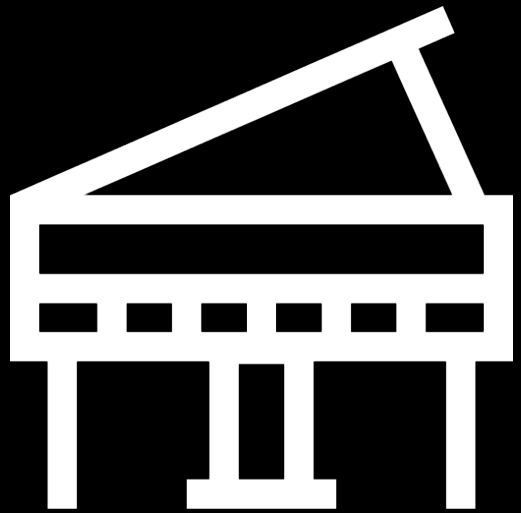
PIANOMAN

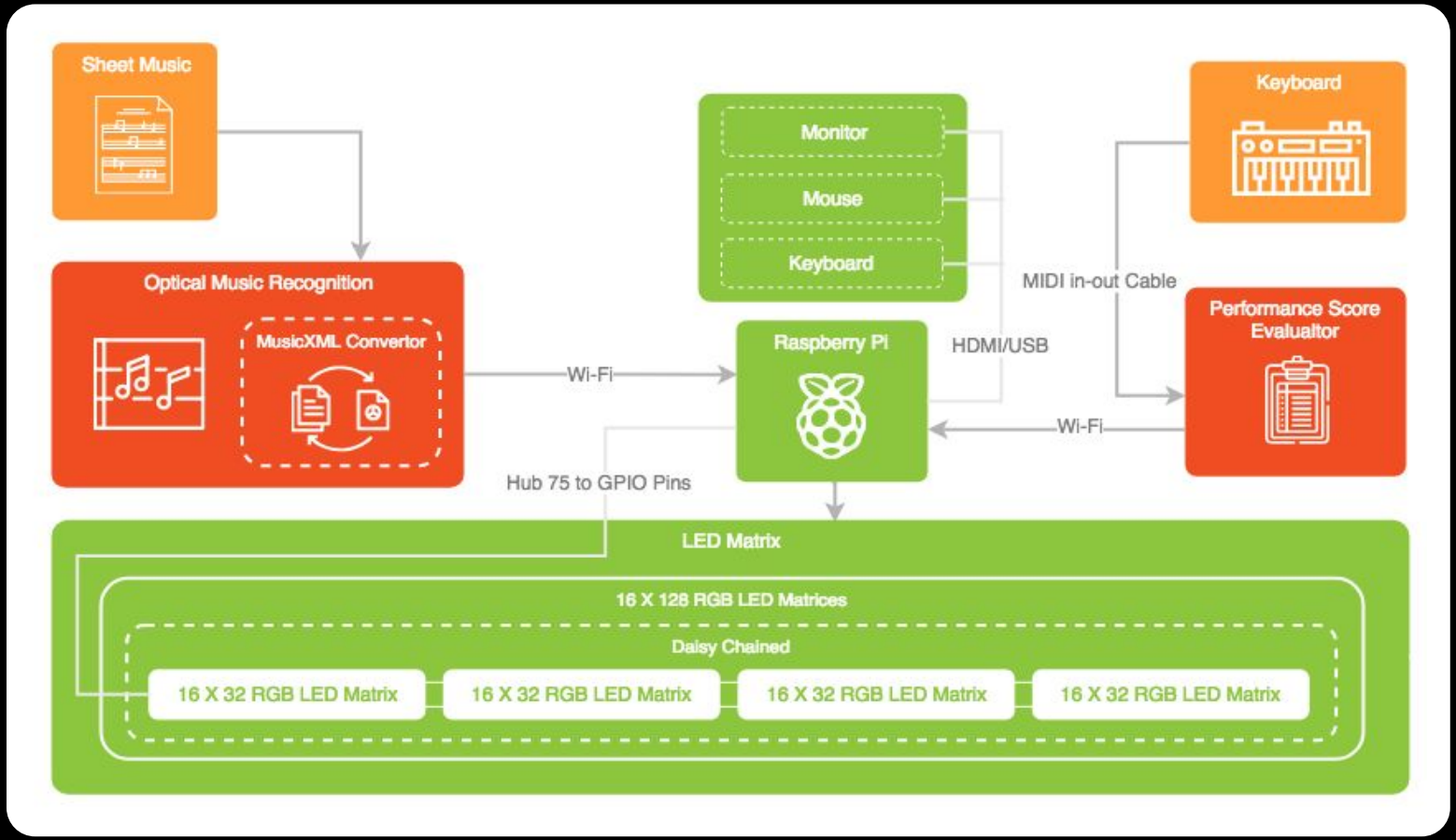
S19 ECE Capstone Design Project - Team D7
Final Presentation



