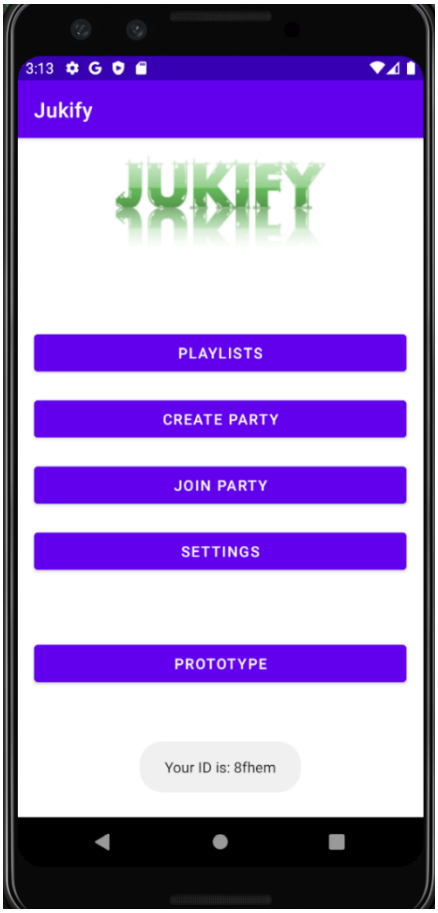
Jukify Progress Statement – Increment 1

1. Progress description

1.1 Frontend prototype app

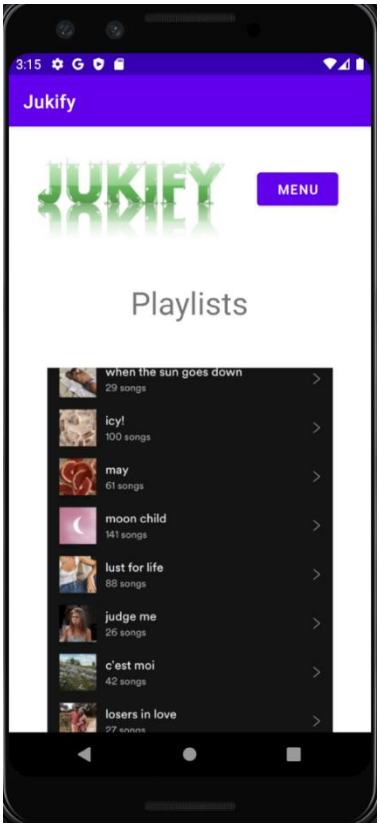
1.1.1 Main Activity

- Buttons starting appropriate activities
- Placeholder logo
- Old prototype version also included
- ID Generation and saving in preferences (to be changed)



