ECE1778 - Creative Applications for Mobile Devices
Tootti: Final Report

April 21st, 2021

Team:
Ricci Ebron (1003527846) | Specialist
Sunaal Mathew (999473225) | Programmer
Hanyu Xi (1007489950) | Programmer

Word Count: 2484

Table of Contents

1.0 Introduction

1.1 What is Tootti?

The goal of the iOS app Tootti is to allow musicians to collaborate virtually in as real-time as possible, providing a streamlined studio to create a click track, record a piece at the same time as another musician, and merge the tracks for immediate listening.

1.2 Why is it needed?

The reason this app is needed is because the pandemic has withheld all opportunities for musicians to meet in person for rehearsals and performances. Unfortunately, remote music collaboration over the internet is labour intensive and can be frustrating to most musicians.

Music collaboration online usually involves a single composer or musician organizing for tracks to be collected from different musicians followed by post-editing using a digital audio workstation solution like Garageband or Ableton . The final product to the audience's eye seems like a very polished and fun performance, but the recording process is actually quite isolating.

Musicians can also collaborate live over existing calling platforms such as Zoom or Jamkazam. Unfortunately,the presence of a time-lag due to the various networks involved still affects music quality.

Tootti aims to provide an accessible solution for all musicians using a streamlined live recording studio that allows music collaboration on iOS. This app will still be useful beyond the scope of the pandemic so that music collaborations can happen internationally at any time.

2.0 Statement of functionality

Tootti provides musicians with a live studio experience on iOS to record, mix and merge tracks with other musicians online. The features of Tootti are as follows:

2.1 User Authentication

The first screen of Tootti allows users to sign in and sign up for a new account. The sign up page leads to a new page for user information, as shown in Figure 1 below.
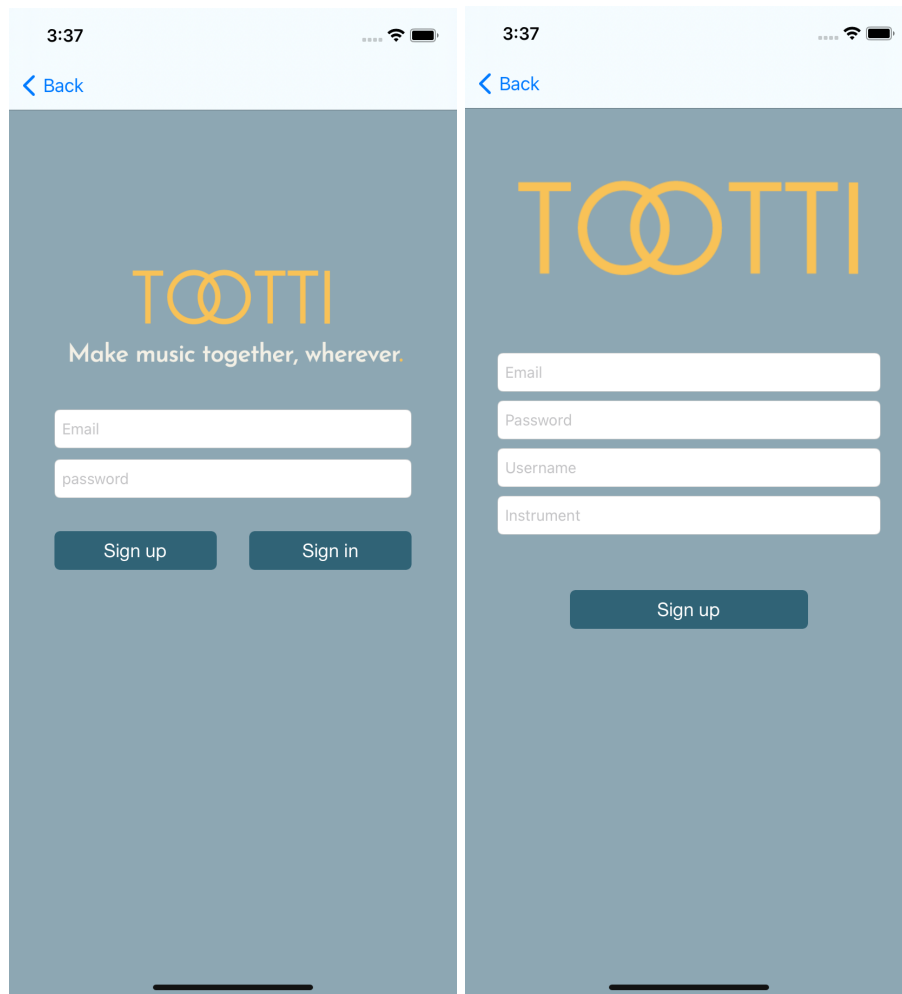

