

Team 39

Jamspac

April 2023

Client: Karan Vadde

Try Pitch



What is Jamspace?

Jamspace is a novel social music listening platform designed to offer music enthusiasts a unique experience. Its key objective is to provide a simple and easily accessible way for people to discover music.



Features

01

Register with Spotify

A new users' information is obtained from Spotify once they give permission for the same. The user then provides their email and password to create their account.

02

Login

Users can log in to Jamspace if they are already registered.

03

Jams

Jams are essentially like scrobbles. Every time the user streams a song on Spotify, the track's metadata is recorded as a jam.

04

Comments

This feature will allow users to comment under a song on the song page.

Features

05

Stars

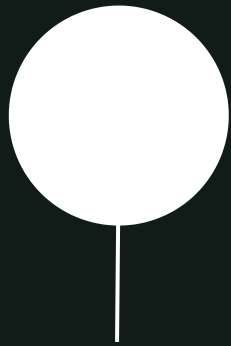
Users can star songs that they like on the song screen. Starred songs get saved to the users' library.

07

User Profile

On the user profile screen, the user's basic details will be displayed along with their library of starred songs and some other basic details and statistics like number of Spotify followers, etc

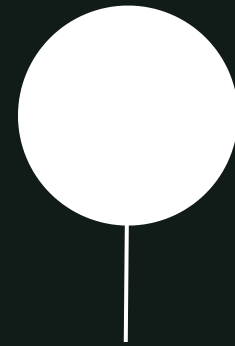
Sprints Completed (pre-R1): $E[x]$



Sprint 1

Documentation:

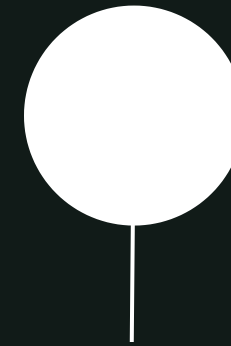
- Project Concept Document v1



Sprint 2

Documentation:

- SRS v1



Sprint 3

Documentation:

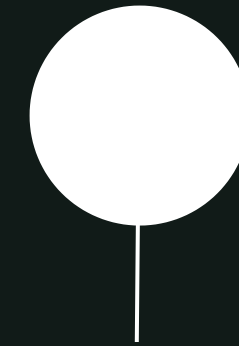
- Project Plan v1
- Project Backlog v1

Development:

- Song Screen (hardcoded)
 - album art
 - star button
 - likes
 - comments

Design:

- Song Screen
- Login Screen



Sprint 4 (Ongoing)

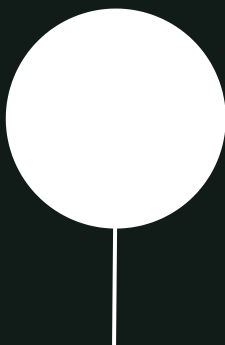
Development:

- Login (Frontend)
- Registration: Spotify Page
- Registration: Get Password

Design:

- Design v1

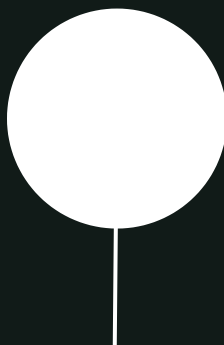
Sprints Completed (pre-R1): Reality



Sprint 1

Documentation:

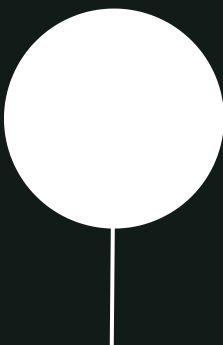
- ~~Project Concept Document v1~~



Sprint 2

Documentation:

- ~~SRS v1~~



Sprint 3

Documentation:

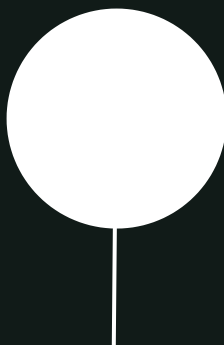
- ~~Project Plan v1~~
- ~~Project Backlog v1~~

Development:

- ~~Song Screen (hardcoded)~~
 - ~~album art~~
 - ~~star button~~
 - ~~likes~~
 - ~~comments~~

Design:

- ~~Song Screen~~
- ~~Login Screen~~



Sprint 4

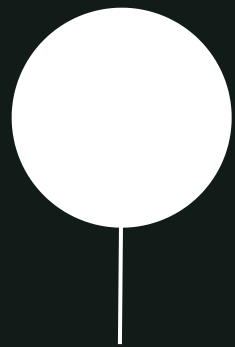
Development:

- ~~Login (Frontend)~~
- ~~Registration: Spotify Page (Frontend)~~
- ~~Registration: Get Password (Frontend)~~

Design:

