MUSIC STREAMING DATABASE MANAGEMENT SYSTEM

By - Ameya Santosh Gidh

Overview

This documentation provides comprehensive guidance on setting up and utilizing a Music Streaming Database Management System. The system offers various functionalities for users, including user registration, playlist creation, song management, and more.

Functionalities Available:

1. User Management:

- Register new users
- Login and Logout for existing users
- View profile information

2. Playlist Management:

- Create playlists
- Add songs to playlists
- Delete songs from playlists

3. Song Information:

- View information about songs

4. Payment Plans:

- Select various payment plans

Prerequisites:

To set up and run the Music Streaming Database Management System, ensure the following prerequisites are met:

1. Python 3.x: Download and install Python from the [official website](https://www.python.org/downloads/).

2. MySQL Community Server 8.0 or above: Install MySQL Community Server from the [official website](https://dev.mysql.com/downloads/mysql/).

3. Operating System: Compatible with Windows 10/8/7, Linux, and MacOS.

4. Minimum System Requirements:

- RAM: 128 MB
- Free Storage: 10 MB

5. Python Libraries: Install the necessary Python libraries using pip:

- `pymysql`: `pip install pymysql`
- `cryptography`: `pip install cryptography`
- `flask`: `pip install flask`
- `flask-pymysql`: `pip install flask-pymysql`

Database Setup:

Import Database Dump:

- Open the provided `StreamingDatabaseDump.sql` file in MySQL Workbench.
- Execute the SQL commands to import the database schema and data.

Environment Setup:

1. Create Virtual Environment:

- Navigate to the project directory in the terminal.
- Create a virtual environment using the following command:
- python -m venv <name_of_virtualenv>

2. Activate Virtual Environment:

- Windows:

- <name_of_virtualenv>\Scripts\activate

- Unix:

- source <name_of_virtualenv>/bin/activate

Running the Application:

1. Configuration:

- Open `db_config.py` in the `FlaskApp` folder.
- Update `DB_SERVER`, `DB_USER`, and `DB_PASS` with your MySQL credentials.

2. Start Application:

- Navigate to the `FlaskApp` folder in the terminal.
- Run the following command to start the Flask application using python app.py

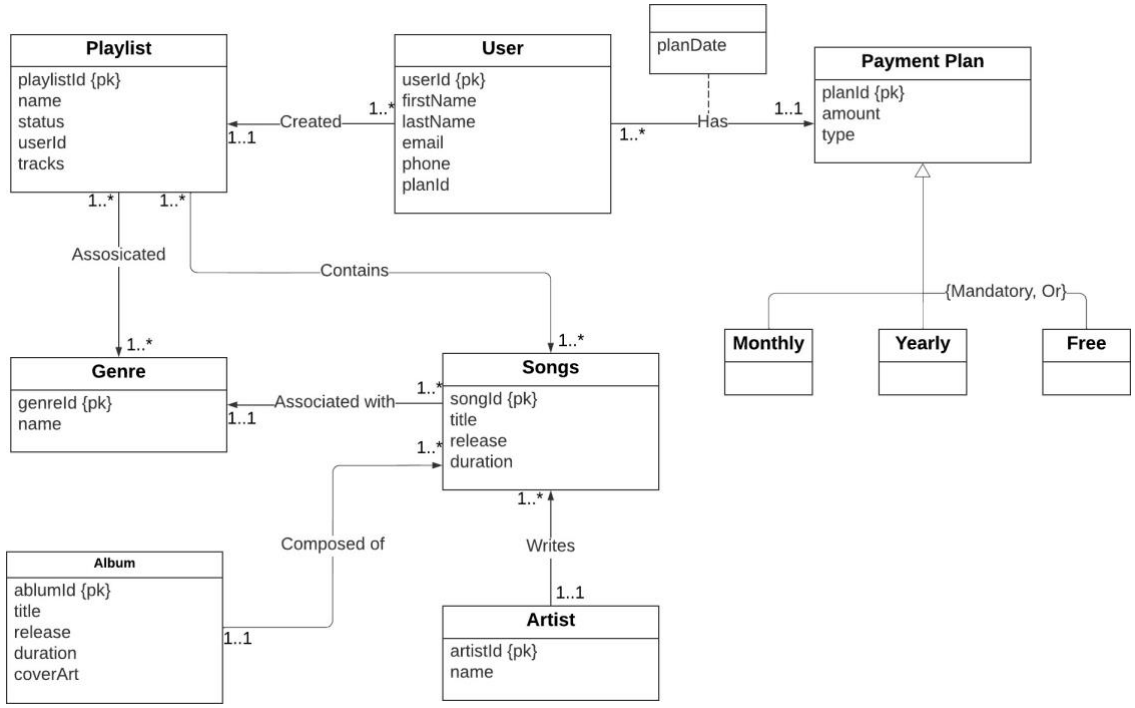
3. Access Application:

- Copy the provided link from the terminal and paste it into a web browser.
- Interact with the application using the provided functionalities.

