**Software Architecture Specification**

**Report #3**

**Team Name:**

Mathematical Maestros

**Team Members:**

Jonathan Hasty

Jacob Coomes

Matthew Branstetter

**Breakdown of individual contributions**

*(subject to change over course of project)*

Jonathan Hasty

1. Team lead

2. Documentation

3. Coding

Jacob Coomes

1. Coding

2. Documentation

3. Testing

Matthew Branstetter

1. Coding

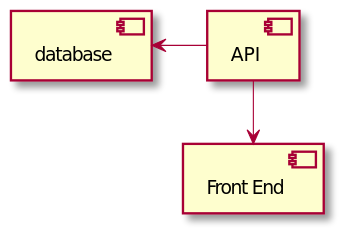
2. Documentation

3. Testing

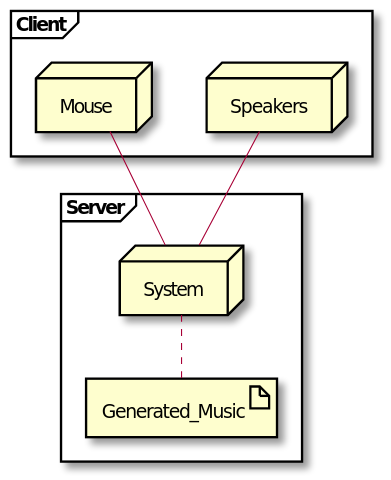
Software Architecture Specification

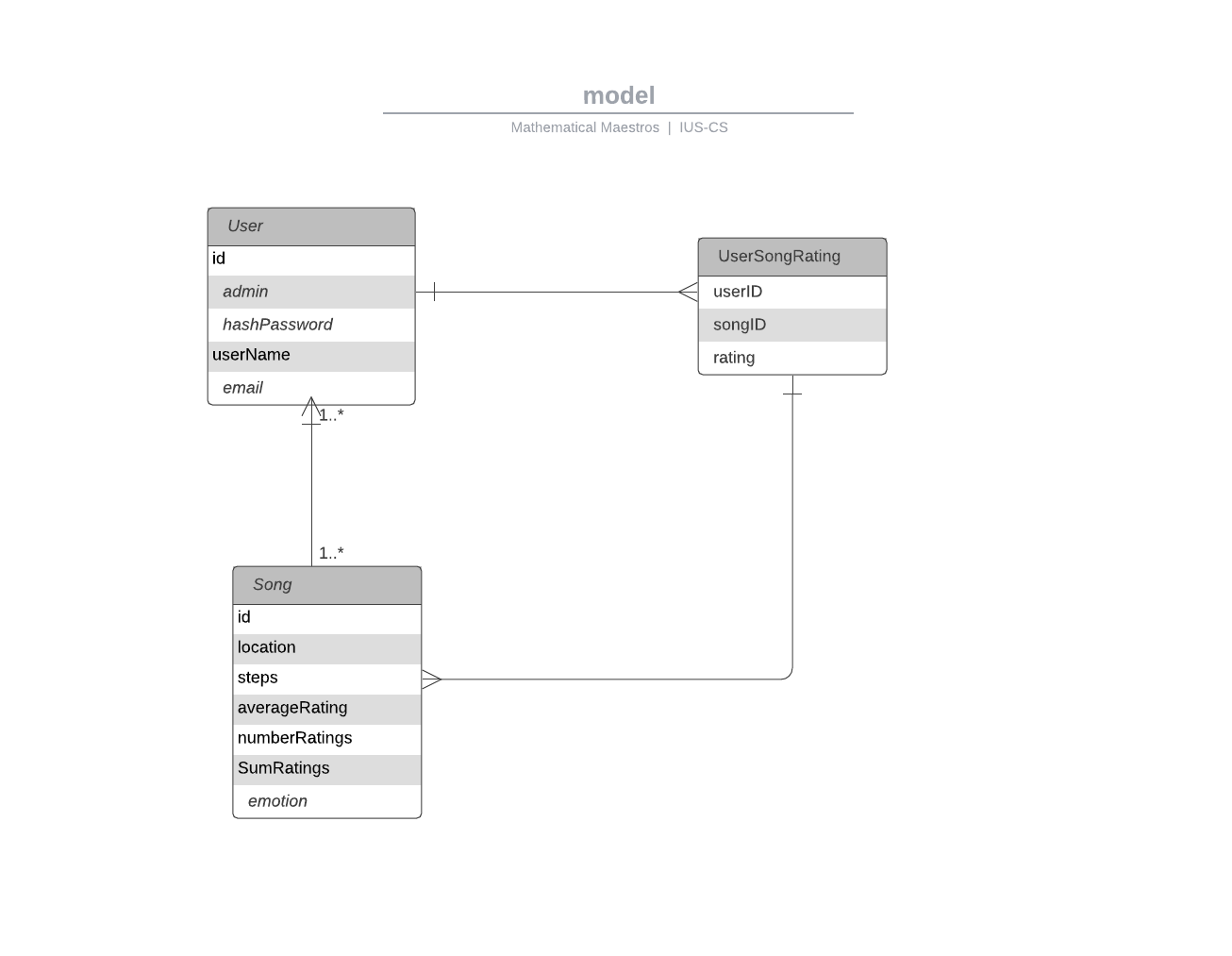
I. PROPOSED SOFTWARE ARCHITECTURE

1. Overview
2. The system will be a web application using React, a javascript user interface library, for the front end; and Flask, a python micro-framework for the back end.
3. Subsystem Decomposition
4. Our product will be based on a client-server architecture. The server side’s responsibility is to compose music, store the file location/ song elements and serve to the front end. The client side’s responsibility is to fetch from the backend API and play music through the browser.



1. Hardware/ software mapping --- DIAGRAM HERE MATTHEW
2. The hardware that will be used for this project is the mouse and speakers. The mouse is to select and press the play button for music. It will also be used to select the mood of music to be generated via radio buttons. The speakers are used to play the music that is selected.



1. Persistent Data Management
2. Persistent data will include user login information (username and hash-password), and user e-mail address. Non-persistent data will include the .midi file as well as form information concerning how the music will be generated.

Project

/\_ Flask-Server

/\_ Library

/\_ song1.midi

/\_ song2.midi

/\_ venv

/\_ app.py

/\_ db.sqlite

/\_ routes.py

/\_models.py

/\_ Front-End

/\_ App.js

/\_ App.css

1. Access Control and Security
2. Users will be able to play, skip tracks, and download music. They will need to login via a username and hash-password system to access the system. Administrators will have access to all user features as well as direct access to the database.
3. Global Software Control
4. Our program will be event-driven, because the user will press a button for the music to either play, pause, skip, and download.
5. Boundary Conditions
6. The system will start up by going to a login/ join screen that the user must join and will later have to use the login screen. The user can then use the app to import their music and let the computer AI to computerize the song. When the user is done with the app and click the close button, they will be first asked to log out of the app first.

