CS 30700
Team 10: Project Charter
Project Title: Rhythm

Team Members: Indhu Meena Ramanathan, Richard Hansen, Steven Dellamore, Columbus Holt

Problem Statement:

Converting handwritten music sheets into a digital format that can be easily tested, customized, and shared with others is not an easily completable task. In order to facilitate this tedious task that may takes hours to complete, in comes Rhythm, Rhythm allows any music writer, new or seasoned, to be able to take a picture of music sheet, which will be automatically loaded and parsed into a digital music sheet. These sheets will open for easy customization, including changing and adding notes, in addition to previewing the music by playing the selected sheet with a selected instrument. While there are applications such as PlayScore that allow users to take an image of their scores and play the music, those applications do not allow users to convert handwritten music to a digital version nor do they allow users to edit the notes after upload.

Background Info:

Audience

Music writing is an ubiquitous task that has existed for a multiple of years. However, there are many nuances while writing music. Many music writers, new or accustomed, sometimes struggle with easily testing, customizing, and sharing their music due to the myriad number of steps. It is quite surprising that despite the huge target audience with anyone interested in music composition, existing options do not provide all the features to enhance the process of creating, composing, and editing music.

Similar Platforms

There are several existing technologies that allow users to take images of already composed music notes and automatically play them; however, technologies such as these do not allow users to take pictures of handwritten compositions and convert them into digital formats. There are other applications such as GarageBand that allow users to choose instruments and play music create live music with those instruments; however, GarageBand does not allow users to enter notes to automatically play compositions.

Limitations

While many of these existing platforms are useful, some of their main limitations is that they do not allow users to take images of handwritten notes to convert to digital formats, nor do allow users to edit their compositions after they have been saved. Our goal is to make a platform that is intuitive, so that users will immediately be able to create, edit, and play compositions, all with one application.

Functional Requirements:

1. As a user, I would like to register my account
2. As a user, I would like to be able to edit my profile
3. As a user, I would like an account recovery option
4. As a user, I would like to take pictures of sheet music, and have it converted into a digital version
5. As a user, I would like to be able to select pictures from my camera roll to convert to a digital version
6. As a user, I would like to be able to view music sheets
7. As a user, I would like to be able to view compositions
8. As a user, I would like to be able to create compositions
9. As a user, I would like to be able to delete compositions
10. As a user, I would like to be able to delete music sheets
11. As a user, I would like to be able to name compositions
12. As a user, I would like to be able to name music sheets
13. As a user, I would like to be able to add music sheets to compositions
14. As a user, I would like to be able to remove music sheets from compositions
15. As a user, I would like to be able to copy music sheets
16. As a user, I would like to be able to copy compositions
17. As a user, I would like to be able to edit music sheets
18. As a user, I would like to be able to add notes to music sheets
19. As a user, I would like to be able to remove notes from music sheets
20. As a user, I would like to be able to move notes within a music sheets
21. As a user, I would like to be able to select an instrument for a single music sheet
22. As a user, I would like to be able to play a single music sheet
23. As a user, I would like to play all music sheets in a single composition
24. As a user, I would like to be able to add other users as friends*
25. As a user, I would like to be able to remove users as friends*
26. As a user, I would like to be able to share my music sheets with friends*
27. As a user, I would like to be able to share my compositions with friends*
28. As a user, I would like to be able to send music sheets as pdfs to my email
29. As a user, I would like to be able to send entire compositions as pdfs to my email
30. As a user, I would like to be able to set the publicity of my compositions (public, private)*
31. As a user's friend, I would like to be able to view my friend's compositions
32. As a user's friend, I would like to be able to comment on my friend's compositions*
33. As a user, I would like to be able to upvote, or downvote, comments*
34. As a user, I would like to be able to delete my own comments*
35. As a user, I would like to be able to export music sheets as mp3s*

36. As a user, I would like to be able to export compositions as mp3s*

* (if time allows)

Non-Functional Requirements

Architecture and Performance

The front-end of the application will be developed using the React Native framework with Javascript. This will allow us to easily export our app to either iOS or Android. We are going to attach Redux to the front-end to make it simple for developers to access and change states of the app. We plan to use Flask for our back-end because it is easy to learn and extend. It also provides an out-of-the-box fast debugging tool which we can utilize. The flask back-end will connect to our database that will be accessed using MySQL to save information, such as their registration information, compositions, and music sheets. Separating the front-end and the back-end is an easily feasible task that will allow us to expand to other platforms in the future (web application). We will be using Jest for integration tests for React Native. We plan to use PyTests to create automated unit tests to to ensure individual method functionality. To increase performance, we will run the conversion from music sheet to music on the back-end, alleviating the phone running our app from having large tasks. We would like to respond quickly, but depending on the composition size, it may take longer to process and send the data.

Security

Security is important for Rhythm, since we will be saving user data. We will filter out requests in the back-end from the front-end to ensure we do not send any unwanted SQL requests. We will check response headers to ensure that proper authentication is provided from the front-end. We will provide Authentication keys that will be sent from the front-end to the back-end to validate the requests are coming from the react native app.

Usability

Our app will have a clean and intuitive interface, that will allow even new users to quickly pick actions and possibilities that they have at their disposal. We have a large number of customization features, so it is critical that we have an easy to understand interface to prevent users from getting overwhelmed. Luckily, many of the music sheet and composition options will be similar, limiting the number of new tools a user has to understand.

Hosting and Deployment

Since the front-end and the back-end are disjoint we can easily separate the deployment of them. The front-end of our application will be available for download on Github Pages. The back-end will be deployed using AWS EC2 instance because it is an easy and

secure hosting service. Because we are using AWS, our servers can be deployed 24/7, for a low cost.