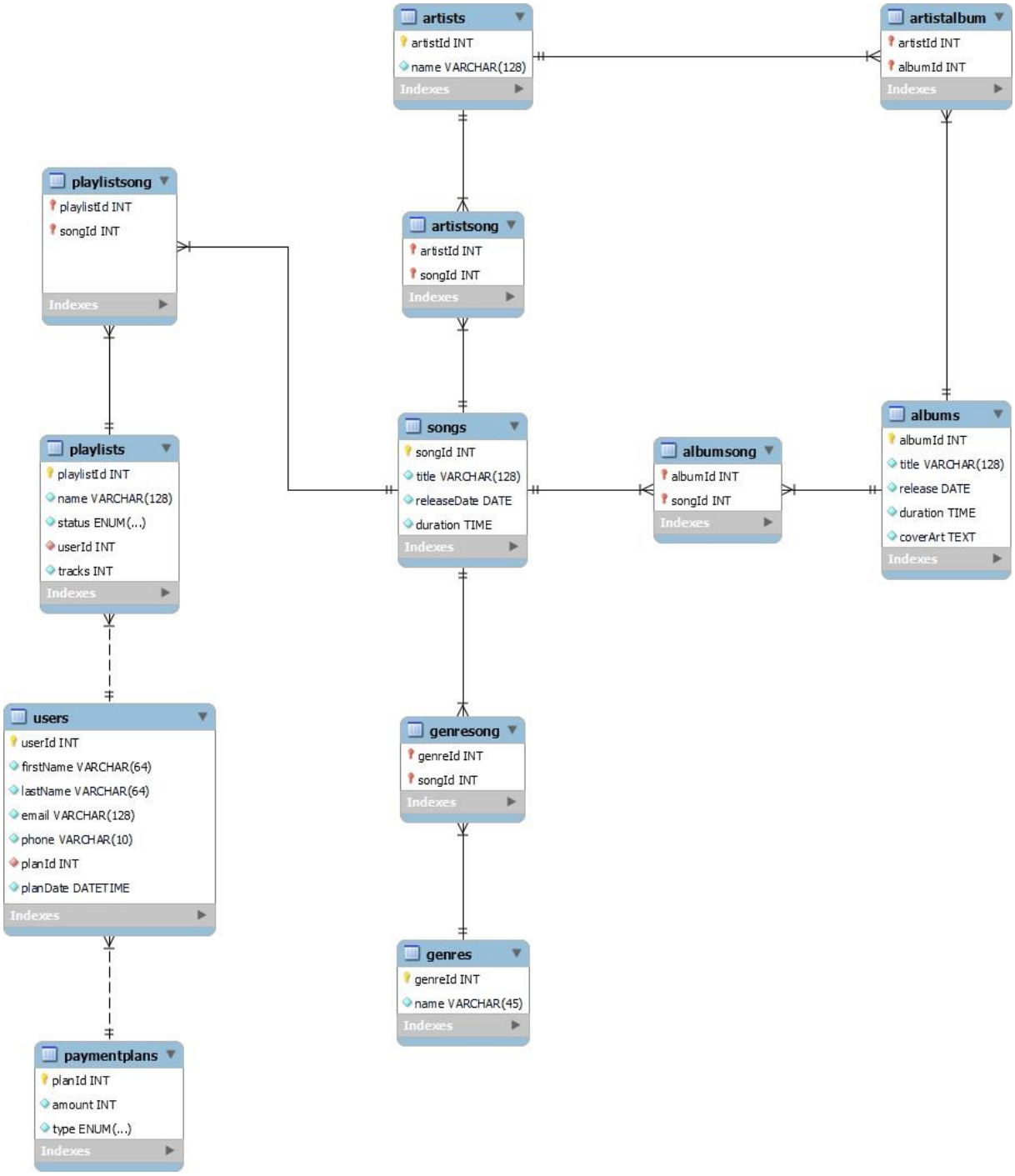
Technical Specifications:

- Software: MySQL Workbench for database management, Flask framework for backend, HTML, CSS, and JavaScript for frontend development.
- Database Connectivity: PyMySQL library for executing SQL commands and managing database operations.
- User Interface: Flask handles GET and POST requests from HTML files, rendering dynamic web pages.
- Database Schema: Designed to accommodate user profiles, playlists, songs, and payment plans.

