allow us to locate a lot of lights in a small space, perfect for the small design of the mount used for the Kalimba. We wanted to make sure the user is able to see the upcoming notes and any mistakes if they play the wrong note. We think that this is the best visual we can provide to the users.
- Pre-amp
    - A strength of the pre-amp is that it makes the sound waves much more readable for our system to pick out each individual note. A weakness, however, would be that a pre-amp is another separate piece of hardware that we would have to add to our design which would increase the cost as well as the size of our entire contraption.
- Wireless microcontroller control chip
    - We are still not completely sure of what model we are going to use, however the particular model we are going to use will be discussed and purchased by the end of our research period. The strengths of the wireless microcontroller include the robustness of the device, allowing us to plug in multiple hardware components and communicate to everything in an easy manner.
- Pick-Up
    - What is a Piezo pickup and why do we need an amp? Piezoelectric amps work due to the piezoelectric effect. Piezoelectric materials, usually a type of crystal or ceramic, produce a voltage in response to mechanical compression. So a piezoelectric material can produce a voltage when squeezed and have the voltage disappear when the squeezing ceases. Just as sound waves are pressure waves in the air, vibrational waves are pressure waves in a material. So, when a musical instrument is making a sound, it is vibrating. By attaching a piezoelectric pickup to an instrument, the piezoelectric material in the pickup will vibrate at the same frequency as the instrument, and thus, the piezoelectric material expands and contracts at that vibrational frequency and a voltage will appear that perfectly matches the sound frequency. However, the output impedance of the piezoelectric material is usually very high (the piezoelectric material is not a conductor after all), which means the potential output voltage that a computer would have to read would be very small to the point a computer might not even recognize that a voltage signal is present. So the signal needs an appropriate amplifier between the piezoelectric material and the computer for the majority voltage drop to occur across the load (i.e. computer) instead of the piezoelectric

material. (Ask me what a voltage divider is to properly explain impedance matching.) Anecdotally, when I plug the kalimba into my roommate's guitar amp and turn the volume all the way up, it is still really quite relative to connecting a guitar. This happens because the output impedance of the piezoelectric pickup is still large relative to the load impedance of the guitar amp.

- Explanation video

Software technology

- Song database
  - there will be a lot of songs that the user can choose from and data for each song so this will be easiest to work with if stored in a database
- External music sheet files
  - musicXML, midi, etc.
    - There are multiple strengths and weaknesses for each external music sheet file type we have considered.
      - Midi would be a lot easier to read from, because the notes can easily be changed into an array of data. However, we would need to create midi for each song or find a library of midi arrangements for each song.
      - Sheet music would have the strength of it's very easy for a user to find, and they could easily upload it. However, a weakness could be that sheet music may be more difficult to parse when changing the notes to an array of data.
  - The purpose of only accepting specific file formats is so that our software can easily read it
- React native
  - React native does well to support designing an application that supports both ios and android
  - React native has lots of supporting documentation and useful resources to be able to learn how to use it for our system.
- Git
  - We decided that git would be a good technology to use for version control because it does a very good job of collaborating on teams with code, and we all have a lot of experience using it.
- Tune function
  - this function will pick up the tune when a user plays a note
  - this can be used to detect whether a user played the right or wrong note instead of using the pre-amp
  - this function could also be an in app function the user can use if their kalimba is out of tune which is important for our app to work overall

---

4.5 Design Analysis

A pickup for the kalimba has been tested. A piezoelectric pickup was attached to the kalimba and then subsequently plugged into a guitar amplifier. The guitar amplifier has to be turned up to full volume for a soft output sound to be heard; although, the output sound was just as clear and full as the kalimba itself. After some research, it has been determined that a preamplifier is going to have to be added to our

design in order for the software to receive a waveform that can actually be used. The issue is that the output impedance of the piezo pickup is high, around 1 Mohm, and the input impedance of most amplifiers is not relatively high, which is why the output on the guitar amp was still soft despite being at max volume.

We have yet to test other aspects of our design, but subsequent testing may alter future plans just as testing with the piezoelectric pickup and the guitar amp has shown that we will need a preamp for the pickup to be effective.