EE/SE/CPRE 491 - Spring 2019

Student Suggested Project

Sheet Vision

Project Plan

Team Number

sddec19-13

Faculty Advisor

Alexander Stoytchev

Team Members

Bryan Fung

Garrett Greenfield

Ricardo Faure

Trevin Nance

Walter Svenddal

Team Website

http://sddec19-13.sd.ece.iastate.edu/

Version: April 24/Final Version

Table of Contents

List of Abbreviations & Symbols

1.	AWS	Amazon Web Services
2.	API	Application Programming Interface
3.	MIDI	Musical Instrument Digital Interface

List of Definitions

1. **Sheet Music** - Music described in written or printed form. Sheet music must have multiple lattices of staffs and bars, with notes or rests placed upon those lattices. An example of this format can be seen in Figure 0.1.
2. **Musical Notes** - A sign or character used to represent a tone, its position in relation to nearby staffs indicating its tone, and its form indicating the duration of the tone.
3. **Tabs** - A form of written music, but instead of being represented in the traditional sense (what tone it makes), notes are represented by the specific position they are supposed to be played in.
4. **Staff** - Element of a piece of sheet music. A long, horizontal bar on which music notes are placed to indicate their tone.
5. **Bar line** - A short vertical bar which crosses multiple staffs, representing time signatures.
6. **MIDI File** - A file format allowing the digital storage of music forms. A MIDI file only contains information on what notes should be played at what times.

List of Figures

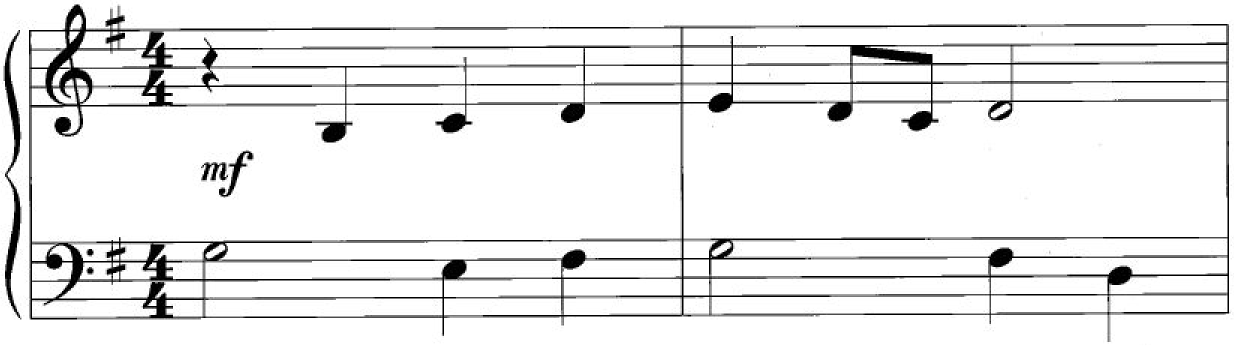


Figure 0.1: An example of the sheet music format we would expect in our inputs.

1 Introduction

1.1 Acknowledgment

We would like to express our gratitude to our advisor Dr. Stoytchev for taking the time to help us map out our project, as well as providing technical assistance. We would also like to thank Dr. Daniels for providing us with course resources and guidelines to follow, to better ensure our project success.

1.2 Problem Statement

Reading sheet music is not easy task for inexperienced musicians. Because of this, many newer musicians tend to prefer learning music through the use of tabs, YouTube tutorials, or other simpler means. However, the issue with tabs and other methods of reading and learning music is that they lack the complexity to be able to convey all of the fine nuances a given piece of music may have. Therefore, the best option to capture all the details most musicians wish to convey when writing music is sheet music. Sheet music is valuable in being able to represent a better set of instructions on how to play a piece of music, but again, its more complex nature can make it intimidating to new learners.