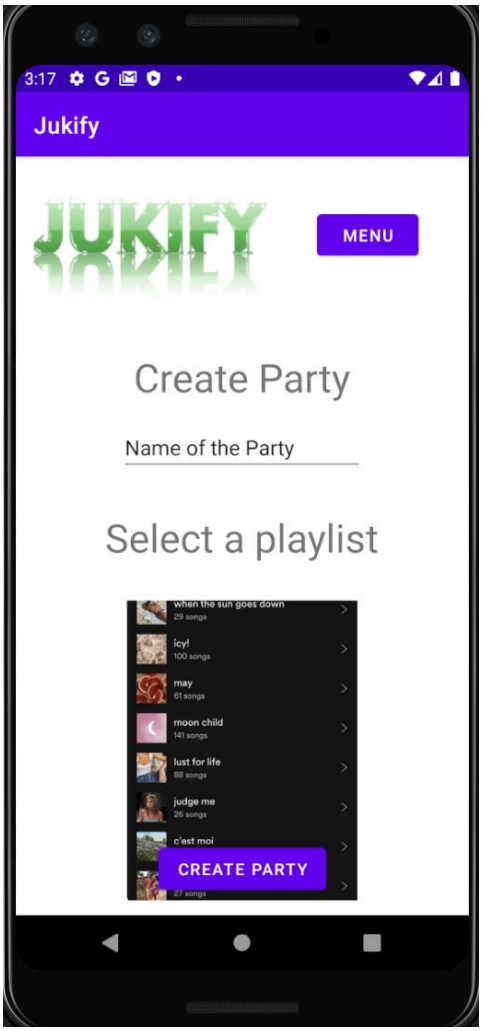
1.1.2 Playlists Activity

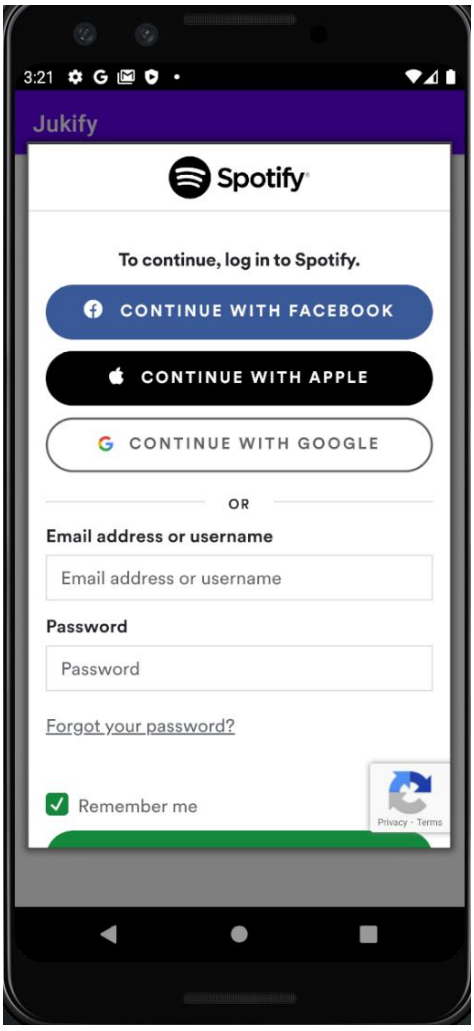
- Playlist list
- Placeholder for now



1.1.3 Create Party Activity

- Allows Party Master to create a party and select entry playlist (placeholder picture)
- Create Party buttons leads to Spotify Authentication through usage of Spotify's Android SDK (uses either a WebView or Spotiy Application to log in)
- Leads to Party Master Activity





1.1.4 Party Master Activity

- Allows Party Master to control the party
- Placeholders for now

1.1.5 Join Party Activity

- Allow users to join by either partyId or QR Code
- No functionality yet, to be added in increment 2

1.1.6 Settings Activity

- Username setting

1.1.7 Next Steps

- Implementing UI functionality
- Implementing better design (suggested Material Design https://material.io/)
- Unit Testing Activities and components

1.2 Backend

1.2.1 Flask Server

Flask server at the beginning had to be run locally. Enes has recently launched the Flask server on his own personal server so it is available under one general IP address (no longer localhost/messing around with Ips when testing connection from various clients)

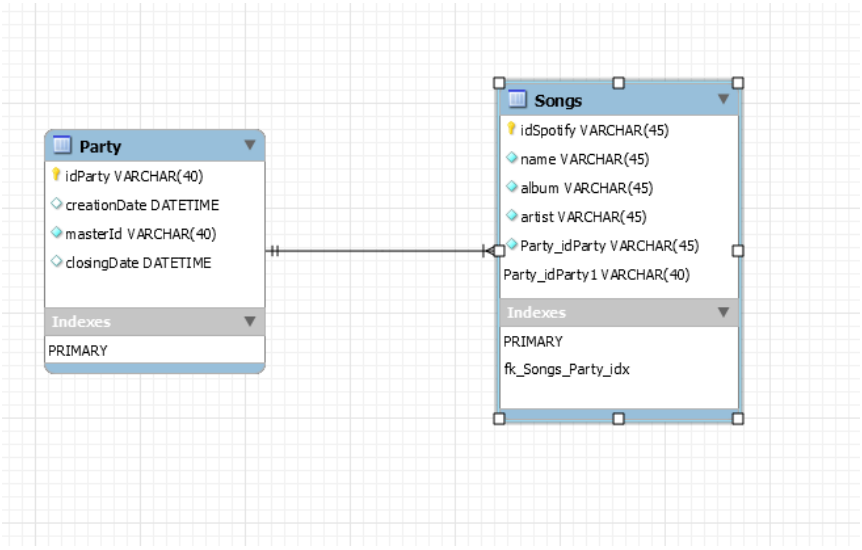
1.2.1.1 Implemented calls

- /createParty
- /closeParty
- /join
- /leave
- /playSong

```
❯ python server.py
 * Serving Flask app 'server' (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: off
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
127.0.0.1 - - [18/Nov/2021 15:28:20] "POST /createParty HTTP/1.1" 200 -
127.0.0.1 - - [18/Nov/2021 15:28:25] "POST /playSong HTTP/1.1" 200 -
```

1.2.2 Database

- MySQL Server Implemented on Enes' server



1.2.3 Next steps

- UnitTesting
- Calls to retrieve data from database