Jukify Design Document

1. Backend
   1. Team:
      1. Chandan
      2. Enes
      3. Duygu
      4. Amai
      5. Valentin
      6. Patryk
      7. Asyl
   2. Chosen technology: Python
   3. Libraries:
      1. Flask
      2. Spotipy
2. Frontend
   1. Team:
      1. Chandan
      2. Ruslans
      3. Patryk
      4. Amai
      5. Valentin
   2. Chosen technology: native Android (Java)
   3. Libraries:
      1. Volley
      2. Spotify API Library
3. Database
   1. Team:
      1. Enes
      2. Asyl
   2. Chosen technology: MySQL Database
4. User Stories

As a party master, I want to see the queue the users made so that I can modify it if I don’t like it.

As a party master, I want to start a poll so that I can see if party attendees like current playlist/mood.

As a party member, I want to suggest a song so that it might get played.

As a party member, I want to start a skip-poll so that we can vote to skip to the next song.

As an user, I want to be able to see what songs were played at the party so that I can listen to them again.

As a party member, I want a QR-Code to be able to instantly download and configure my application.

As an application user, I want to use the application on my desktop and my android phone.

As a user, I want to be able to see my own history ( what songs were suggested by me and played in the party )

As a party organizer, I want to be to free and allocate new party master at will.

As a party member, I want to be able to rate the song being played.

As a party master, I want to be able to ask party members to rate the song.

As an application user, I want to have multiple language options.

As a party master I want to be able to revoke and grant party members voting permissions at any point.

As a party master I want to be able to timeout party member to prevent him from requesting music for a small period of time.

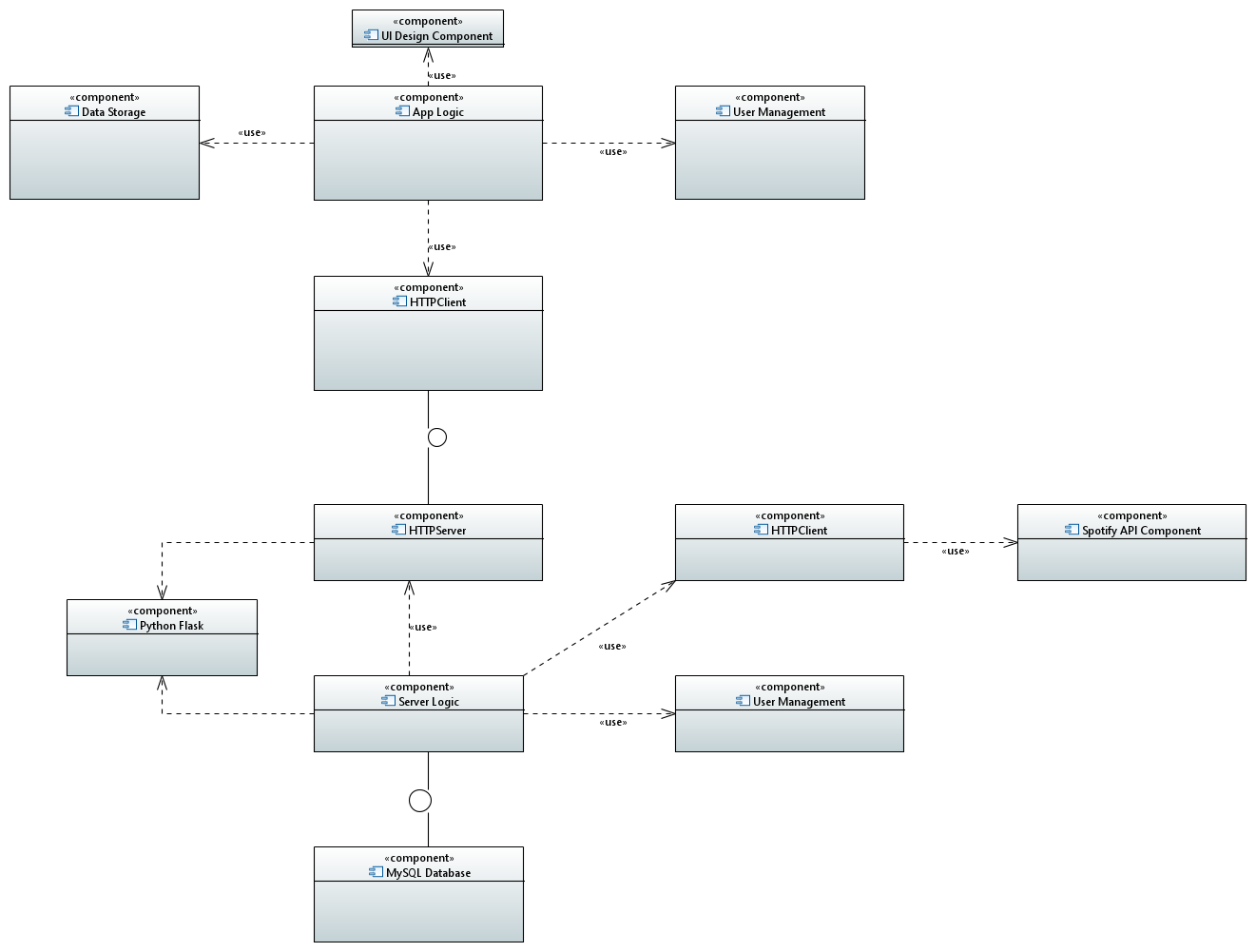
As a party master, I want to be able to share the playlist that was played at the party (after the party has happened).

