# CSCI 5253 Project Proposal

**Project Title:** Music Identifier

**Participant:** Mengyu Wu

**Project Goals:** When you happen to hear some melodies that fit your taste, knowing its’ name, singer, year and more information really helps if you want to add it to your playlist. In this project, I prepare to make such an online service: you give a melody segment, you get all its details back. And you can also add your favorite songs with metadata to help others. There have been some commercial services like Shazam, but the procedures to build our own system is still a good practice. The service would have following functions:

1. You upload audio record of the raw soundtrack or just your hum. The server would generate a fingerprint and compare it with records in a pre-built database. If found, the meta data would be returned to user, include name, singer and so on.
2. You could also upload audio file with meta data to add your favorite songs into our database. The server would also generate a fingerprint and use it for further identification.
3. If the fingerprint match is ambiguous, the server will try to extract the lyrics to help distinguish if we have the lyrics in our database.

**Components:**

* Opensource package to generate fingerprint (<https://github.com/itspoma/audio-fingerprint-identifying-python>)
* User interface: NGINX + template from webflow.com
* REST server based on flask
* Database: Cassandra
* Google cloud speed-to-text service
* Message Queue: RabbitMQ
* Docker and Kubernetes

**Architectural Diagram:**

Diagram

Description automatically generated

**Workflow:** All servers deployed in Kubernetes engine in Google Cloud. Users could visit the website and send their audio file to web server. Then, web server generates rest request based on user’s submission and send to REST server. REST send the request to RabbitMQ and waiting for the result back through another message queue. Workers received request from RabbitMQ, calculate fingerprint, and compare with database’s records. Add new record or retrieve exist meta data based on request. Speech-to-Text api might be used to generate lyrics to compare with records if necessary. The result from workers send back to REST server through RabbitMQ, and finally REST server send response back to Web Server and display result to users.

**Debug and test:** A log pod would be generated to receive logs from all pods through RabbitMQ. The whole system would be deployed in local machine first to make sure the basic task works for tens of songs. Then it would be moved to google cloud and run larger scale test.

During each test, the initial database would be built using songs download from website. The audio that user uploaded would consist of records of raw mp3 file using smartphone, records of mp3 files in other version, records of human hum.

**Cloud technologies used:**

1. REST API: user interface send request through it
2. Message Queues: offer easy and clear control when assign tasks to fingerprint workers
3. Database: save all fingerprints and metadata in key-value format, add or retrieve values based on requests
4. Containers: whole service would be deployed using Docker and Kubernetes.