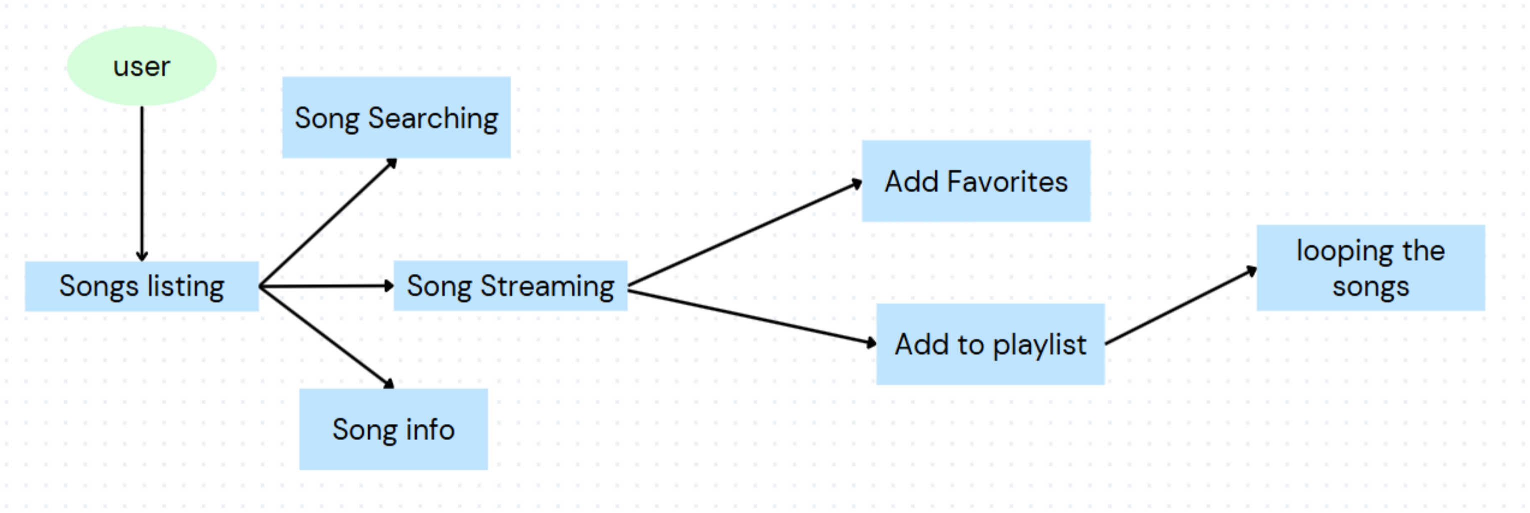
Project Design Phase-II
Data Flow Diagram & User Stories

Date	31 January 2025
Team ID	SWTID1741150324
Project Name	Rhythmic Tunes
Maximum Marks	4 Marks
Team Leader	C Auxilia Sharin
Team member	Prenitha K
Team member	V Kavyashree
Team member	G Divya Bharathi

Data Flow Diagrams:

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.



User Stories

User Story Table – Music Streaming App

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance Criteria	Priority	Release
Music Discovery		USN-1	As a user, I can search for songs, albums, or artists.	I can see a list of relevant search results.	High	Sprint-1
		USN-2	As a user, I can view trending and recommended songs.	I can see recommended music on my dashboard.	Medium	Sprint-2
Playback		USN-3	As a user, I can play, pause, and skip songs.	I can control playback with	High	Sprint-1

			basic buttons.		
	USN-4	As a user, I can view album artwork and song details while playing a song.	I can see album art, song title, and artist name.	Medium	Sprint-2
Playlists & Favorites	USN-5	As a user, I can create my own playlists.	I can save a collection of songs under a custom playlist name.	High	Sprint-2
	USN-6	As a user, I can add or remove songs from my playlists.	I can successfully manage songs within a playlist.	High	Sprint-2
	USN-7	As a user, I can like/favorite songs.	I can save my favorite songs and access them later.	Medium	Sprint-2
Audio Streaming	USN-8	As a user, I can stream high-quality audio.	I can listen to songs without buffering.	High	Sprint-1

Project Design Phase-II
Solution Requirements (Functional & Non-functional)

Date	31 January 2025
Team ID	
Project Name	Rhythmic Tunes
Maximum Marks	4 Marks

Functional Requirements:

Following are the functional requirements of the proposed solution.

Functional Requirements – Music Streaming App

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	Music Search & Discovery	Search for Songs, Albums, and Artists
		View Trending and Recommended Music
FR-2	Playback & Streaming	Play, Pause, and Skip Songs
		Display Album Art and Song Details
FR-3	Playlist & Favorites	Create and Manage Playlists
		Add or Remove Songs from Playlists
		Like / Favorite Songs
FR-4	Audio Streaming	Stream High-Quality Audio

Non-functional Requirements:

Following are the non-functional requirements of the proposed solution.

NFR No.	Non-Functional Requirement	Description
NFR-1	Usability	The app should have an intuitive and user-friendly interface, ensuring smooth navigation and accessibility for users of all demographics.

NFR-2	Security	User authentication and data must be secured using encryption (e.g., HTTPS, OAuth for third-party logins). The app should prevent unauthorized access and follow best security practices.
NFR-3	Reliability	The app should ensure a consistent and uninterrupted music streaming experience, minimizing crashes and downtime.
NFR-4	Performance	Songs should load and stream with minimal buffering. The app should respond to user interactions (search, playback, playlist management) within 2 seconds.
NFR-5	Availability	The system should maintain an uptime of at least 99.9%, ensuring accessibility across different time zones.
NFR-6	Scalability	The app should handle increasing numbers of users and concurrent streams efficiently without performance degradation. The architecture should support future feature expansion.

Project Design Phase-II
Technology Stack (Architecture & Stack)

Date	31 January 2025
Team ID	SWTID1741150324
Project Name	Rhythmic Tunes
Maximum Marks	4 Marks
Team Leader	C Auxilia Sharin
Team member	Prenitha K
Team member	V Kavyashree
Team member	G Divya Bharathi

Technical Architecture:

The Deliverable shall include the architectural diagram as below and the information as per the table1 & table 2

Example: Rhythmic Tunes

Reference: https://open.spotify.com/

Table-1 : Components & Technologies:

S.No	Component	Description	Technology

	User Interface	Web-based interface for music streaming	HTML, CSS, JavaScript / React Js etc.
	Application Logic-1	Music streaming and metadata management	React js, Node js
	Application Logic-2	Playlist and user preference management	React js, Node js
	Database	Stores Songs, playlists, and metadata	JSON Web Server

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
	Open-Source Frameworks	Frontend frameworks	React.js, Node.js, BootStrap, Tailwind CSS
	Scalable Architecture	3-tier architecture with RESTful APIs	Microservices

References:

React.js Documentation

Node js Best Practice

JSON Web Server Referance

https://medium.com/the-internal-startup/how-to-draw-useful-technical-architecture-diagrams-2d20c9fda90d