As a party member, I want to see current played song at real time with time stamp.

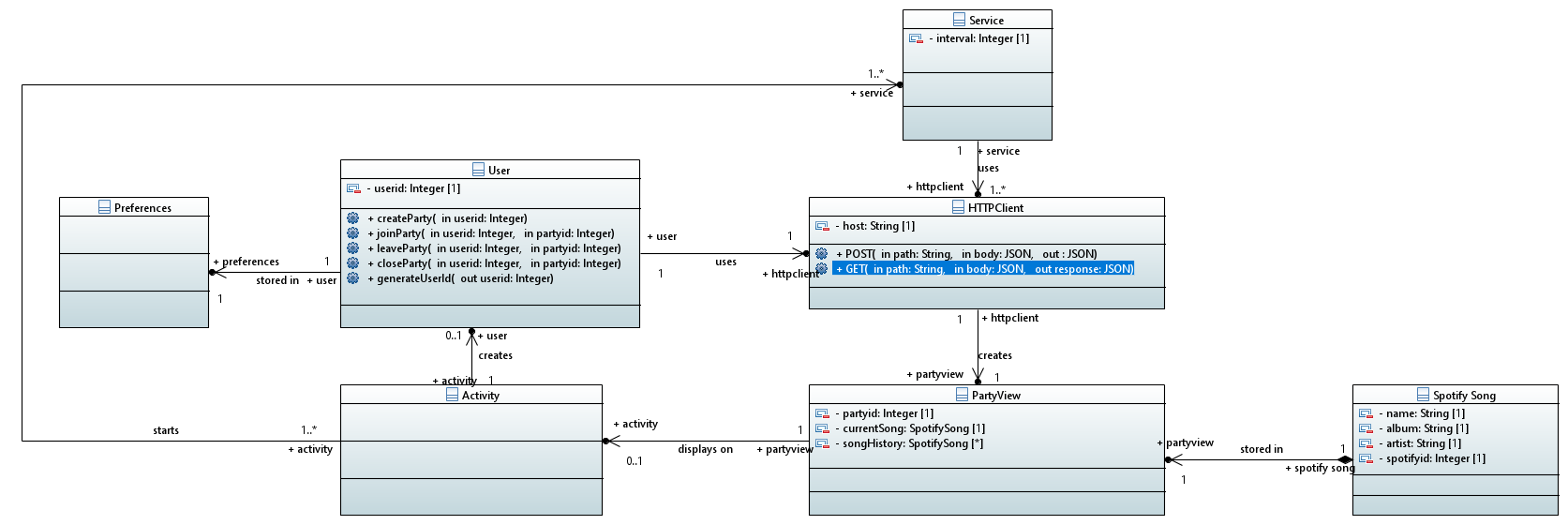
1. Identified Requirements

|  |  |  |
| --- | --- | --- |
| **Requirement** | **Priority** | **Planned Release** |
| Simple UX | High | First release |
| Song search/selection | Essential | First release |
| User management | High | First release |
| Party Database | Essential/High | First release |
| Sort/filter song search | High | First release |
| API Connection | Essential | First release |
| Safety Concerns | Essential | First release |
| Client keeping connection with server | Essential | First release |