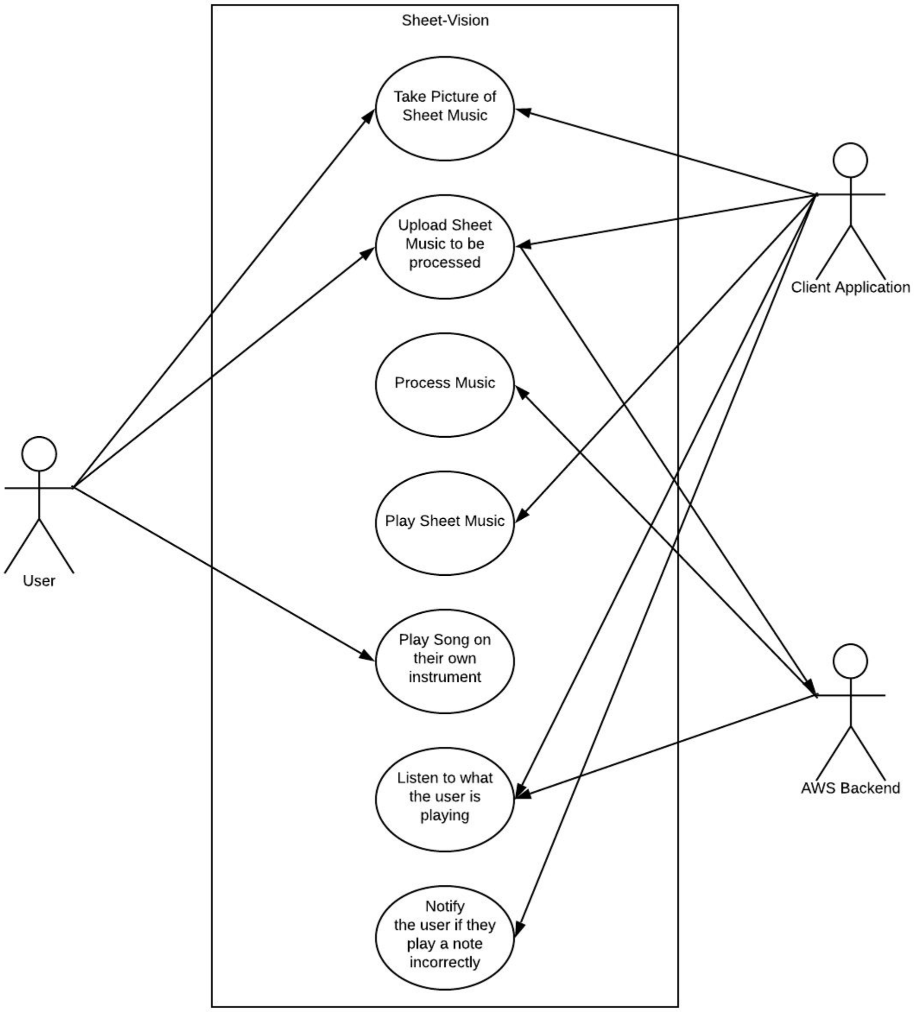
Our solution for this is Sheet Vision, a multiplatform application to help its users practice reading sheet music. Sheet Vision has three main functionalities:

1. The user can use Sheet Vision to take a picture of a piece of sheet music with the device's camera, analyze the music on a cloud server, then return a MIDI file containing the musical information found on the sheet music. The user can then choose to have the scanned song played back to them. This will help the user to confirm their understanding of what the song should sound like.
2. Sheet Vision can display a digital keyboard which will play the notes on the sheet music as the song is played back to the user. This will help the user visualize what actions they should perform to play the scanned song on a piano.
3. Sheet Vision will be able to listen to the user perform the musical piece on an instrument of their choice, and provide feedback by comparing their playing to the sheet music the user scanned. This will help the user identify areas of improvement in their playing.

1.3 Operating Environment

Our product is expected to be used quiet indoor environment, with bright and uniform lighting. Our ideal environment is quiet to allow our application to more accurately pick up sound from the user when they're playing along with our on-screen piano prompt, showing which keys should be played. Lighting is also important to allow the image taken to be clear and evenly colored. This is necessary so our computer vision algorithm can accurately detect the notes and rests, and where they lie on the staffs.

1.3.1 Use-Case Diagram



Sheet-Vision
Take Picture of Sheet Music
Upload Sheet Music to be processed
Process Music
Play Sheet Music
Play Song on their own instrument
Listen to what the user is playing
Notify the user if they play a note incorrectly
User
Client Application
AWS Backend

1.4 Intended Users

This product is intended for beginners who have a basic understanding of sheet music, and need help improving their comprehension. This application should provide instructions simple and clear enough for even first-time musicians can keep up with using our product, yet powerful enough to ease some of the struggle of reading more complex songs.

We also envision this app being useful for veteran musicians, music instructors, or casual music listeners, though they are not our target audience. More advanced musicians may be interested in using our app to help refine their own playing of more complex pieces. All these groups may be interested in using the sheet music scanning segment to quickly hear the "correct" way of playing the song on a piece of sheet music.

1.5 Intended Uses

Sheet Vision is intended to be used as a supplemental product for learning how to play sheet music. Sheet Vision will show musicians how the song is supposed to sound based on the music written on sheet, with the use of audio and visual cues, as well as detect if the user is playing the music correctly. The application is mainly produced for educational purposes.

1.6 Assumptions and Limitations

Assumptions

- Sheet music, though varying in format, will have the same characteristics and symbols you'd expect in a modern piece of sheet music, such as the symbols used to represent notes and rests. An example of the assumed form of the sheet music can be found in Figure 0.1.
- User will have a strong internet connection to be able to take full advantage of note corrections.
- User will have access to a computer or mobile device.
- If the user is using a mobile device, that device must have a camera.

Limitations

- Some features will be limited based on the users possession of sheet music and a musical instrument.
  - The user needs to provide their own sheet music to scan, since our application will not provide it for them.
  - The user will need an instrument to make use of the play-along feature.
- The quality of our output will rely on the quality of both a users camera and microphone, depending on which feature they are making use of.