Application Area

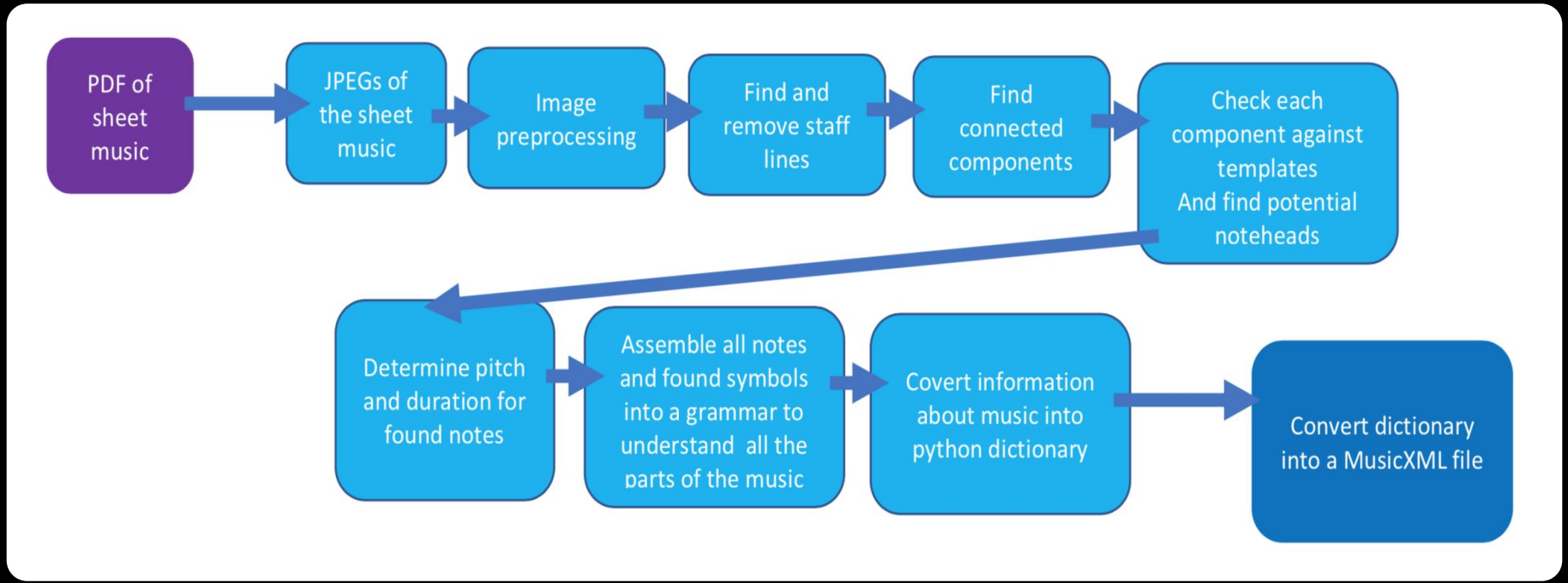
A Self learning tool for Piano players.