| Feedback system for helping party master | Essential | First release |
| Local history | High | First release |
| Database search filters | High | First release |
| Voting system | High | First release |
| Party Master control system | Essential | First release |
| Song suggestion system | Essential | First release |
| QR-Code party join system | High | First release |
| Desktop Application | Low | Later releases |
| Share system | Medium | First/second release |
| Realtime current track display | High | First release |

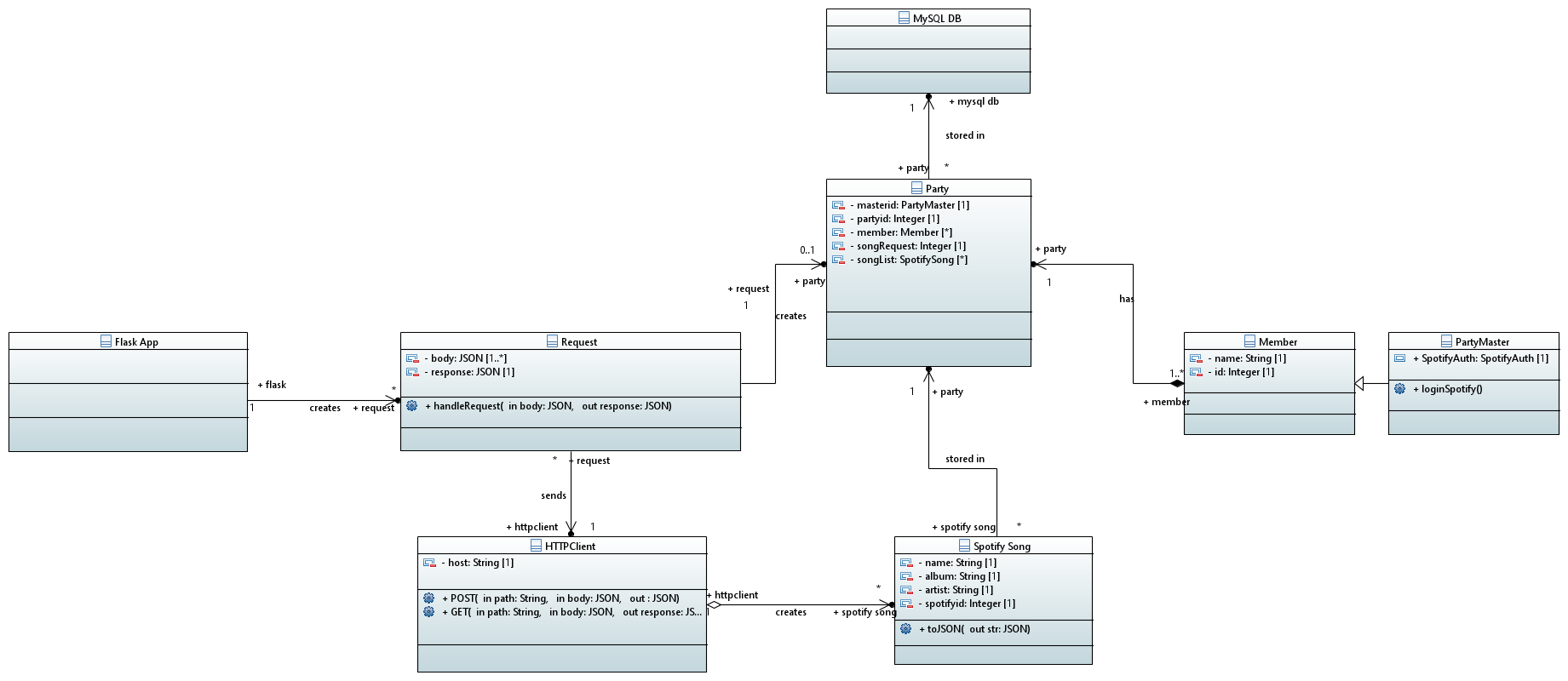
1. Design overview
   1. Component diagram (top: client, middle: server, bottom: persistance)