1.7 Expected End Product and Other Deliverables

- Sheet Vision Multi Platform Application - Expected Delivery Date : December 2019
  - System to read in images of sheet music.
  - Computer vision system used to decode sheet music into information useful for the application.
  - System that uses information provided by the computer vision system to select what notes should be played and when.

    - Audio processing system that can detect and decode audio into information useful for application.
    - System that uses information provided by audio processing system to notify the user if they played the song correctly or not.
    - Can run on a multitude of systems:
        - Desktop Environment.
        - Mobile Environment.

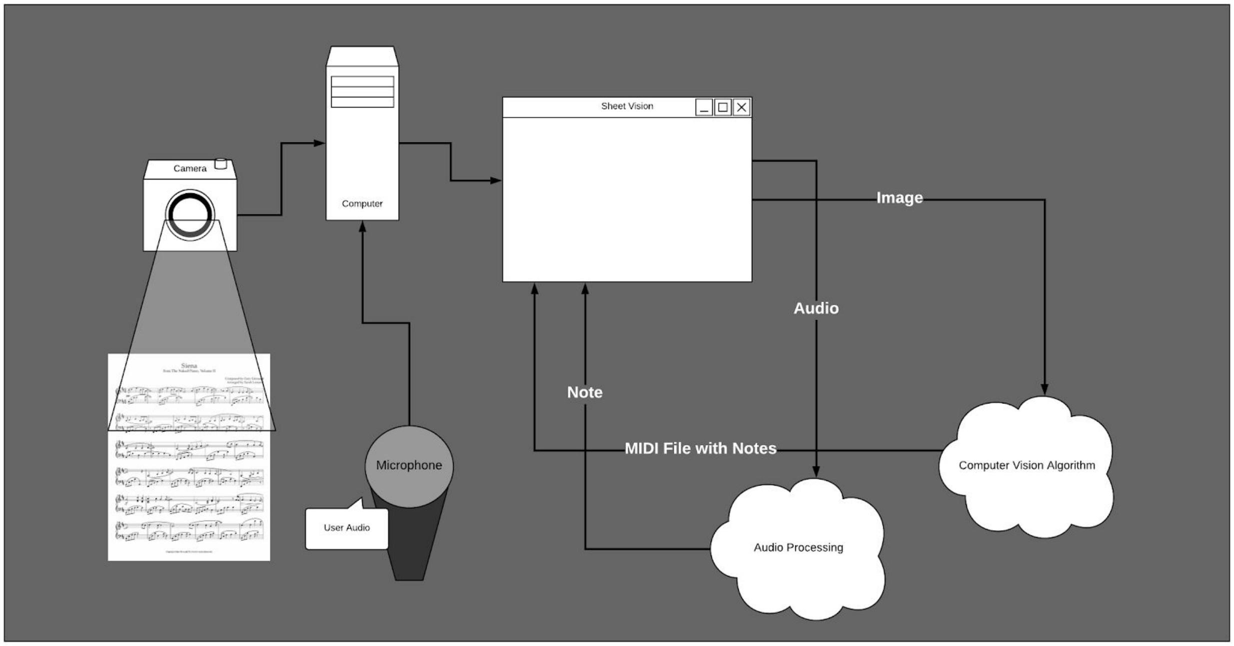
2 Proposed Approach and Statement of Work

2.1 Market Survey

During our market research, we found no products which could directly compete with our projected finished product. We did however find products with similar goals of providing additional support to those looking to improve their music reading:

- **My Sheet Music** - Allows the user to scan a piece of sheet music, then have that music displayed to them as the song progresses. It does not analyze the sheet music, nor does it play the associated song. Our application will perform better by playing the music that is transcribed in a given piece of sheet music.
- **PlayScore** - PlayScore is a mobile application which allows the user to take a picture of a piece of sheet music and plays back the corresponding audio. However, there were several problems with PlayScore. One of them being that the app features a difficult to use user interface. Another problem is that the note recognition was subpar, finding many notes on the page which in reality did not exist, while also not recognizing several notes which were on the page. We can do better by providing an easy to use user interface, having accurate note detection, and providing additional features such as piano animations, and playback correction.
- **Sheet Music Scanner** - Sheet Music Scanner is an IOS application which allows a user to take a picture of a piece of sheet music and allows the user to play the music back with a given instrument. The user interface seems to be fairly easy to use, however this application does not provide any sort of correction for listening to a users playing, so this is an area in which we can do better. Sheet Music Scanner is also only available on IOS, and is not portable, which ours will be.

2.2 High-level Block Diagram of System



This diagram represents the various components that will make up our application. The parts of the application that will be needed for the user to be able to use it would be a camera, microphone and a computer. The portions of the application that will be developed are the audio and computer vision algorithms, the client applications and the communication between all of these.

2.3 Functional Requirements

- Access images from IPhone memory.
- Access images from Android memory.
- Access images from desktop memory.
- Access images from IPhone cameras.
- Access images from Android cameras.
- Access images from desktop cameras.
- Access audio from microphone.
- Create MIDI files from picture of sheet music and play them back.
- Process audio and provide feedback on audio correctness (check against data provided from processed image).
- Play audio based on data provided from processed image.