Reads sheet music of a song, lights up LED matrices indicating which notes they should play and generates an evaluation of their performance at the end.



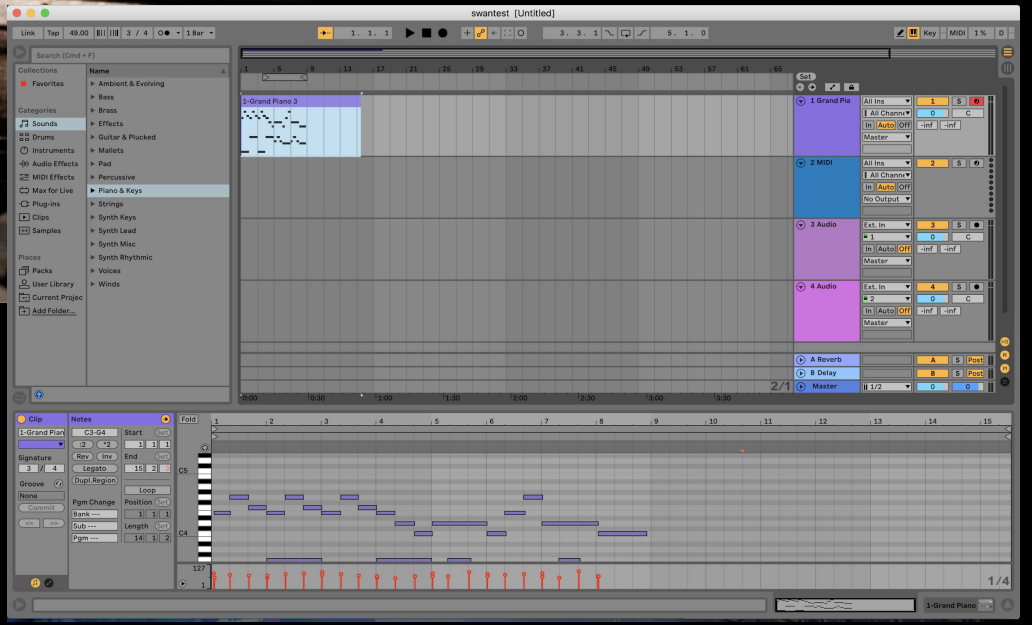
Solution Approach

Optical Music Recognition



GUI Application

Ableton Live 10



LED Matrix Chain Display - File Transfers

GUI sends
MusicXML to Pi

GUI press
START

Start Record,
Play song

End Record,
wait for score

Display score

Complete Solution

Metrics and Validation

1. Optical Music Recognition

Requirement: Correctly identifies 90% of all noteheads, rests, accents (dots, sharps, flats, naturals), time signature, clef symbols, and key signature to produce a musicXML file. Can complete OMR in 1 minute/120 recognizable items.

Validation:

- Data: Digital sheet music pdfs
- Test: Compare original pdf to output from SoundSlice (reads in musicXML into sheet music)

Metrics and Validation

2. Raspberry Pi - LED Matrices Chain

Requirement: Displays 99% of notes from the MusicXML files sent by OMR side GUI at the right time for the right duration. Takes less than 3 seconds each to parse files (musicXML, start, end).

Validation:

- Data: MusicXML files from MuseScore
- Test: Compares the start times and durations of segments produced on the LED Matrix chain against calculations from the Song’s Sheet Music PDF