Figure 1: Login page (left) and create new user page (right)

2.2 Create and Join Sessions

Once the user signs in, they can create a new session or join an existing session from the home screen, as shown in figure 2. Session information includes the name of the session, instruments used, and the number of musicians expected in the session. Once in the session, users can navigate to the home, clicktrack, studio, library or merge pages by simply selecting the desired page on the tab bar interface at the bottom of the screen.
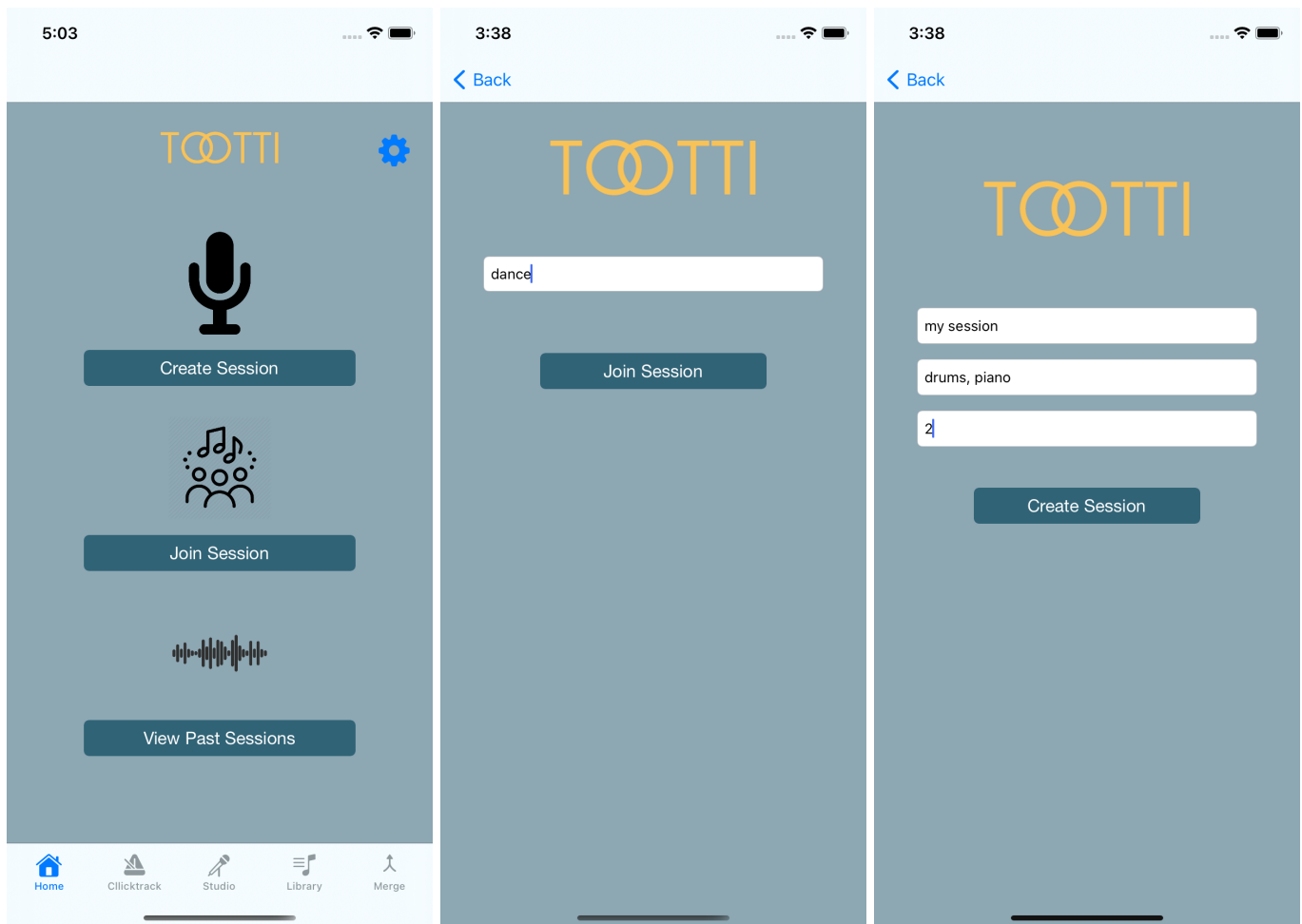

