MIDDLE EAST TECHNICAL UNIVERSITY

NORTHERN CYPRUS CAMPUS

Computer Engineering Program

CNG 495 – CLOUD COMPUTING

25-26 FALL – TERM PROJECT

CLOUD BASED MUSIC SHARING & RATING PLATFORM

Name: Ahmet Caner

Surname: Karaca

ID: 2585123

Project: Cloud-Based Music Sharing & Rating Platform

TABLE OF CONTENTS

LIST OF FIGURES

CLOUD-BASED MUSIC SHARING & RATING PLATFORM–PROJECT OVERVIEW

Cloud-Based Music Sharing & Rating Platform is a free and open web-based application that enables users to upload their original musical work. The primary objective of this project is to guide the up and coming artists along their journey to gain visibility and announce their talent with other musicians and global audience. With this project, the target is to establish a bridge between the intended consumer audience and independent musicians. For example, when a newly brought artist upload a work, they will have a chance to be discovered all around the globe.

One of the primary concerns of this project is the user authentication and the data security. Since the uploaded data is considered as copyrighted material, protecting the artist confidentiality and ownership rights plays a crucial role. All user interactions, including account creation, authentication, and music uploads, ratings-commentings, are secured through cloud-based security mechanisms

The other key functionalities may include:

- costless music uploading,
- seamless music streaming that is uploaded by other artists,
- listeners having the chance to rate the artist that they desire them to be in the weekly top 10 chart,
- generation of the 'weekly top 10' chart according to the user ratings,
- displaying playlists according to the users recent music taste,
- viewing the artists page, commenting and giving feedback to artists,
- receiving notifications when a track releases from a followed artist

Overall, this project aims to illustrate a real-world scenario while utilizing cloud-based application designs and solutions. In addition to that, this project will demonstrate the Software as a Service (SaaS) model from Cloud delivery models.

IMPLEMENTATION & TECHNOLOGIES USED

The implementation will be carried out utilizing the cloud native full stack application approach. The application will provide smooth integration of the front-end, back-end, database and cloud services. This integration will be accomplished by sending RESTful API requests from front-end to the back-end to process responds in coordination with cloud based storage, database and environment. This flow of design allows the application to be scalable, easy to use and efficient.

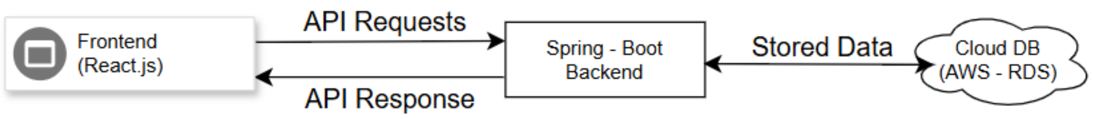

Figure 1:Implementation

Front-End (React.js)

Front-end side of the project will be developed using **JavaScript** framework **React** to ensure user-friendly, highly interactive user interface design. This side will manage all incoming user requests and interaction types by interacting with the backend services. In the development process using React, various aspects will be used such as, **React Routers** to navigate through the pages, **Axios** for efficient backend – frontend interaction and API request management. With this structure, modern and interactive web design will be achieved.

Back-End (Spring Boot)

Back-end side of the project will be developed using **Java** framework **Spring-Boot**, used for efficient web application development and robust back-end **RESTful** service management. This layer will provide a solid foundation for essential functionalities such as, user authentication, music upload and retrieval and interacting with database. In terms of data storage, back-end will interact with **PostgreSQL** database hosted on **AWS RDS** to store user data. This database is also integrated with the **AWS S3** for music file storage.

Database (PostgreSQL / AWS RDS)

PostgreSQL is a relational database management system that will be utilized in the project for its scalability and reliability. PostgreSQL will be hosted at **AWS RDS (Relational Database Service)**. Expected information that the database will hold is as follows: user credentials, music metadata, ratings, comments, playlists etc. Please note that this structure may vary as the project is being developed.

Cloud Services (AWS)

This project will utilize the **Amazon Web Services** infrastructure. This aspect can be an example of **Infrastructure as a Service (IaaS).** For example, the deployment of the backend services will be on **AWS EC2 (Elastic Compute).** AWS EC2 provides a virtual environment for backend hosting. In addition to that, storage of music files (mp3s) will be managed by **AWS S3** offering scalability and efficiency. As an integrated environment, this cloud services will provide secure and high performance system.