II. NINE BASIC COMPONENT TYPES

1. Use Cases
2. After the music has been generated there will be a play/ pause button that the user will use to start/ stop the song. The download button will ask the user to find the path that the music can be downloaded to.
3. Function
4. The functions for this program will be the play/ pause button that will allow the music to play or stop. The rewind/ fast forward button allows the music to go back or go forward. The download button will grab the song from the server and put it in the user’s path directory.
5. Triggers
6. When the user runs out of space in their account and wants more space to create more music then they can go to a “Buy More Space” button for $--.-- and it will transact the money to our products account. When they want to extend the use of our product they will click the “Extend My Use” button which is $--.-- which will go to our products account.
7. Data Stores
8. The data that’s going to be on the cloud can be moved to the user’s computer when they want to update their music list and save it on the cloud when they're done.
9. Data Flows
10. The data will be bi-directional between client and server with the client sending music parameter information and the music itself. The server will send the processed data back to the user for streaming and later on downloading which is also a bi-directional flow.
11. Data Elements
12. Form data: this describes how the machine learning model will generate results.
13. User data: this includes user login, hash-password, email address and a breakdown of time listened in seconds for each genre.
14. Song data: This is an extension of form data that includes the song’s file path.
15. Processors
16. The users are going to fill out a form to figure out what the next song should be. The user will be allowed to download the computer generated song for them to keep. The computer will process the user request and generate songs.
17. Data Storage
18. The data that is going to be stored for the user’s music that is created is going to be on the cloud.

I. Data Connections

1. When the user wants to ask a question on why the system is not working as it should or it sounds too much like the original music they put to be computerized. When we find that there is something wrong with the system, we will email the people that signed up for our product and we will tell them that we will be updating the system or fixing bugs that the user’s are encountering.

J. Actors/ External Entities

1. The system will interface with Spotify’s API as a means to classify song data by emotion which will eventually be used in the training data set for the machine learning model.