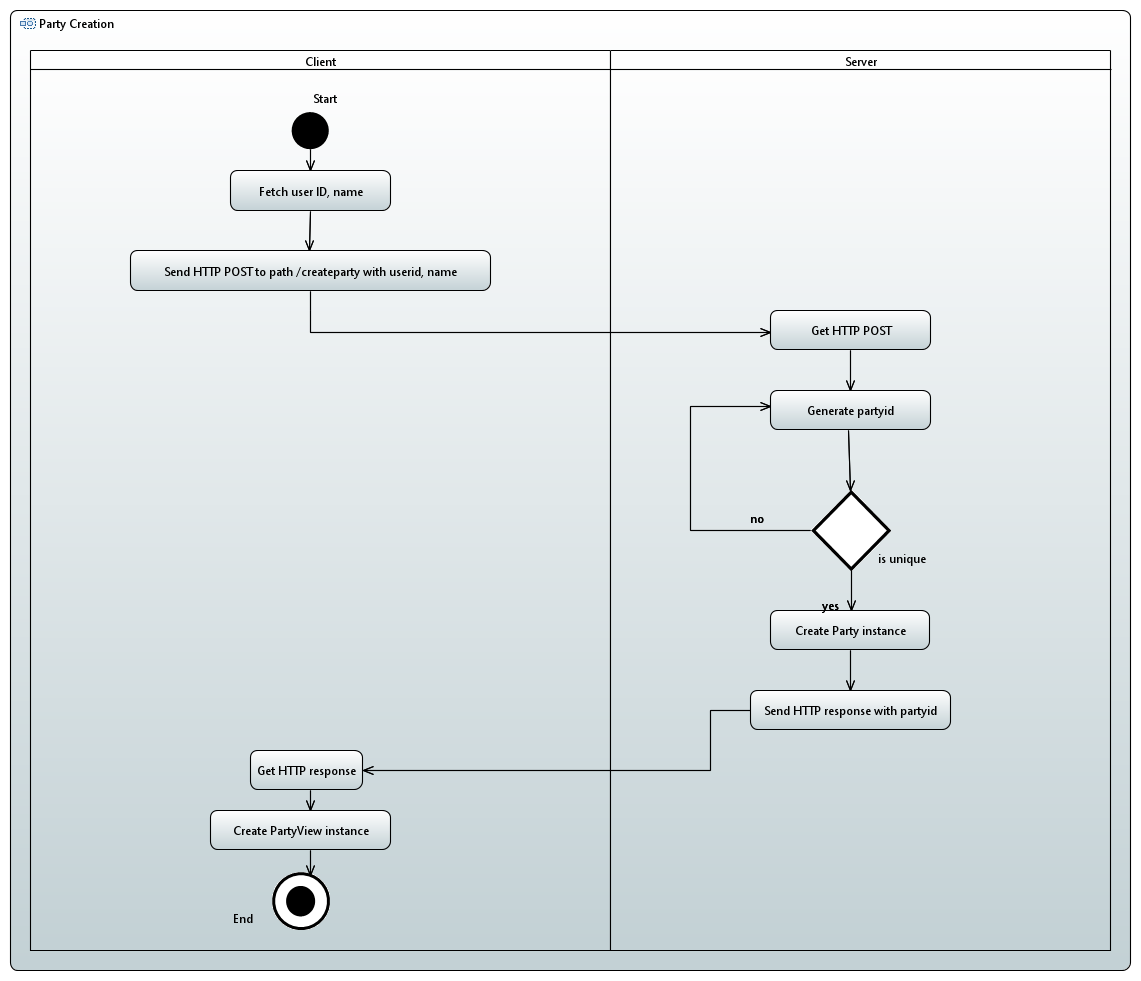
* 1. Frontend class diagram

