Product Name: Robot Ear
Team Name: Robot Ear
Release name: Robot Ear
Release date: 12/5/2023
Revision number & revision date: 0

High-level goals:

- Simple even-tempo input audiofile, produce pdf with notes
- Have the web application be able to classify notes
- Have users be able to save the generated music sheets on the web application
- Implement in-app/ in browser recording feature and have Robot Ear read, understand and convert that recording to sheet music.
- Implementing a Log-In database
- Friendly user interface

User Stories: As a {user role}, I want {goal} [so that {reason}]

1. As a user, I want to be able to upload a file of music in order to eventually get sheet music back
2. As a user, I want to have accurate sheet music so that I can play my music correctly
3. As a user, I want to be able to save my generated sheet music so that I can revisit the sheets
4. As a user, I expect a well-thought-out UI design using a designing tool such as Figma for the web application so that it is intuitive
5. As a user, I want the web application to be visually appealing so that I have a good experience
6. As a user, I expect the web application to have a functioning front-end so that I can have a good experience
7. As a user, I expect the web application to be able to segment the audio file precisely so that it can identify notes
8. As a user, I want the web application to be able to segment the audio files with other techniques so that it can classify more precise notes
9. As a user, I expect the web application to be able to classify notes by giving it an audio file so that it can find the notes
10. As a user, I want to be able to receive a pdf of generated sheet music so that I can reference it
11. As a user, I want to be able to log into the web application
12. As a user, I want to be able to save my sheet music on the web application so that I can reference the sheet music in the future
13. As a user, I want the web application to have advanced technology such as ML technologies so that I can have the most precise sheet music
14. As a user, I want the web application to be well-tested, so that I get the most precise sheet music
15. As a user, I want the web application to well connected so that it is functioning

16. As a user, I expect the web application to the well documented so that in the future if the application needs support their is documentation
17. { High } As a user, I want to be able to click on a button to upload a file of music in order to eventually get sheet music back { 3 }

Sprint 1:

Overall goal: segmenter -> classifier -> web app

Segmenter

- Fixed window 500 milliseconds for the segment
- Lilypond that abjad is the library for
- Research machine learning on how to make a better

Web App

- Main goal: design how it looks and do research on functionality
- Upload button recording but no functionality till next sprint
    - Wav File would get send to the back end
- Status (please input -> processing -> finished)
- Research on React how to send the wav file to backend

Classifier

- Finish frequency based classifier
- Testing the limits of classifier (chords/ complex rhythms)
- Allowing tests to inform decision on what ML classifier

Complete list of user stories:

1. { High } As a user, I expect a well-thought-out design using a designing tool such as Figma for the web application so that it is intuitive { 2 }
2. { High } As a user, I want to the web application to be visually appealing so that I have a good experience { 2 }
3. { Medium } As a user, I expect the web application to have a functioning front-end so that I can have a good experience { 4 }
4. { High } As a user, I expect the web application to be able to segment the audio file precisely so that it can identify notes { 3 }
5. { High } As a user, I expect the web application to be able to classify notes by giving it a audio file so that it can find the notes { 4 }

Sprint 2:

Overall goal: backend can receive data from frontend

Segmenter:

- More complex: Either segmentation based on energy, spectral peaks or onset detection

- Micro-segmentation; splitting the audio file into tons of extremely short bits (10 milliseconds) and using note-consistency to identify quarter/half/full notes
- Research machine learning on how to make a better
- Convert to a pdf with abjad (look into python library that does this) create a simple pdf of 4/4 time with a list of notes as the input,

Web App:

- Send .wav file to the backend
- Upload button
- Work with back-end for API's

Classifier:

- Researching and implementing basic machine learning model

Complete list of user stories:

1. { High } As a user, I want to be able to upload a file of music in order to eventually get sheet music back { 3 }
2. { High } As a user, I want to be able to click on a button to upload a file of music in order to eventually get sheet music back { 3 }
3. { Medium } As a user, I expect the web application to be able to classify notes even better by giving it an audio file so that it can find the notes { 4 }
4. { Medium } As a user, I want the web application to have advanced technology such as ML technologies so that I can have the most precise sheet music { 5 }
5. { High } As a user, I want to be able to receive a pdf of generated sheet music so that I can reference it { 3 }
6. { Low } As a user, I want the web application to be able to segment the audio files with other techniques so that it can classify more precise notes { 4 }

Sprint 3:

Overall goal: Frontend sends .wav file and backend sends a pdf back, connected web application, start testing

Web app:

- Receiving a pdf file of sheet music from the backend to have the user download
- Start implementation for saving music + login feature
- Implement a database
- Download button

Classification:

- Testing, researching and polishing classification model

PDFificatin:

- more complex sheet music including time signature, key signature, bpm, genre

Complete list of user stories:

1. {High} As a user, I want the web application to well connected so that it is functioning {3}
2. {High } As a user, I want to be able to receive a pdf of generated sheet music so that I can reference it {3 }

3. {Medium} As a user, I want to be able to log into the web application {4}
4. {Medium} As a user, I want to be able to save my sheet music on the web application so that I can reference the sheet music in the future {4}
5. {Low} As a user, I want the web application to be well-tested, so that I get the most precise sheet music {2}

Sprint 4:

Overall goal: A fully functioning and tested web application with ML technology. User friendly and users can login and save their music

Web app:

- implementation for saving music + login feature
- Implement a database
- Front-end tweaks
- Testing

Classification:

- Testing

PDFificatin:

- Testing

Documentation

Complete list of user stories:

6. {High} As a user, I want the web application to well connected so that it is functioning {3}
7. {High } As a user, I want to be able to receive a pdf of generated sheet music so that I can reference it {3 }
8. {High} As a user, I want to be able to log into the web application {4}
9. {High} As a user, I want to be able to save my sheet music on the web application so that I can reference the sheet music in the future {4}
10. {High} As a user, I want the web application to be well-tested, so that I get the most precise sheet music {2}
11. {High} As a user, I expect the web application to the well documented so that in the future if the application needs support there is documentation {3}

Sanity check (as team)

- Is the plan within the team's capacity?
    - Yes, the plan is within the team’s capacity. We have split up the team based on what their expertise are and where they have experience.
- Is the work distributed across the sprints fair and reasonable?
    - Yes, the work is distributed across the sprints fairly. Every component of the web application is crucial and every team member is part of one of the components
- Add up user story points and compare with teams capacity:

- Adding all the user story points, it is within the team’s capacity

---

Product Backlog

- No product backlog for this release