Figure 2: Home page (left), join session page (middle), and create new session page (right)

2.3 Click track

The click track serves as a reference for recording where all users will play to the selected click track. This ensures that all tracks from different users can be synchronized easily once the recording is completed. The host user can upload a click track from their iOS music library and this click track is stored in Firebase. Figure 3 below shows the click track screen for the host user as well as a sample click track stored in the iOS music library. All users in the session can preview the click track on the click track page.
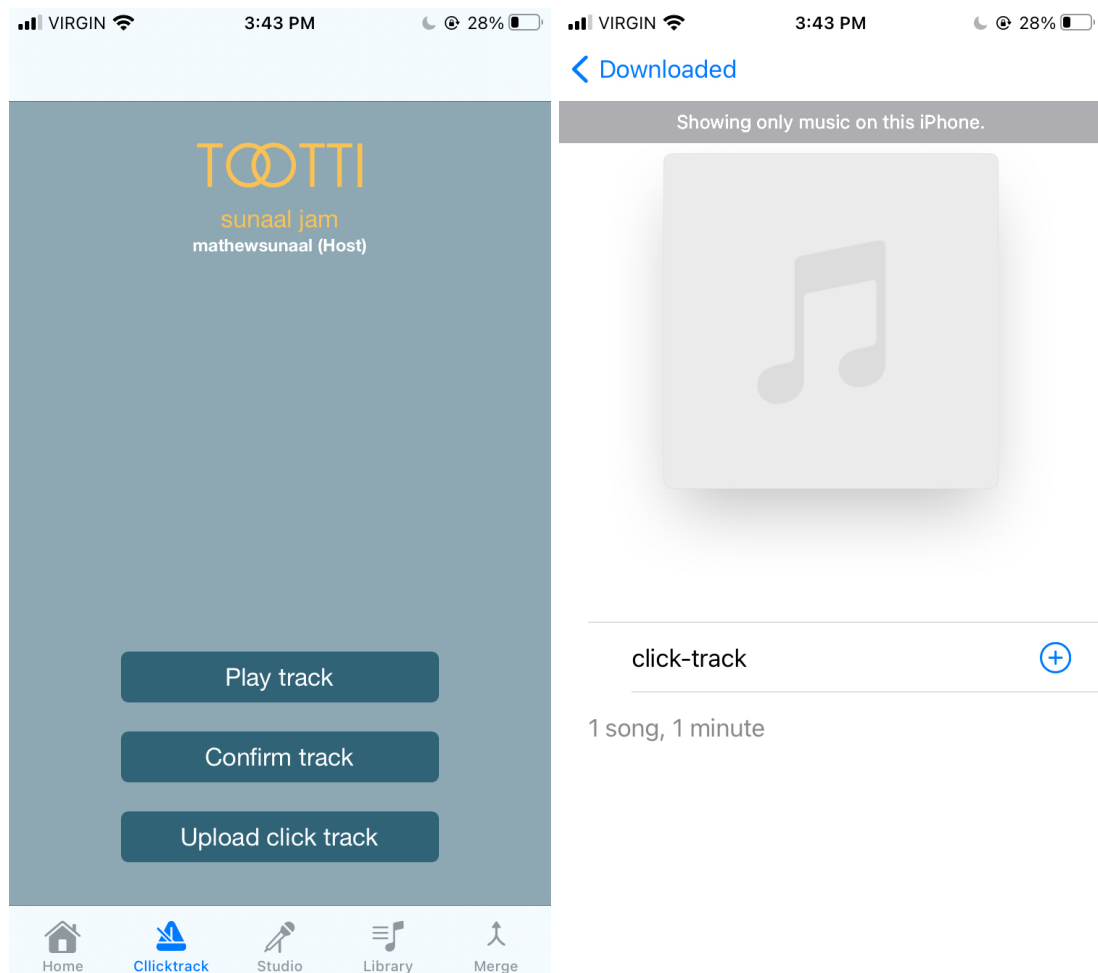