2.4 Constraints Considerations

Our team has little experience with machine vision and react native so we will have to do a great amount of research and learn quickly to complete the project. We would like to have a feasible model by May 2019, so this particular deliverable will need to be completed by that time.

We will also need to work closely with our team members and faculty to ensure that our system is developed in a way where it could be eventually incorporated into IOS and Android apps. We also need to take into account that our team is composed of undergraduate students who are working part-time, so some of the timelines might need to be readjusted.

2.5 Technology Considerations

Performing computer vision operations can be expensive in terms of memory. In order to be take the strain of expensive computations off of a user's mobile phone, we will perform the image processing on a remote AWS server. This will also prevent any issues which may arise from different operating systems when creating our computer vision algorithm. Relying on an external machine to do all of our image/audio processing is not only vital to avoiding inconsistencies between operating systems, but also will play a huge role in our algorithms to be able to be reused across a multitude of operating systems, ranging from desktop environments to mobile environments, covering as was an audience as possible. Keeping all of our python processing in an AWS machine also gives us the liberty to develop in any kind of framework we choose, giving us the flexibility to choose what frameworks best suit our needs for our clients.

2.6 Technical Approach Considerations

Since our end goal is to have an application which can be used on a computer as well as a mobile device we are limiting ourselves to using a portable front end framework for our application. As well as implementing our machine vision in a language which will be easily interfaced with through our front end application, and is portable to different devices.

2.7 Testing Requirements Considerations

Testing will be done by feeding songs written in the format of sheet music of which are of beginner to intermediate difficulty to the application and seeing if the output of the algorithm is coherent with what is on the sheet. For testing the audio we will play an instrument and make sure the audio processing algorithm detects the correct note that was played. Finally after all individual components have been tested we will test use-cases, in which we have the application read sheet music, play it correctly and then have a user play the song themself, the application then should be able to determine if the user is playing it correctly or not, based on the information outputted from the individual components.

Testing the animations and front end of the application we will have a dataset of all possible data-structures that can be sent through from the back-end and run them through. Because each data-structure will be an independent animation attached with the correctly provided midi file, we will fully test every possible situation.

In order to test the playback correction we will input a correct playing of several songs with different amounts of unwanted noise and ensure that the algorithm is outputting that the song is correct. Then we will do the same thing with incorrect playings and ensure that it is counted wrong.

2.8 Safety Considerations and Security Considerations

Sheet-Vision is purely a software application, meaning our safety concerns are purely in based on network traffic, and making sure our application, when run on the client side, the user's machine will not be vulnerable to attacks that aim to give unauthorized administrator access to this machine. There will be no system receiving or transferring user information at all, meaning no sensitive data is being transferred from our machines, for now there is no need to use SSL encryption for our requests, since the information being transferred just data structures containing music and images of sheets of music. A plaintext protocol is sufficient for this.

2.9 Previous Work/Literature Review

2.9.1 Sheet Music Guide

Sheet Music Guide will provide a source for information about sheet music. We will be able to use this source in order to verify that the information about sheet music which we are using in our project is accurate, and inclusive of the different pieces of sheet music which will be useful to beginning musicians.[4]

2.9.2 Hough Lines

An important part of scanning a piece of sheet music is knowing where the staff and bar lines are so the positions of the notes and rests can be interpreted. An efficient way of detecting these lines is to use Hough Transforms to do line detection. The OpenCV library contains a built in algorithm for applying Hough Lines which will be used to find the staff and bar lines. A helpful tutorial is posted on their website for its use.

2.9.3 Template Matching

This tutorial provides a brief explanation of how to use template matching for detecting objects in an image. This is the main technique which will be used in the computer vision portion of the system. This technique will be used to find each type symbol which we will accommodate for. The method shown here however will not be enough for the completion of the note detection. Template matching as implemented in the OpenCV library is not size or rotation invariant. In order to accommodate for this we will need to employ other techniques to prepare the images for this search. [7]

2.9.4 React Development

This tutorial provides the basics needed to take advantage of what makes react different from other web development frameworks. It covers from how set up a react project, how to make components for our application, styling them, manipulating them by taking advantage of states, and finally how to run and test our application. The literature covers this by teaching the reader how to write a game of tic-tac-toe in react, making the information easy to digest, while giving the reader instant pay off on the work they are putting into learning this framework. [2]

2.10 Possible Risks and Risk Management

There is a risk that we may not be able to implement all of our planned features. To avoid this we need to make sure that we have a realistic project plan and schedule that we follow. We need to ensure we give ourselves enough time to research and execute our tasks. We also need to make sure that if we are not getting a good accuracy with what we want to do, we should reach out to the faculty advisor for help.

3 Project Timeline, Estimated Resources, and Challenges

3.1 Project Proposed Milestones and Evaluation Criteria

- ReactJS desktop application for windows.
- Application can read images from camera/file directory.
- Algorithm can recognize music notes in sheet music.
- Application can play the correct notes that come from the processed sheet music.
- Application can listen to user audio using the microphone.
- Algorithm can recognize what music notes are being played by the user based on audio input.
- Application can compare output from audio processing and image processing to determine if user is playing the correct notes.