DIAGRAMS:

1)Use Case Diagram:

Below shows the use case diagram containing actors, listener and artist and their corresponding use cases:

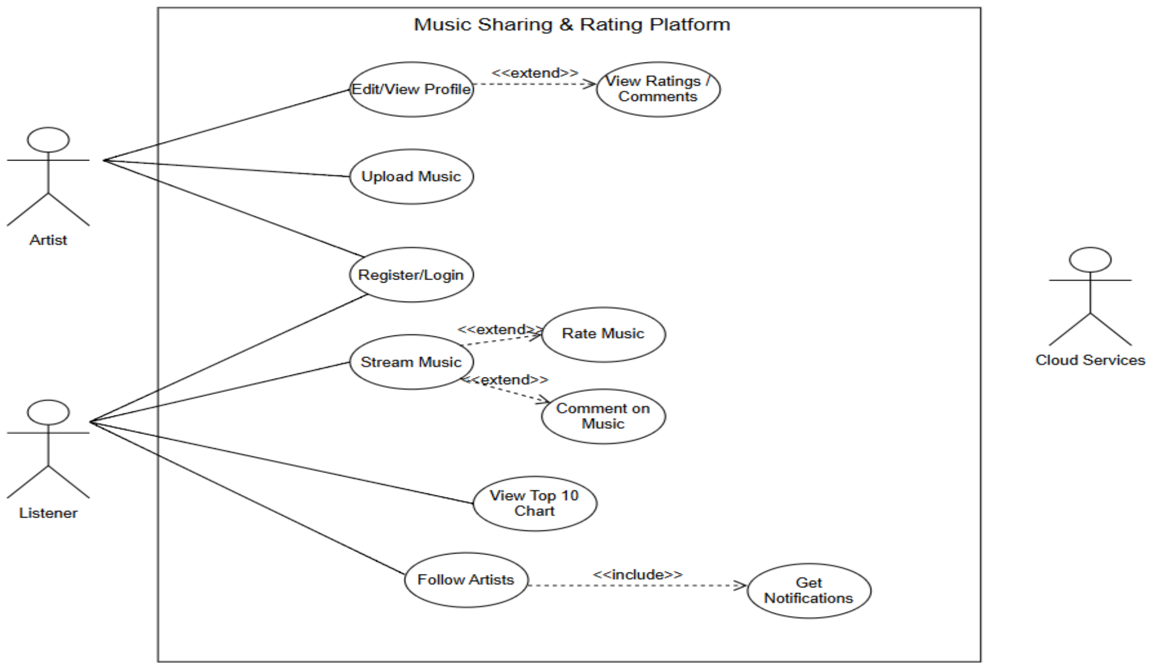

Figure 2: Use Case Diagram

2) Data Flow Diagrams:

2.1)Context Level DFD

Below is the context level data flow diagram which explains the data exchange between agents in a broader way. Main process exchanges various data with cloud database, artist and listener.

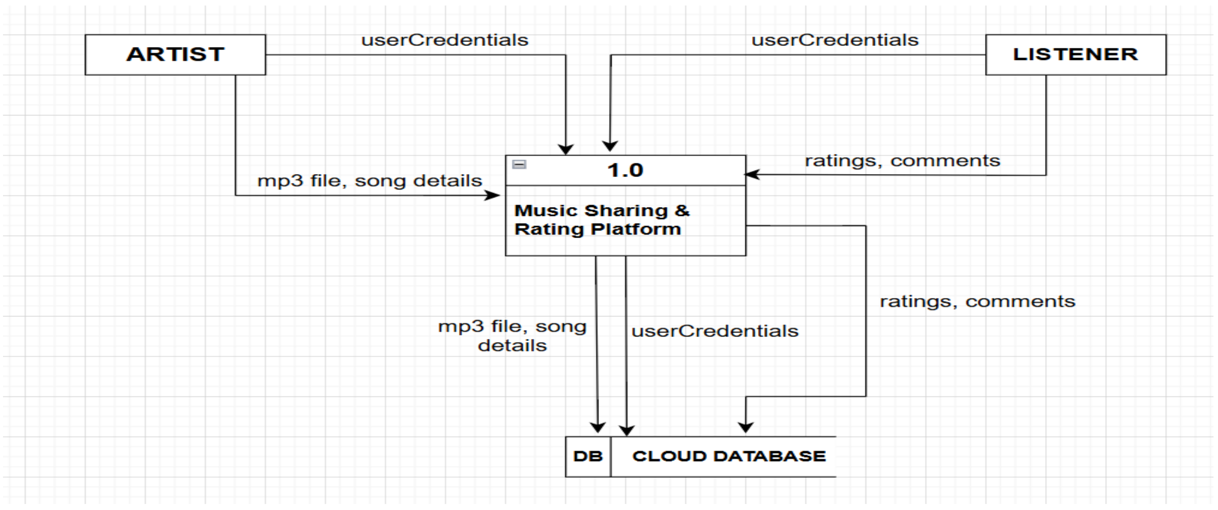

Figure 3: Context Level DFD

2.2) Level 0 DFD

In level 0 DFD, main process is divided into 6 different subprocesses in detail and shown below.