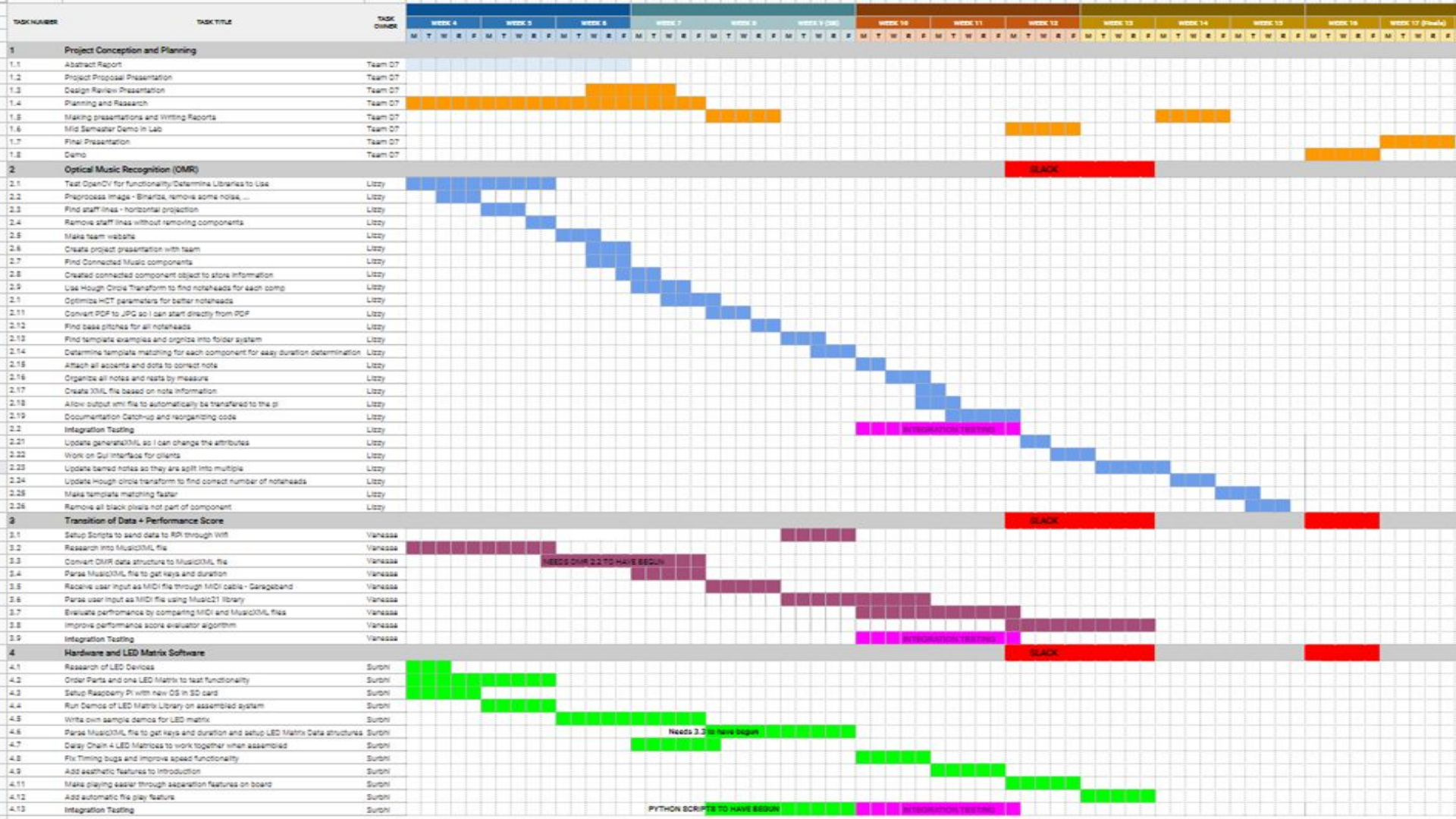
Metrics and Validation

3. Performance Score Evaluator

Requirement: Evaluates the user's performance score and sends to the RPi in less than 3 seconds after end recording and exporting to MIDI file.