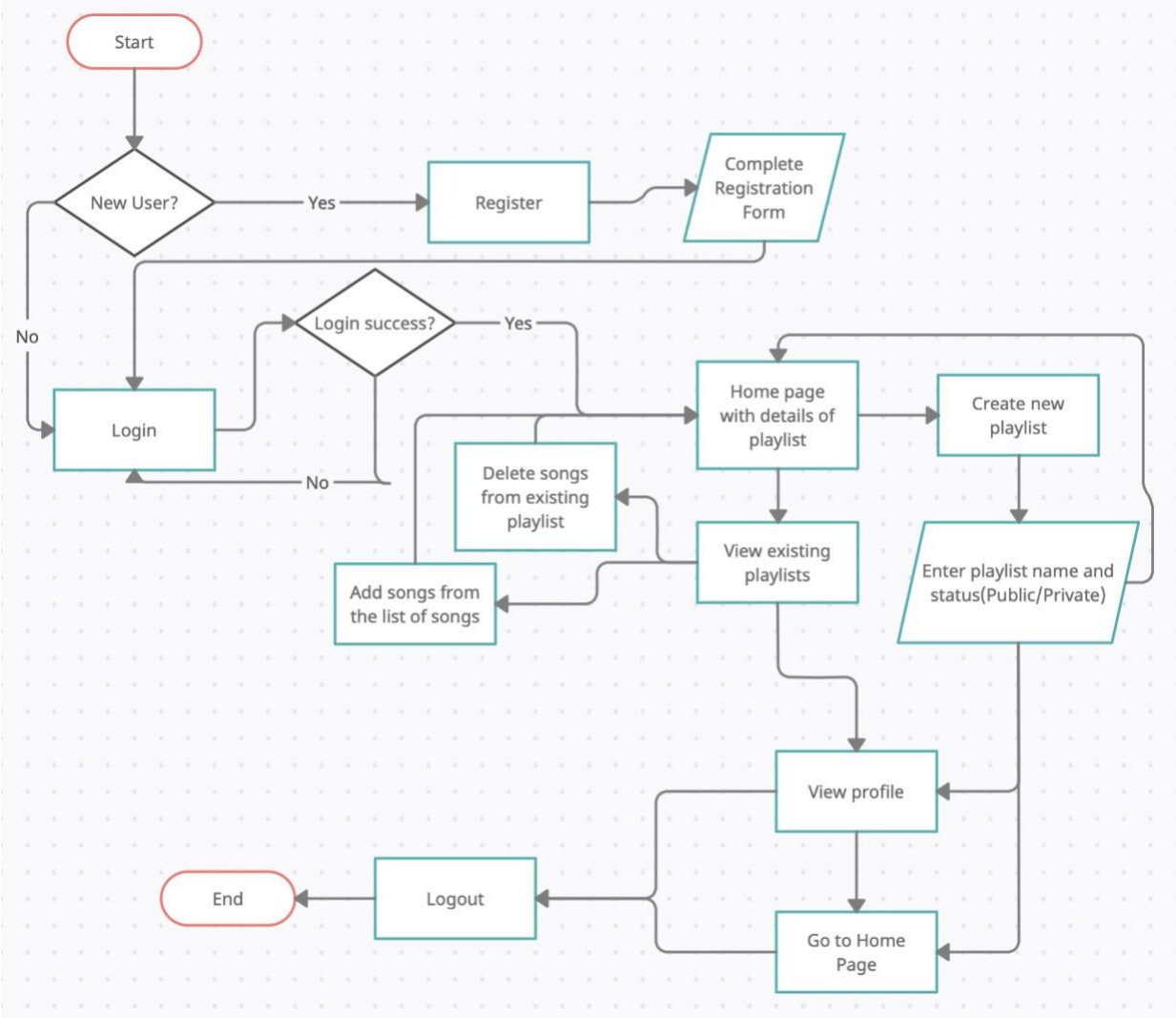
CONCEPTUAL DESIGN - UML



LOGICAL DESIGN (REVERSE ENGINEERED)



USER FLOW OF THE SYSTEM



Lessons Learned:

- Technical Expertise: Gain proficiency in Flask framework, HTML, CSS, and JavaScript integration, and PyMySQL library for database connectivity.
- Time Management: Understand the importance of efficient database schema design
- Data Domain Insights: Explore the complexities of streaming platform architectures

Future Work:

- Enhanced Functionality: Implement song recommendation algorithms based on user preferences and genre information.
- UI Improvements: Add features like multiple genre support for songs, filtering songs by albums, and showcasing featured artists.
- Machine Learning Integration: Utilize machine learning algorithms to enhance song recommendation capabilities.

References/Resources:

- Flask: [Official Documentation](https://flask.palletsprojects.com/en/2.0.x/)
- Font Awesome: [Icon Library](https://fontawesome.com/)
- MySQL: [Official Website](https://www.mysql.com/)
- Python Virtual Environment: [Documentation](https://docs.python.org/3/library/venv.html)

Data Sources:

- Album and Song Information sourced from Wikipedia pages for reference.
By following this documentation, users can successfully set up, configure, and utilize the Music Streaming Database Management System for managing streaming content effectively.

- https://en.wikipedia.org/wiki/Kids_See_Ghosts_(album)
- https://en.wikipedia.org/wiki/Ye_(album)
- https://en.wikipedia.org/wiki/Post_(Bj%C3%B6rk_album)
- https://en.wikipedia.org/wiki/It%27s_Almost_Dry