Figure 3: Click track screen to upload and preview click track (left), and iOS music library (right)

2.4 Recording Studio

The studio page is where both the host and guest users can record tracks in reference to the selected click track. The status of each user in the session is shown in the participant's table in the top half of the page, and the status toggle can be used to change a user's status from *"Standby"* to *"Ready"*. All status updates are live and shown on the studio page for all users in that session. Recording can be controlled with the recording button at the bottom of the screen. The audio waveform and timer provide the necessary visual feedback to the user while recording as shown in Figure 4 below. Once the recording is completed, the user can name and save their recording. All recordings saved are locally in the library page.

When the host starts recording, Tootti begins recording for all other users in the same session. Each user will only hear the click track and no real-time audio from other users while recording, and the click track will begin playback at the same time as when the recording begins. Hence, this design prevents any network-based latency from affecting audio quality or a user's ability to stay on tempo.
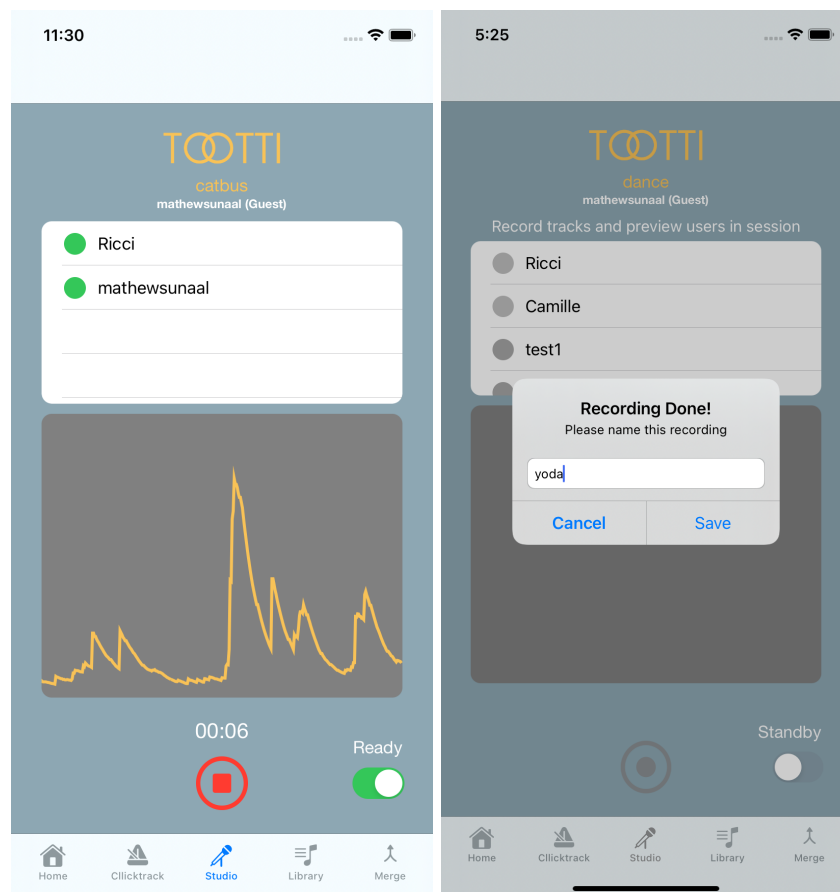

Figure 4: Live recording in studio screen (left), and saving the recorded file (right).

2.5 Library & Merge

On the library page, the user can preview tracks by clicking on the play button beside each track as shown in Figure 5 below. Tracks can be selected by tapping them in the table and pressing the *"Confirm"* button to send the selected tracks to the merge page as shown in Figure 5. Once on the merge page, users can tap the refresh button on the top right to download all tracks recorded and selected by users in the session.