3.2 Project Tracking Procedures

We will use Trello to assign tasks into two week sprints. These tasks will be assigned according to team roles, what has been accomplished in the past weeks, and our goals moving ahead. We will try to split up tasks such that they can be completed within a two week time period. Our team will meet weekly or once every two weeks, both as a team and with our faculty advisor to make sure we are on track with our sprints and deliverables.

3.3 Personnel Effort Requirements

- Expected 6 hours a week per team member.
- Constant availability for communication.
- Meeting participation and attendance.

3.4 Resource Requirements

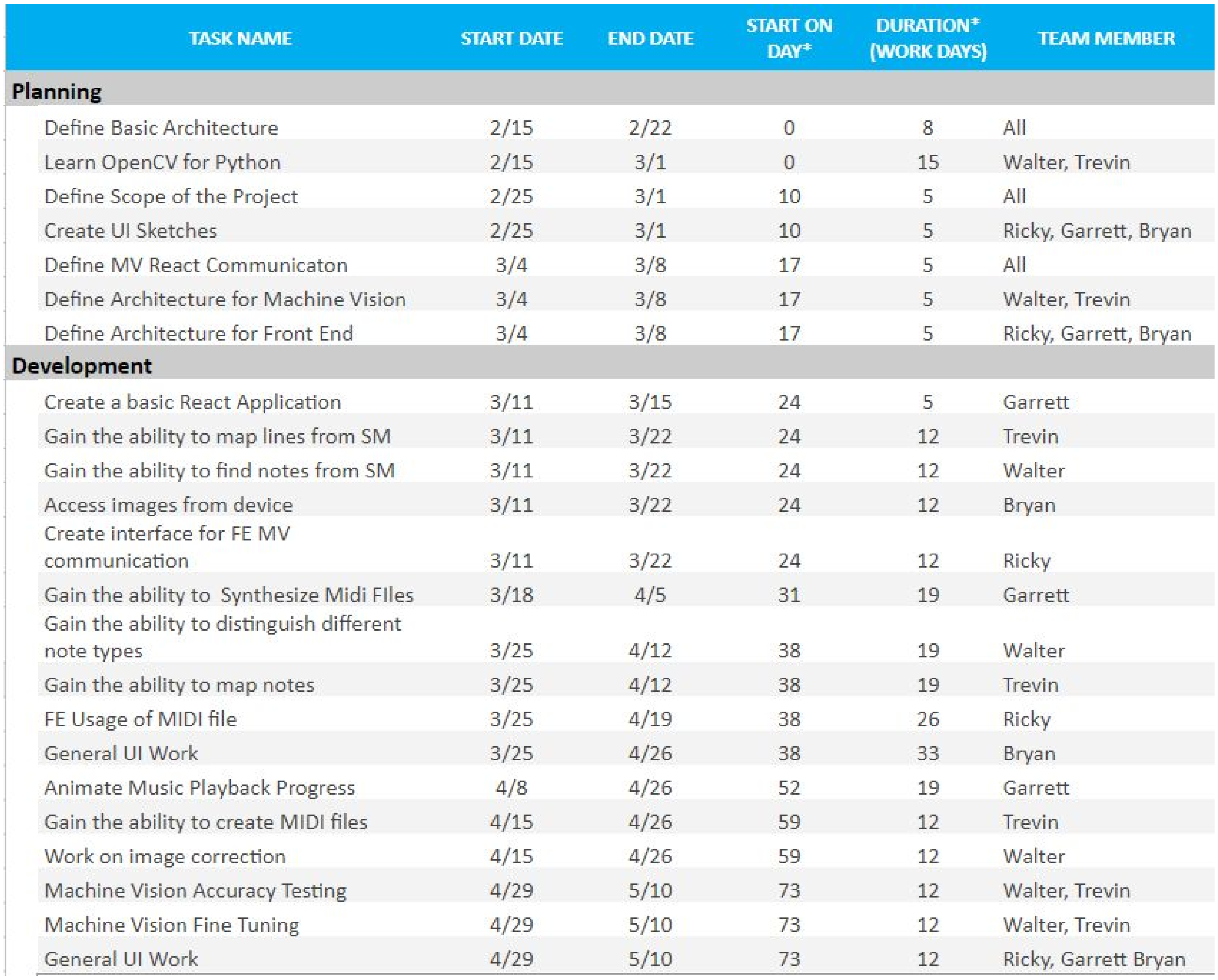
- Access to a computer running Mac OS for iOS and Mac OS testing.
- Access to a computer running Windows for Windows testing.
- Access to a computer running Linux for Linux testing.
- Access to an Android cell phone device for Android testing.