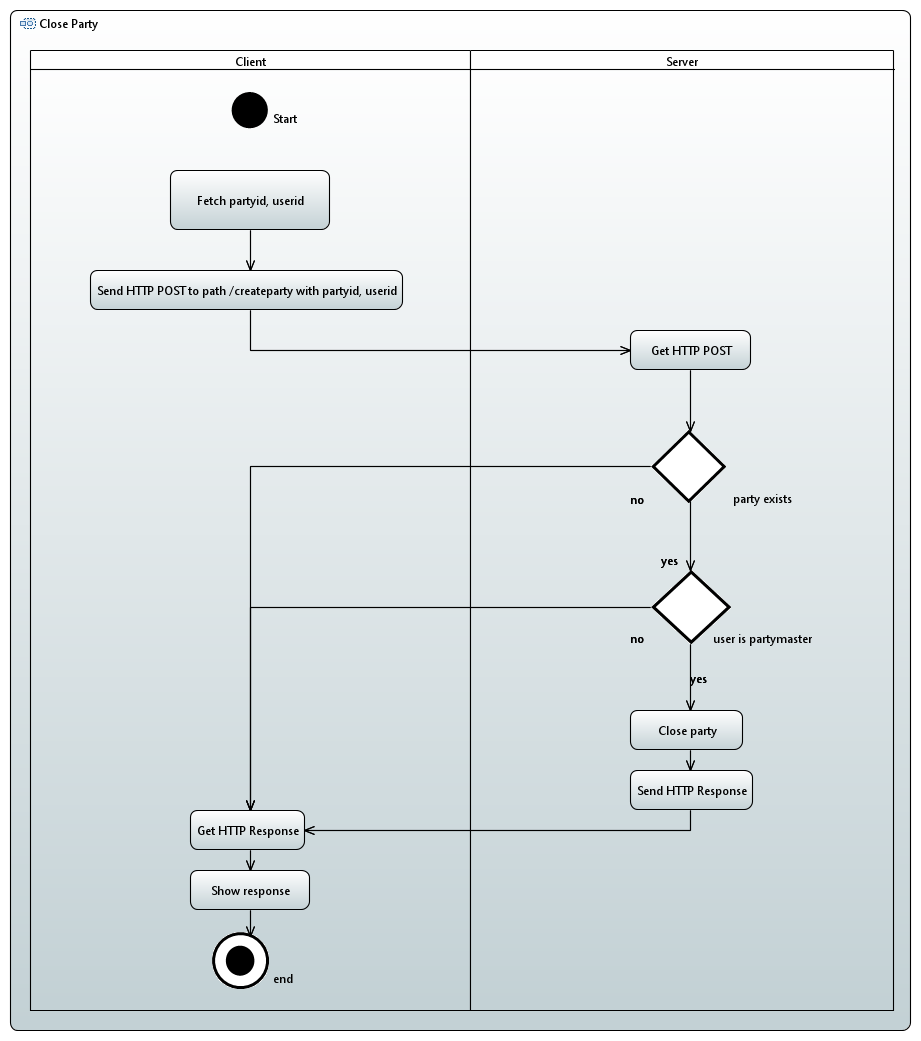


* 1. Backend class diagram