Validation:

- Data: MIDI/XML from MuseScore
- Test: Check if the performance score evaluator correctly calculates the performance score and sends to RPi


TASK NUMBER
TASK TITLE
TASK OWNER
WEEK 4
WEEK 5
WEEK 6
WEEK 7
WEEK 8
WEEK 9 (SB)
WEEK 10
WEEK 11
WEEK 12
WEEK 13
WEEK 14
WEEK 15
WEEK 16
WEEK 17 (Finals)
1 Project Conception and Planning
1.1 Abstract Report Team 07
1.2 Project Proposal Presentation Team 07
1.3 Design Review Presentation Team 07
1.4 Planning and Research Team 07
1.5 Making presentations and Writing Reports Team 07
1.6 Mid Semester Demo in Lab Team 07
1.7 Final Presentation Team 07
1.8 Demo Team 07
2 Optical Music Recognition (OMR)
SLACK
2.1 Test OpenCV for functionality/Determine Libraries to Use Lizzy
2.2 Preprocess Image - Binarize, remove some noise, ... Lizzy
2.3 Find staff lines - horizontal projection Lizzy
2.4 Remove staff lines without removing components Lizzy
2.5 Make team website Lizzy
2.6 Create project presentation with team Lizzy
2.7 Find Connected Music components Lizzy
2.8 Created connected component object to store information Lizzy
2.9 Use Hough Circle Transform to find noteheads for each comp Lizzy
2.1 Optimize HCT parameters for better noteheads Lizzy
2.11 Convert PDF to JPG so I can start directly from PDF Lizzy
2.12 Find base pitches for all noteheads Lizzy
2.13 Find template examples and orgnize into folder system Lizzy
2.14 Determine template matching for each component for easy duration determination Lizzy
2.15 Attach all accents and dots to correct note Lizzy
2.16 Organize all notes and rests by measure Lizzy
2.17 Create XML file based on note information Lizzy
2.18 Allow output xml file to automatically be transfered to the pi Lizzy
2.19 Documentation Catch-up and reorganizing code Lizzy
2.2 Integration Testing Lizzy
INTEGRATION TESTING
2.21 Update generateXML so I can change the attributes Lizzy
2.22 Work on Gui interface for clients Lizzy
2.23 Update tamed notes so they are split into multiple Lizzy
2.24 Update Hough circle transform to find correct number of noteheads Lizzy
2.25 Make template matching faster Lizzy
2.26 Remove all black pixels not part of component Lizzy
3 Transition of Data + Performance Score
SLACK
3.1 Setup Scripts to send data to RPi through Wifi Vanessa
3.2 Research into MusicXML file Vanessa
3.3 Convert OMR data structure to MusicXML file Vanessa
NEEDS OMR 2.2 TO HAVE BEGUN
3.4 Parse MusicXML file to get keys and duration Vanessa
3.5 Receive user input as MIDI file through MIDI cable - Garageband Vanessa
3.6 Parse user input as MIDI file using Music21 library Vanessa
3.7 Evaluate performance by comparing MIDI and MusicXML files Vanessa
3.8 Improve performance score evaluator algorithm Vanessa
3.9 Integration Testing Vanessa
INTEGRATION TESTING
4 Hardware and LED Matrix Software
SLACK
4.1 Research of LED Devices Surbhi
4.2 Order Parts and one LED Matrix to test functionality Surbhi
4.3 Setup Raspberry Pi with new OS in SD card Surbhi
4.4 Run Demos of LED Matrix Library on assembled system Surbhi
4.5 Write own sample demos for LED matrix Surbhi
4.6 Parse MusicXML file to get keys and duration and setup LED Matrix Data structures Surbhi
Needs 3.2 to have begun
4.7 Daisy Chain 4 LED Matrices to work together when assembled Surbhi
4.8 Fix Timing bugs and improve speed functionality Surbhi
4.9 Add aesthetic features to introduction Surbhi
4.11 Make playing easier through seperation features on board Surbhi
4.12 Add automatic file play feature Surbhi
4.13 Integration Testing Surbhi
PYTHON SCRIPTS TO HAVE BEGUN
INTEGRATION TESTING

Lessons Learned

1. Order your parts as early as possible to conduct functionality tests. Order extra in case the parts go out of stock later.
2. Make a detailed schedule by reducing each task into list of agreeable goals.
3. Think of a design project that will have enough independent work for three people but will come together at the end and not end up like three separate projects.