3.5 Financial Requirements

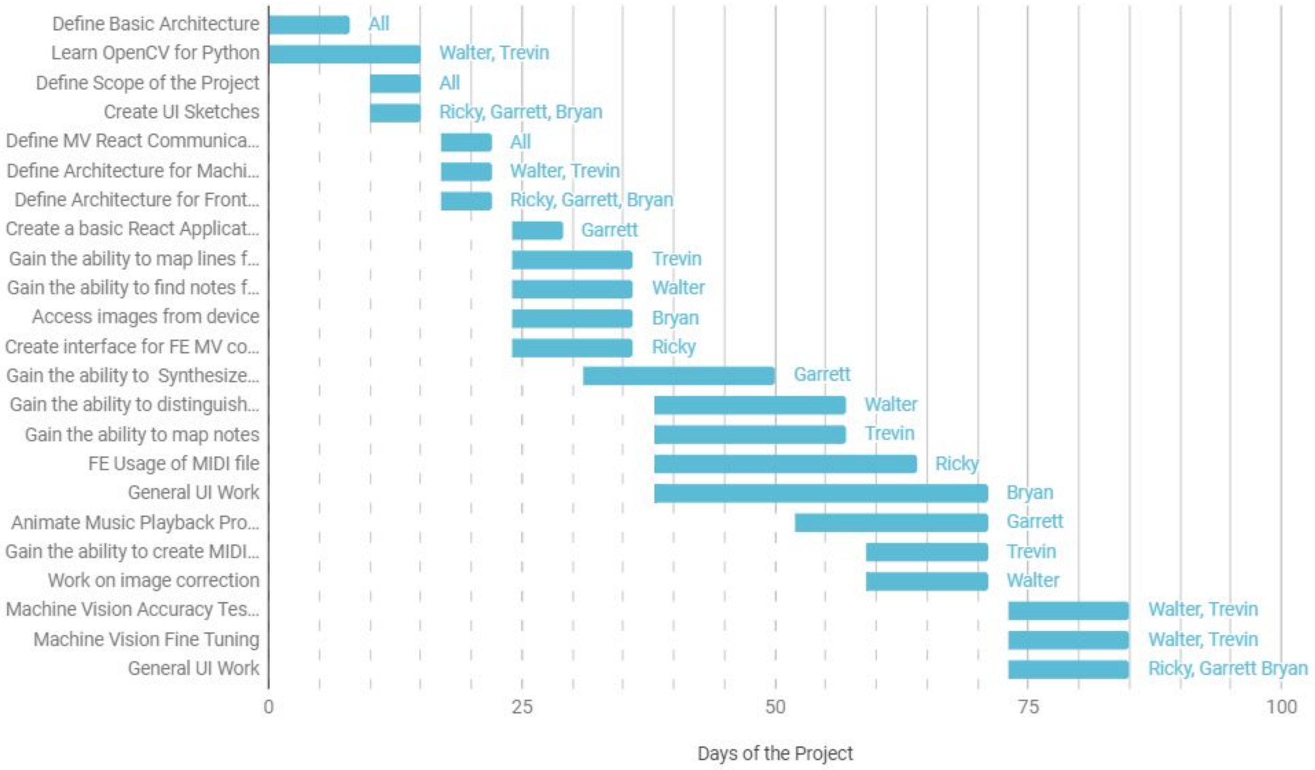
- Rolling Stone Easy Piano Sheet Music Classics
  - Link to Store Page
  - Price: ~$20
- eloam Mini Document Camera Scanner S300P
  - Link to Sore Page
  - Price: ~$130
- Audio-Technica AT2020 Cardioid Condenser Studio XLR Microphone
  - Link to Store Page
  - Price: ~$100