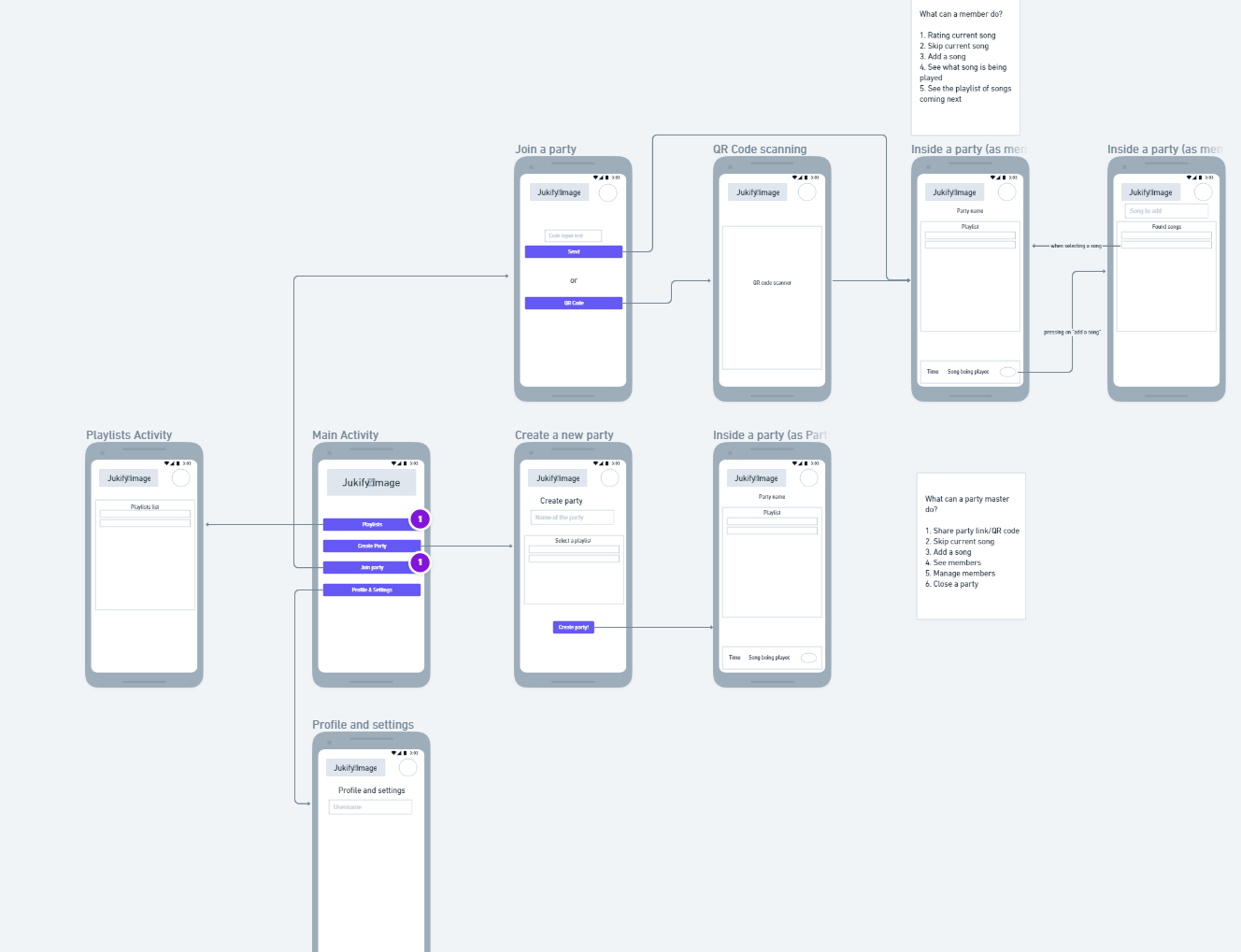


* 1. Activity Diagrams (based on prototype version 1)