**Jacob Coomes**

**6012 Heil Rd. Email: coomes.jacob@yahoo.com**

**Henryville, IN 47126 Phone: 812-252-9284**

**Objective:**

Obtain position at SamTech as programmer

**Employment History:**

· Short Order Cook Dairy Queen

Sellersburg, IN

8/16/16-1/31/17

o Duties:

§ Utilize Grill and Deep Fryer to cook various types of food

§ Take the trash out

§ Clean up work area

· Wholesale Employee Walnut Ridge

Jeffersonville, IN 4/18/17-Present

o Duties

§ Water Plants

§ Put Merchandise in Customer’s Cars

§ Take out trash

§ Clean and weeded the Outside area

§ Landscape Assistance

**Education:**

**Henryville High School**

· Address: 213 N Ferguson St, Henryville, IN 47126

· 12th Grade

o Prossor:

o Computer Programming

§ HTML5

§ Visual Basic 6

§ Visual Basic 2010

§ Javascript

**Indiana University Southeast**

· Address: 4201 Grant Line Rd, New Albany, IN 47150

· Major: Bachelor Degree in Computer Science

· Minor: Math

· August 24, 2017 – Present Currently Junior

· Computer Programming

o F#

o C++

o Java

o Assembly Language

· General Education

o Calculus 1 & 2

o Speech

o Art

**Academic Honors:**

· Henryville High School National Honor Society

· Henryville High School Technical Diploma

**Academic Scholarships:**

· Henryville High School Key Club Award

· Orrin E Weber Special Effort Award

· The Terry Hill Higher Education Award

· Willis Drake Helping Hands Award

**Volunteer Work**

· Saint Francis Xavier Church

o Fish Fry

o Cleaning/ Maintenance

o Live Nativity

o Septemberfest

o Usher

o Halloween Trunk or Treat

**Traits**

· Hard-Working

· Focused

· Organized

· Trustworthy

**SIMEON BRANSTETTER**

**Phone:** (812) 406-6814 || **Email:** matbrans@twc.com || **Github:** github.com/SimeonBranstetter

**Objective:** To obtain real world experience in my field of discipline that will enhance my skills in software engineering and application programming as well as computer architecture and data management.

**EDUCATION**

**Indiana University Southeast – New Albany, IN** May 2021

Bachelor of Science in Computer Science – Math & Science Track GPA 3.7/4.0

Minor in Mathematics

**Relevant Coursework:**

**Completed:** Object Oriented Programming, Discrete Mathematics, Computer Structures, Data Structures, Programming Languages, Software Engineering, Computer Networking, Analysis of Algorithms, Calculus 1, Calculus 2, Calculus 3, Linear Algebra, Elements of & Probability

**Additional Coursework:**

**Completed:** Survey of Economic Issues & Problems, Public Speaking, Reading-Writing & Inquiry, Writing in the Arts & Sciences, Principles of Chemistry, Experimental Chemistry

**PROJECTS AND EXPERIENCE**

**IUS Software Engineering Class Group Project** January 2020 – May 2020

* Small development team using RESTful and other architectural styles for application programming in a team to create an application.
* Using Kotlin via android studio a phone application was programmed to read album records from taking a picture of the cover and then sending back pertinent information of the album including songs, writers, producers, etc.
* Google Cloud API was used for the image recognition along with the Discogs API and database for searching up the appropriate album info from the parsed data received from Google Cloud API.

**IUS Computer Networking Class Group Project** August 2020 – December 2020

* Small development team using language, IDE, API/Framework of choice to develop a game that would use networking capabilities for internet communication via 2 or more players.
* Lua via Sublime Text Editor along with a Lua 2d game creation API called LOVE Engine was used to develop a crude version of Battleship using UDP socket connections for internet communication.

**Clark County Indiana 4-H Robotics Club** February 2013 - July 2019

* Physically constructed and programmed with Lego Mindstorms in a group to understand basic robotics, engineering, and programming knowledge.
* Last couple of years spent as a club leader for the robotics group teaching children about basic programming knowledge along with hardware communication.

**SKILLS**

**Languages:** Proficiency in Java, C++ || Comfortable with F#, Lua || Worked with Python, C#, C, Kotlin

**Dev Tools:** Git, IntelliJ IDEA, Netbeans, Visual Studio Code

**AWARDS/HONORS**

**Chancellor’s List - Two Semesters** || **Dean’s List - Two Semesters**