3.6 Project Timeline

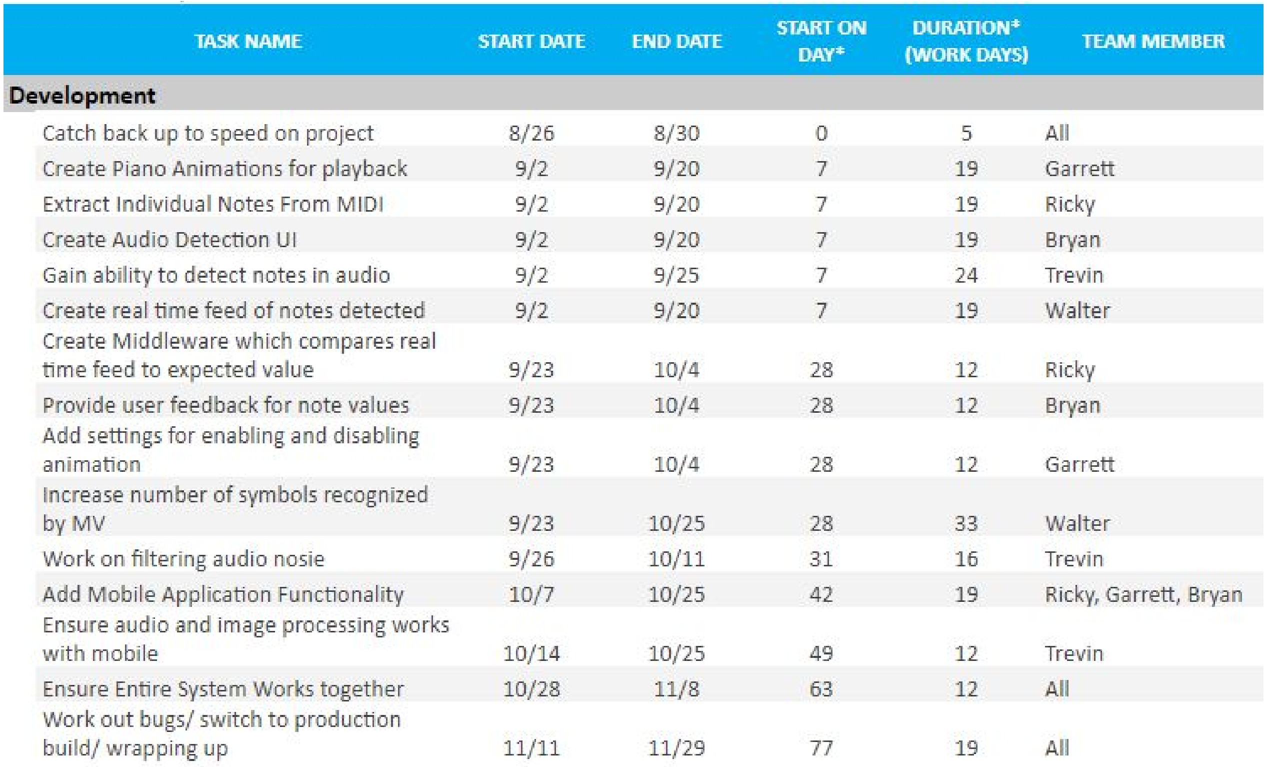
Spring 2019



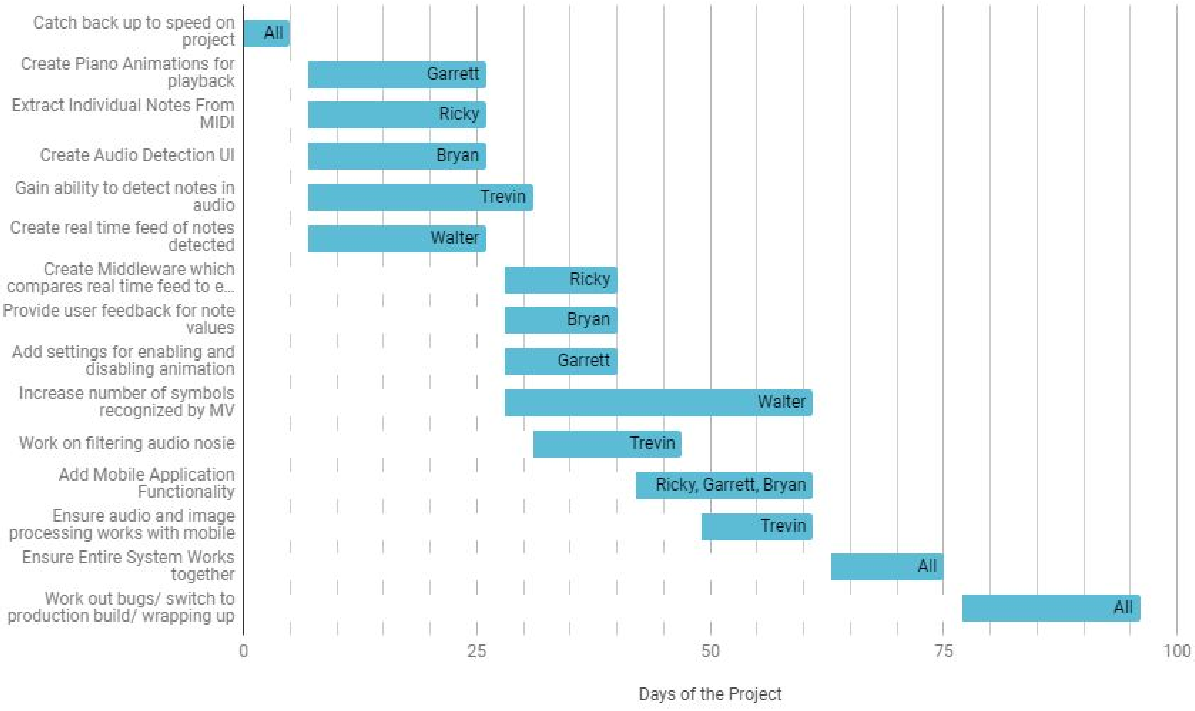
TASK NAME	START DATE	END DATE	START ON DAY*	DURATION* (WORK DAYS)	TEAM MEMBER
Planning					
Define Basic Architecture	2/15	2/22	0	8	All
Learn OpenCV for Python	2/15	3/1	0	15	Walter, Trevin
Define Scope of the Project	2/25	3/1	10	5	All
Create UI Sketches	2/25	3/1	10	5	Ricky, Garrett, Bryan
Define MV React Communicaton	3/4	3/8	17	5	All
Define Architecture for Machine Vision	3/4	3/8	17	5	Walter, Trevin
Define Architecture for Front End	3/4	3/8	17	5	Ricky, Garrett, Bryan
Development					
Create a basic React Application	3/11	3/15	24	5	Garrett
Gain the ability to map lines from SM	3/11	3/22	24	12	Trevin
Gain the ability to find notes from SM	3/11	3/22	24	12	Walter
Access images from device	3/11	3/22	24	12	Bryan
Create interface for FE MV communication	3/11	3/22	24	12	Ricky
Gain the ability to Synthesize Midi FIles	3/18	4/5	31	19	Garrett
Gain the ability to distinguish different note types	3/25	4/12	38	19	Walter
Gain the ability to map notes	3/25	4/12	38	19	Trevin
FE Usage of MIDI file	3/25	4/19	38	26	Ricky
General UI Work	3/25	4/26	38	33	Bryan
Animate Music Playback Progress	4/8	4/26	52	19	Garrett
Gain the ability to create MIDI files	4/15	4/26	59	12	Trevin
Work on image correction	4/15	4/26	59	12	Walter
Machine Vision Accuracy Testing	4/29	5/10	73	12	Walter, Trevin
Machine Vision Fine Tuning	4/29	5/10	73	12	Walter, Trevin
General UI Work	4/29	5/10	73	12	Ricky, Garrett Bryan