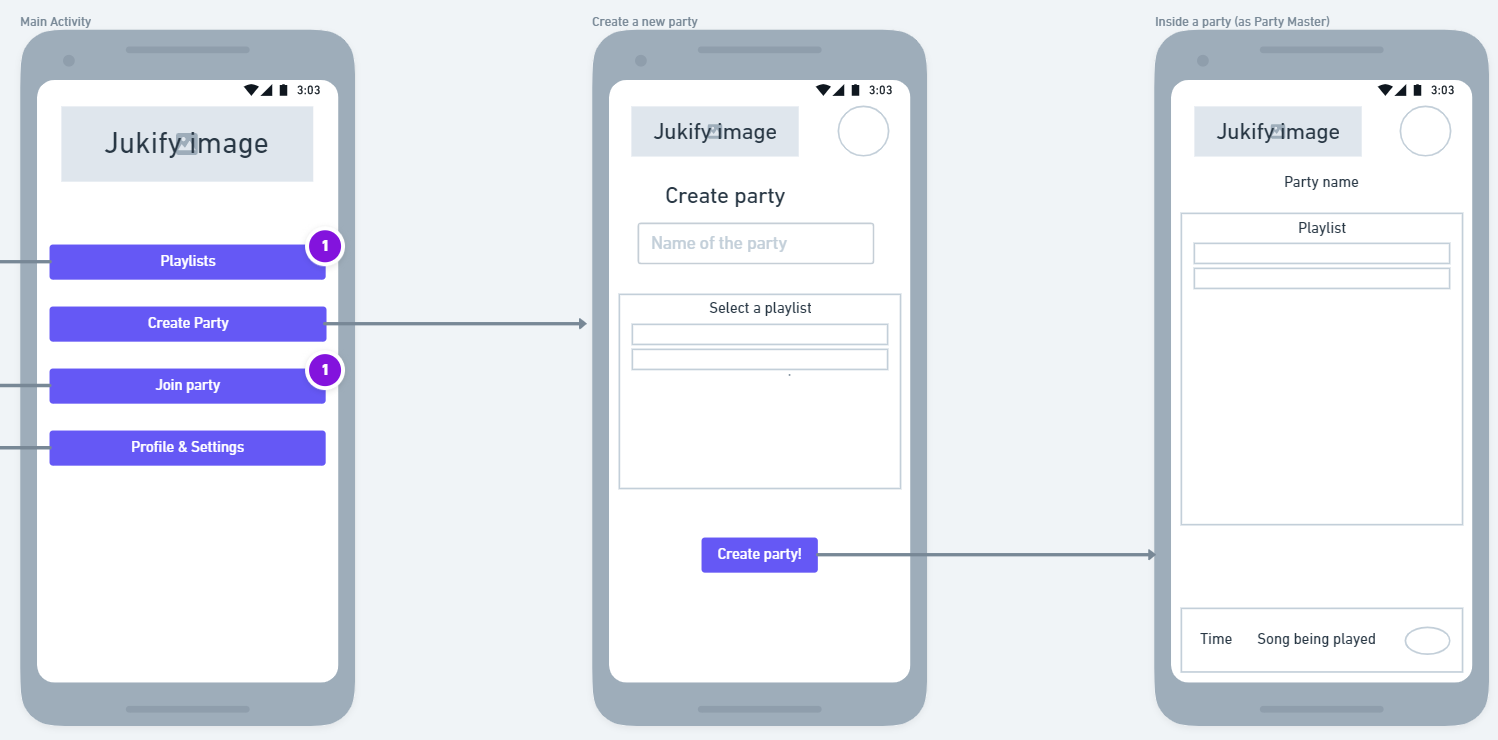




1. UI Design
   1. Overall



* 1. Specific parts





Ein Bild, das Text enthält.

Automatisch generierte Beschreibung