- ~~Design v1~~

Sprints Completed (post-R1): $E[x]$



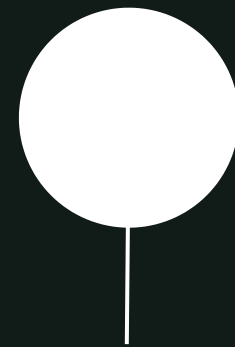
Sprint 5

Documentation

- SRS v2

Development

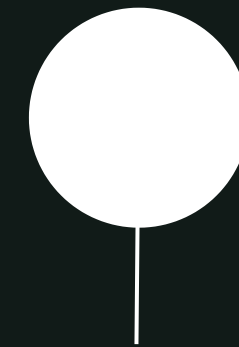
- Login Backend
- Registration - Create Account (Backend)
- Registration - Retrieve Data from Spotify (Backend)
- Song Screen - Backend
- Song Screen - Retrieve Album Art and Likes
- Login Integration
- User Screen (Frontend)



Sprint 6

Development:

- Song Screen Integration
- User Profile Integration
- Registration Page Integration
- UserProfile - Retrieve User Data
- Jams - Update Data
- Jams - Song Tracking
- Song Screen - Comments Backend



Sprint 7

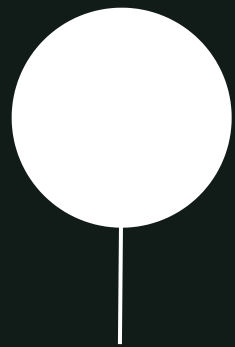
Testing

Debugging

Documentation:

- Backlog v2

Sprints Completed (post-R1): Reality



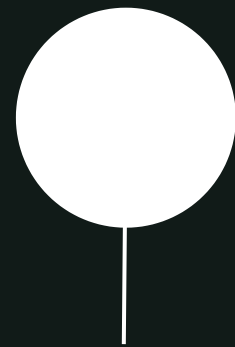
Sprint 5

Documentation:

- ~~SRS-v2~~

Development:

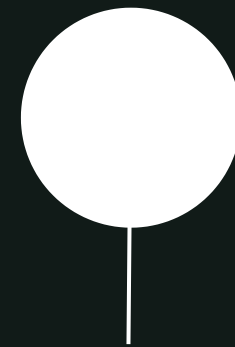
- ~~Login Backend~~
- ~~Registration—Create Account (Backend)~~
- ~~Registration—Retrieve Data from Spotify (Backend)~~
- ~~Song Screen—Backend~~
- ~~Song Screen—Retrieve Album Art and Likes~~
- ~~Login Integration~~
- ~~User Screen (Frontend)~~



Sprint 6

Development:

- ~~Song Screen Integration~~
- ~~User Profile Integration~~
- ~~Registration Page—Integration~~
- ~~User Profile—Retrieve User Data~~
- ~~Jams—Update Data~~
- ~~Jams—Song Tracking~~
- ~~Song Screen—Comments Backend~~



Sprint 7

~~Testing~~

~~Debugging~~

Documentation:

- ~~Backlog-v2~~

Tech Stack



Environment:

Android Studio



Programming Language:

Kotlin



Database:

Firebase

Resources Used

01 Android SDK | Spotify for Developers

02 Kotlin and Android | Android Developers

03 Kotlin Multiplatform for Cross-Platform Mobile
Development

04 Documentation | Spotify for Developers

05 Firebase Documentation

06 App Remote SDK and the Application Lifecycle | Spotify
for Developers

07 Android SDK Quick Start (Kotlin) | Spotify for Developers

08 Local Database in KMM (SQLDelight)

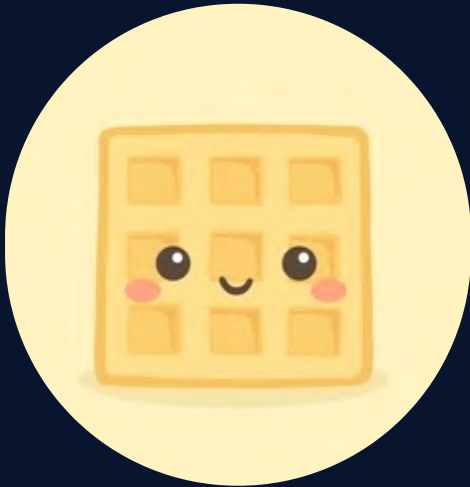
09 Room Database

Challenges faced

- Insufficient resources were available for Kotlin Multiplatform Mobile (KMM) development. So we spent a significant amount of time learning and troubleshooting KMM-related issues.
- The client then suggested that we shift to using only Android, as it is more convenient for the project (9th April, which is just exactly two weeks before R2).
- We had to redo almost our entire code when we switched to Android-only development.
- In that same meeting (9th April), our client also changed our database from SQL Delight to Room DB, as SQL Delight could not be connected to our original frontend.
- After investing time in learning and setting up Room DB over the next week, in the very next meeting, our client realised that it was not useful and so, we had to discard it.
- Firebase authentication services were disrupted, making it hard for us to make progress for a while. Firestore has also been experiencing server issues, which prevented us from testing code for a significant amount of time.

TA Mentor:
Abhijeeth Singam

Team 39



Anuhya Nallapati
2021101076



Maanasa Kovuru
2021101101



Muskan Raina
2021101066



Arghya Roy
2021115008

Thank you!