To preview the merging of different tracks, users can select the tracks they would like to preview as merged (yellow highlight indicates selection) and press the play button below to merge and play all tracks. A link to download the final audio file with the merged result can be shared by pressing the *"Merge"* button. Once the export is completed, a prompt will appear where the user can share a direct link to the final merged track on Firebase, as shown in Figure 5 below.
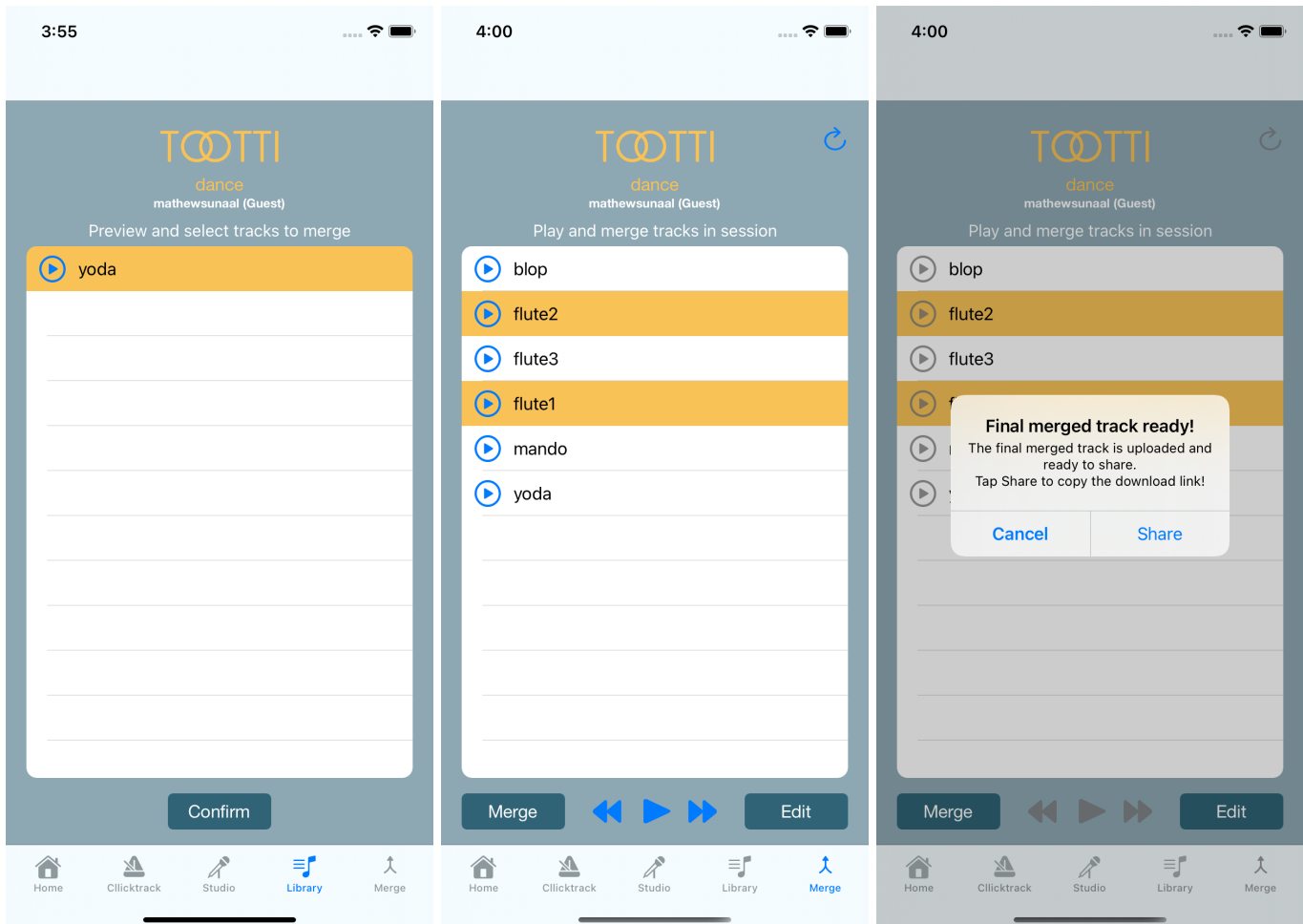

Figure 5: Library page with local recordings only (left), merge screen with all recordings in current session from all users (middle), and sharing final merged track as a download link (right)

The demo in the final presentation consisted of Ricci playing music with her teacher, Camille. Ricci created a session as the host and Camille joined her session. Both Ricci and Camille were able to successfully record their tracks in reference to an uploaded click track. Ricci was able to save her recording and send it to the merge screen, where her track was uploaded to Firebase and then shared with all users in the session. Ricci could also see all tracks ever recorded in that session after refreshing the merge screen.