Define Basic Architecture
All
Learn OpenCV for Python
Walter, Trevin
Define Scope of the Project
All
Create UI Sketches
Ricky, Garrett, Bryan
Define MV React Communica...
All
Define Architecture for Machi...
Walter, Trevin
Define Architecture for Front...
Ricky, Garrett, Bryan
Create a basic React Applicat...
Garrett
Gain the ability to map lines f...
Trevin
Gain the ability to find notes f...
Walter
Access images from device
Bryan
Create interface for FE MV co...
Ricky
Gain the ability to Synthesize...
Garrett
Gain the ability to distinguish...
Walter
Gain the ability to map notes
Trevin
FE Usage of MIDI file
Ricky
General UI Work
Bryan
Animate Music Playback Pro...
Garrett
Gain the ability to create MIDI...
Trevin
Work on image correction
Walter
Machine Vision Accuracy Tes...
Walter, Trevin
Machine Vision Fine Tuning
Walter, Trevin
General UI Work
Ricky, Garrett Bryan
0
25
50
75
100
Days of the Project

Fall 2019



TASK NAME	START DATE	END DATE	START ON DAY*	DURATION* (WORK DAYS)	TEAM MEMBER
Development					
Catch back up to speed on project	8/26	8/30	0	5	All
Create Piano Animations for playback	9/2	9/20	7	19	Garrett
Extract Individual Notes From MIDI	9/2	9/20	7	19	Ricky
Create Audio Detection UI	9/2	9/20	7	19	Bryan
Gain ability to detect notes in audio	9/2	9/25	7	24	Trevin
Create real time feed of notes detected	9/2	9/20	7	19	Walter
Create Middleware which compares real time feed to expected value	9/23	10/4	28	12	Ricky
Provide user feedback for note values	9/23	10/4	28	12	Bryan
Add settings for enabling and disabling animation	9/23	10/4	28	12	Garrett
Increase number of symbols recognized by MV	9/23	10/25	28	33	Walter
Work on filtering audio nosie	9/26	10/11	31	16	Trevin
Add Mobile Application Functionality	10/7	10/25	42	19	Ricky, Garrett, Bryan
Ensure audio and image processing works with mobile	10/14	10/25	49	12	Trevin
Ensure Entire System Works together	10/28	11/8	63	12	All
Work out bugs/ switch to production build/ wrapping up	11/11	11/29	77	19	All



4 Closing Material

4.1 Conclusion

Throughout these two semesters we will work on creating a application that can scan through a sheet of music and create a simple animation of which keys on a piano you should press that correlate to the beats that will be playing. With this you can change features and after hearing what it should sound like and seeing what you should play; you can turn the application into practice mode and it can tell you if you are playing the correct notes.

4.2 References

[1] Duda, R. O. and P. E. Hart, "Use of the Hough Transformation to Detect Lines and Curves in Pictures," *Comm. ACM, Vol. 15* (January, 1972)

[2] Use React Native. *Use React Native.* http://www.reactnative.com/ (April 24, 2019)

[3] OpenCV Team. *OpenCV*. https://opencv.org/ (April, 24, 2019)

[4] Simplifying Theory. *Simplifying Theory.* http://www.simplifyingtheory.com/ (April 24, 2019)

[5] Zemsky David. *Sheet Music Scanner.* https://itunes.apple.com/us/app/sheet-music-scanner/id884984324?mt=8 (April 24, 2019)

[6] Playscore. *Playscore* https://www.playscore.co/ (April 24, 2019)

[7] Revison, Abid K and Mordvintsev Alexander. *Template Matching*. https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_template_matching/py_template_matching.html. (April 24, 2019)

[8] Revinson, Abid K. Mordvintsev Alexander. *Hough Line Transform***.** https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_houghlines/py_houghlines.html. (2013)