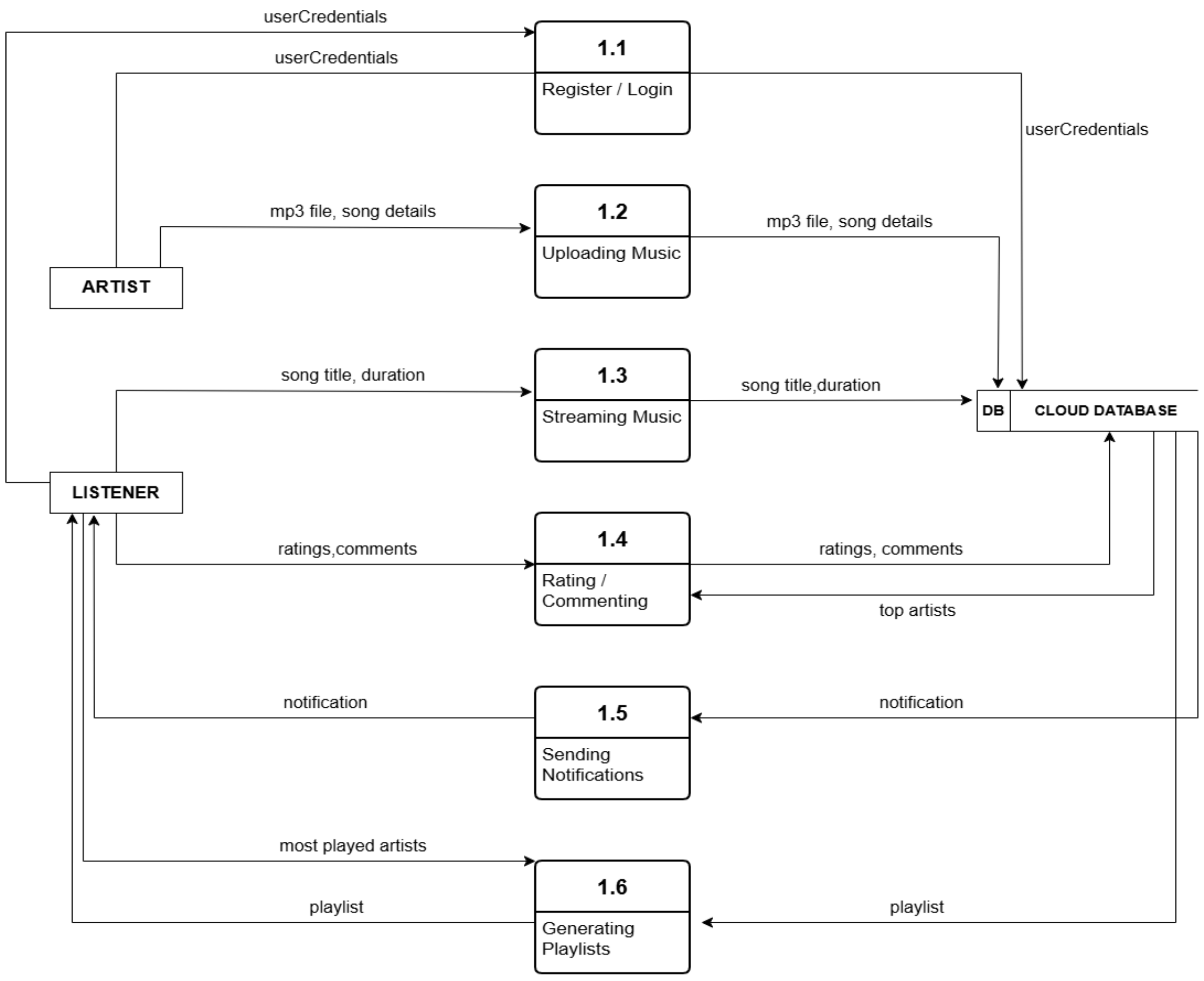

Figure 4: Level 0 DFD

3) Activity Diagrams:

3.1) Activity Diagram for Uploading Music

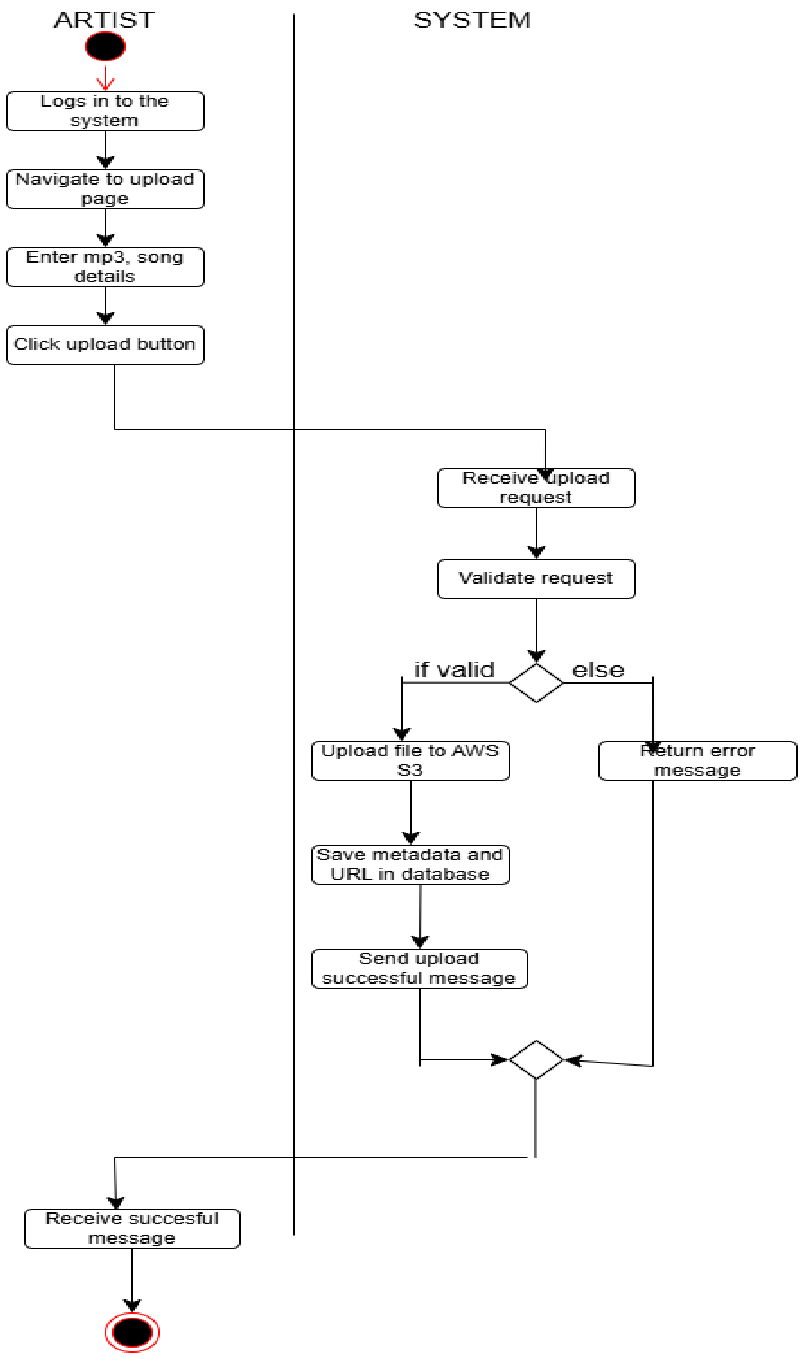

Figure 5: Activity Diagram for Uploading Music

3.2)Activity Diagram for Streaming Music

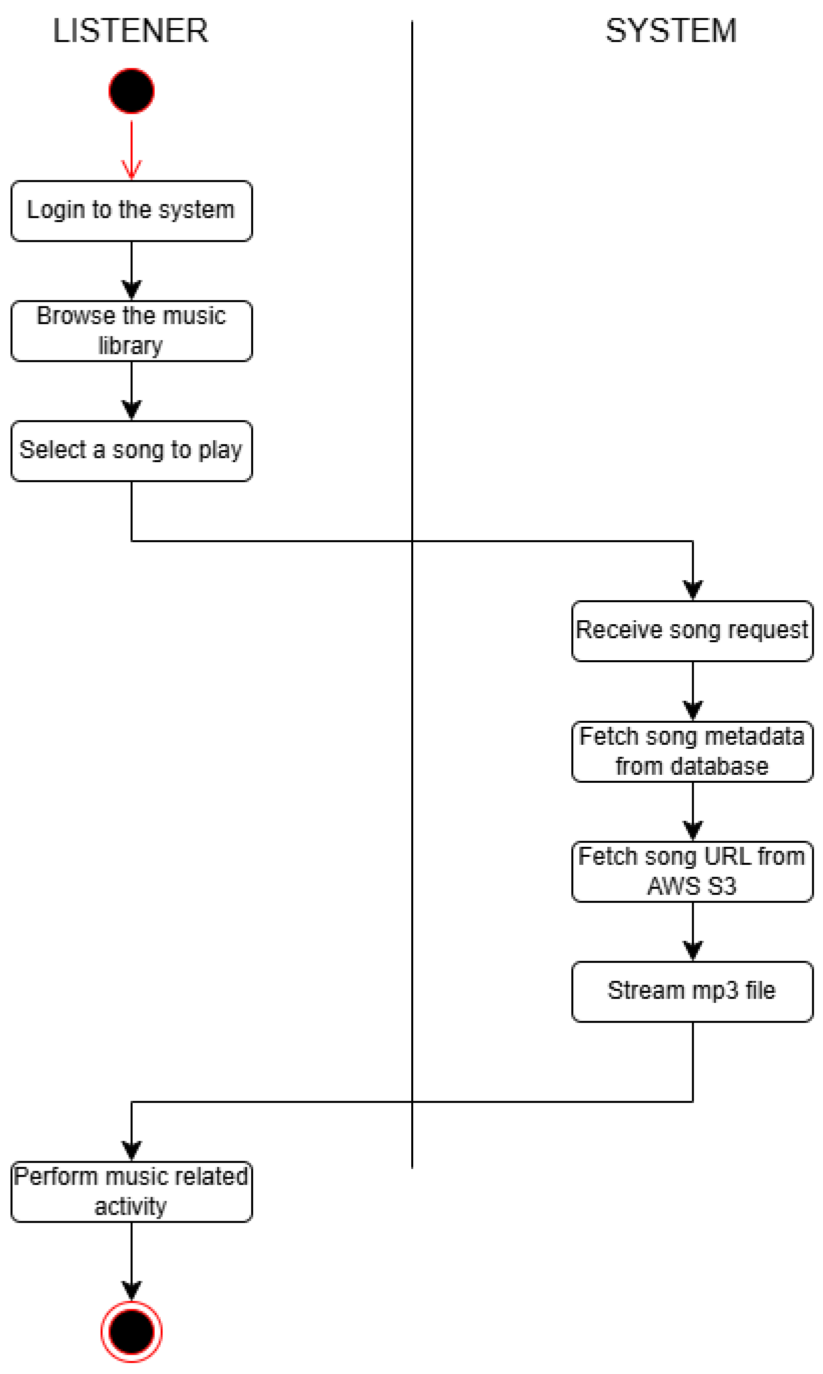

Figure 6: Activity Diagram for Streaming Music

DATA TYPES

In this project, following data types will be used:

Text:

- Song names and URLs: Song names and their corresponding storage URLs will be stored in the text/string format.
- User credentials: Security will be assured by login process with username and password.
- Database: Database will store the following data.
    - username
    - password
    - song name
    - song URL
    - ratings
    - comments
    - playlist names etc.

Numeric:

Primary keys in the database such as songID, userID and other variables such as song duration etc. will be stored in numeric format

Binary:

MP3 files that is to be stored in the AWS S3 storage, can be considered as binary data.

COMPUTATIONS

In this project, several computations will be managed by backend server side, which can be found below:

- System will calculate and store the average ratings for each song uploaded
- System will update the ‘Weekly Top 10’ list according to the ratings calculated beforehand
- System will manage the secure authentication by verifying login details, generating unique access tokens.
- System will produce playlists based on users recent preferences.
- System will notify the user when a new song is released by the artists followed.

EXPECTED CONTRIBUTIONS

The entire development process, including design, implementation, and project management, will be carried out only by Ahmet Caner Karaca.

REFERENCES

Amazon Web Services. (2025). *Amazon EC2 Documentation*. Retrieved from https://docs.aws.amazon.com/ec2/

Amazon Web Services. (2025). *Amazon S3 Documentation*. Retrieved from https://docs.aws.amazon.com/s3/

Amazon Web Services. (2025). *Amazon RDS Documentation*. Retrieved from https://docs.aws.amazon.com/rds/

Spring.io. (2025). *Spring Boot Reference Documentation*. Retrieved from https://spring.io/projects/spring-boot