Unfortunately, due to an app crash on Camille's side, her recording was not saved and hence could not be fully uploaded to Firebase for merging with Ricci's recording. We speculate Slow internet upload speeds are not handled well in the app.

Firebase calls and operations need to be better handled to avoid crashes in the future work for Toottti. Our team re-recorded the demo portion of our presentation later to display the proper functioning of the app. The video can be viewed here: https://drive.google.com/file/d/1l3n-qe4VvjdFzDQWtv_YI2dTlOssfkme/view
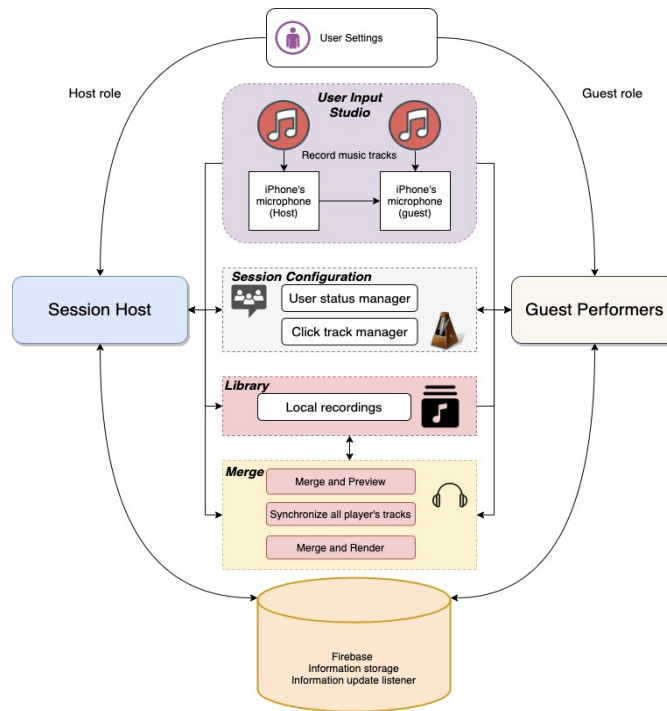
3.0 Overall Design

Figure 6: Block diagram

Figure 6 above describes the block diagram for Tootti. Considering the scope of this course, Tootti will allow for two user roles - Guest and Host. The Guest user can record music clips using the iPhone's microphone and share them with other users in the session. In addition to the above features, the Host user will also act as an administrator of the recording session and manage the click track.

The critical components of this design include the *User Input* module, *Session Manager* module, *Local Library* module, *Merge* module and storage (Firebase). The user input module consists of the iPhone's microphone and associated native iOS software frameworks to record audio/music played by the host or guest user. The *Session Manager* module, controlled by host users, handles synchronization, waveform displaying and recording music tracks from all users. After the user uploads the selected track, the *Merge Manager* module handles mixing/merging of different music tracks from users and generates the resulting merged track, which is stored in Firebase and accessible by all users in the respective session.

4.0 Reflection

4.1 Being adaptive and flexible

Being adaptive to new problems and not sticking with just one path is important, especially at the early stages of a project. At the early stage of our app development, we focused on video sharing instead of audio output, but later on we made the decision to switch to audio method due to the technical debt.

4.2 User-centred design and iterative development

Throughout this design cycle, the importance of user-centred design can not be overlooked. The lack of user centred design can cost time and effort and can greatly determine the success or failure of a project. The programmers and specialists had weekly spiral meetings to discuss any user experience issues that can negatively impact our potential users.

4.3 Working with a multidisciplinary team is fun!

Working in a multidisciplinary team on the app, presentations and reports was a great learning opportunity. Communication and cohesive vision are essential to make sure that everyone's efforts are for the same cause. The multidisciplinary team makes the design thinking phases more innovative and creative.

5.0 Contribution by each group member

Ricci Ebron was the music specialist in this project. She came up with the idea of the app, compiled a list of all the necessary features, and created the flow of pages in Tootti by designing the user interface in Marvel. Ricci worked closely with Sunaal to keep the UI updated based on the needs and limitations of the features. She created the logo and found all the assets for the app. Ricci also created all the musical assets for the pre-recorded demo tracks and clicktracks of the first few presentations. For the final demo, she invited, taught, and rehearsed with Camille on how to use Tootti.

Hanyu Xi was a programmer in this project. Hanyu primarily focused on the backend development of this project. He designed the database and set up object oriented class utility objects to help integrate the server-side application logic with frontend work. He also utilized various Firebase platforms and API commands to store and update the data. In addition to the backend tasks, he also designed and implemented the waveform display algorithm and the user status update handler in real-time processing.

Sunaal Mathew was a programmer in this project. Sunaal worked very closely with Han throughout the development of the iOS app and integration of Firebase functions in the app. He developed the storyboard sequences for the user interface, implemented UI elements and functions based on the mockups provided by Ricci. He was also responsible for implementing the recording function and interface, playback controls for single and merged tracks, refreshing local data sources based on Firebase updates, and finally exporting the merged track as a single audio file.

6.0 Specialist context

Tootti changes the game for performing musicians. Of course, nothing beats making music with others in person, but when that is not possible due to the pandemic or because the musicians are living in different countries, Tootti creates the next best environment. The sense of community between musicians is part of what makes music creation so special, so recreating the real-time aspect was a top priority for Tootti. We quickly realized that no matter how we approached it, performing virtually together in real time would introduce the problem of lag, and it would only get worse when the distance between musicians increased. Most low-latency programs such as Jamkazam and Jamulus try to find ways to overcome network lag that is out of developers' control, but it was an essential early decision for the development of Tootti to change off this path. As a result, we had more time to think of a creative way around this age-old problem: that recording at the same time as another musician while muted but still hearing the other part(s) in a bass track and staying in time with the click track eliminates lag!

Tootti allows for flexible recording scenarios with its live versus offline modes. Musicians can record at the same time as each other in the live mode, but if it were an ongoing project where parts needed to be recorded individually, the offline mode would still offer that possibility. Even with Tootti still being its early stages, it is already a sought-after application by my fellow musicians. When asking for feedback, a friend commented that her acapella group has had varying degrees of success with Jamulus, and they would be very interested in using Tootti instead.

In education settings, Tootti could revolutionize the boundaries of teamwork through music. Tootti could partner with music teachers and institutions such as elementary and high school ensembles, university music faculties, and local music schools. Not only would students be able to collaborate with each other within their schools, whole classrooms could collaborate with other ensembles in different countries. As a type of musical exchange, students would learn about the culture and music of other countries, network further than they could have ever imagined, and form lasting memories and recordings.

Finally, Tootti was made to be as straightforward and easy to use as possible. Performing musicians spend countless hours practicing their instruments, which does not leave extra time to

learn how to use audio production programs such as Garageband or Pro Tools. Personally, I found that constantly re-recording with complicated setups caused a lot of frustration, so Tootti aims to minimize the amount of steps between the performance and the final recorded track. Tootti's ability to automatically merge all the tracks truly alleviates a lot of the production process that would otherwise add a lot more work for musicians before they can publish their recordings. With a linear project path and simple interface, Tootti is made to be accessible for all musicians all over the world.

7.0 Future work

Tootti has a wide scope for improvement in usability and functionality, and the key areas for future work are as follows:

7.1 Live and offline modes for recording

Currently, both modes are present in the app, however, the distinction is not clear in usability. This can be confusing to users as they will not know which model is active at any moment. In the future, it is recommended to implement a clear option to switch between live and offline modes.

7.2 Audio compatibility

Tootti currently only works with wired headsets. Future work will involve supporting wireless or bluetooth headsets as well as integrating 3rd party libraries such as AudioKit to extend Tootti's capabilities in editing and post-processing.

7.3 Audio editing features in merge

There are existing third-party libraries such as AudioKit that provide audio editing and mixing capabilities including adjusting the alignment, tempo, pitch and loudness of individual tracks. It is recommended to explore the integration of AudioKit to extend Tootti's capabilities for musicians.

Permission statement

All groups members of Tootti agree to share the following:

- Video of final presentation
- Report

All group members of Tootti do not agree to share the source code due to plans of future commercialization.