Bulbul

♪Highly Personalized Music Recommendation Platform♪

A. Enes Akdoğan - Ömer Akgül
A. Burak Erdoğan - Mesut Gürlek - Fatih Taş

www.icon8.com

Summary

- About Bulbul
- Bulbul Functionalities
- How It Works
- Data & Core Algorithms
- Demonstration

About Bulbul

Music Recommendation

- Personalized music recommendation
- Music Streaming Platform
- Exploring new music tastes
- Based on moods and genres
- Music in background
- Android 4.4 or above

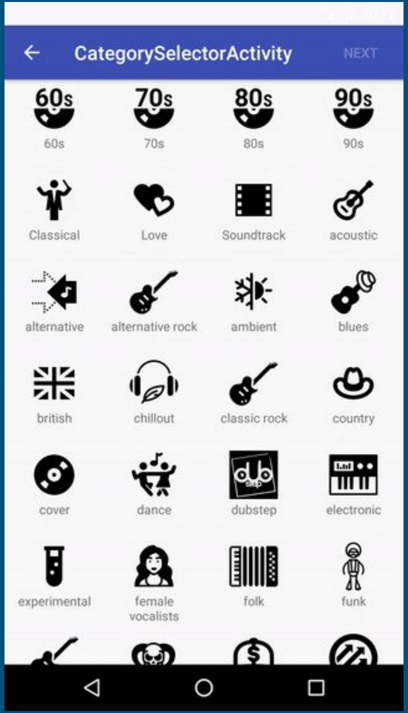
Bulbul Functionalities

Music Recommendation

Instant personalized recommendation

- Choose Category
- Choose Artists
- Rate Musics

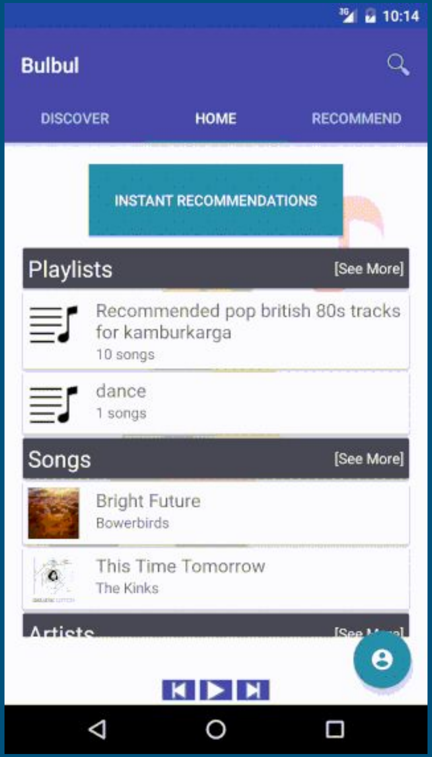
Collaborative + Content Based



Music Recommendation

Cumulative personalized recommendation

- ○ Choose one or more Category
- ○ Takes previous Playlists
- ○ Collaborative + Content Based



Playlists [See More]

Recommended pop british 80s tracks for kamburkarga
10 songs

dance
1 songs

Songs [See More]

Bright Future
Bowerbirds

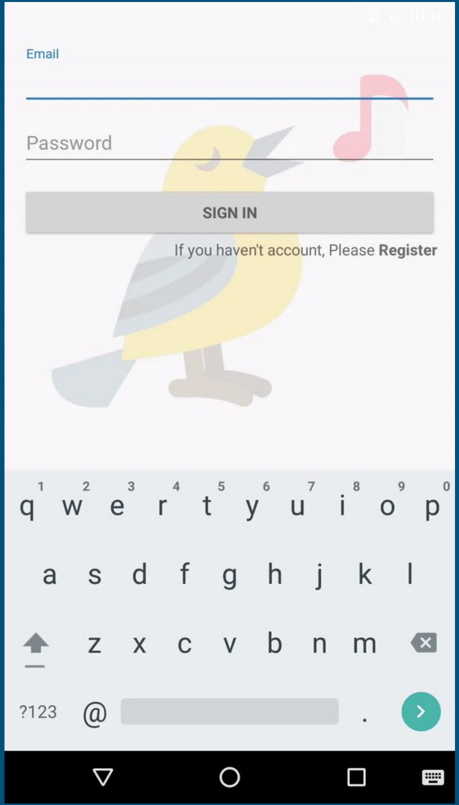
This Time Tomorrow
The Kinks

Artists [See More]

Music Recommendation

Discovery

- Pick music features as you wish
- Create a mix of features
- Enjoy the unique tastes



Email

Password

SIGN IN

If you haven't account, Please **Register**

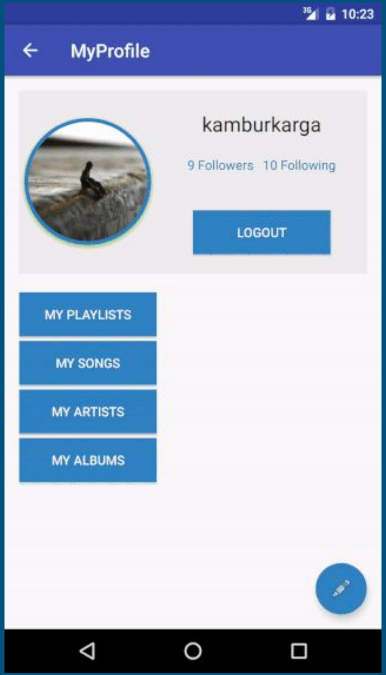
Other Features

Streaming

Rating Musics

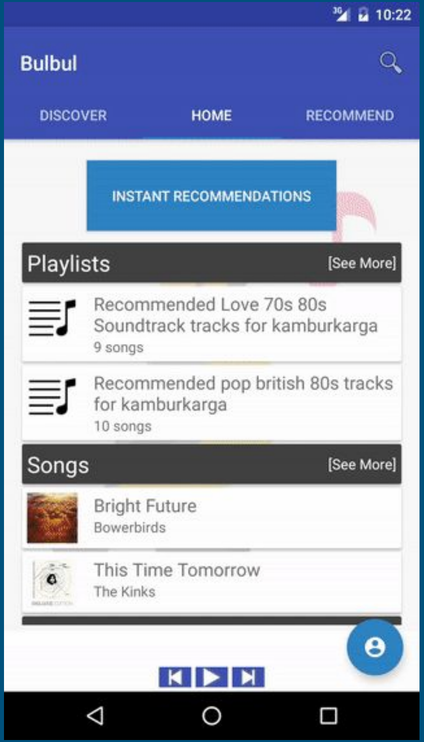
Profile and Playlists

Search Musics



kamburkarga

9 Followers 10 Following



Playlists

Recommended Love 70s 80s Soundtrack tracks for kamburkarga
9 songs

Recommended pop british 80s tracks for kamburkarga
10 songs

Songs

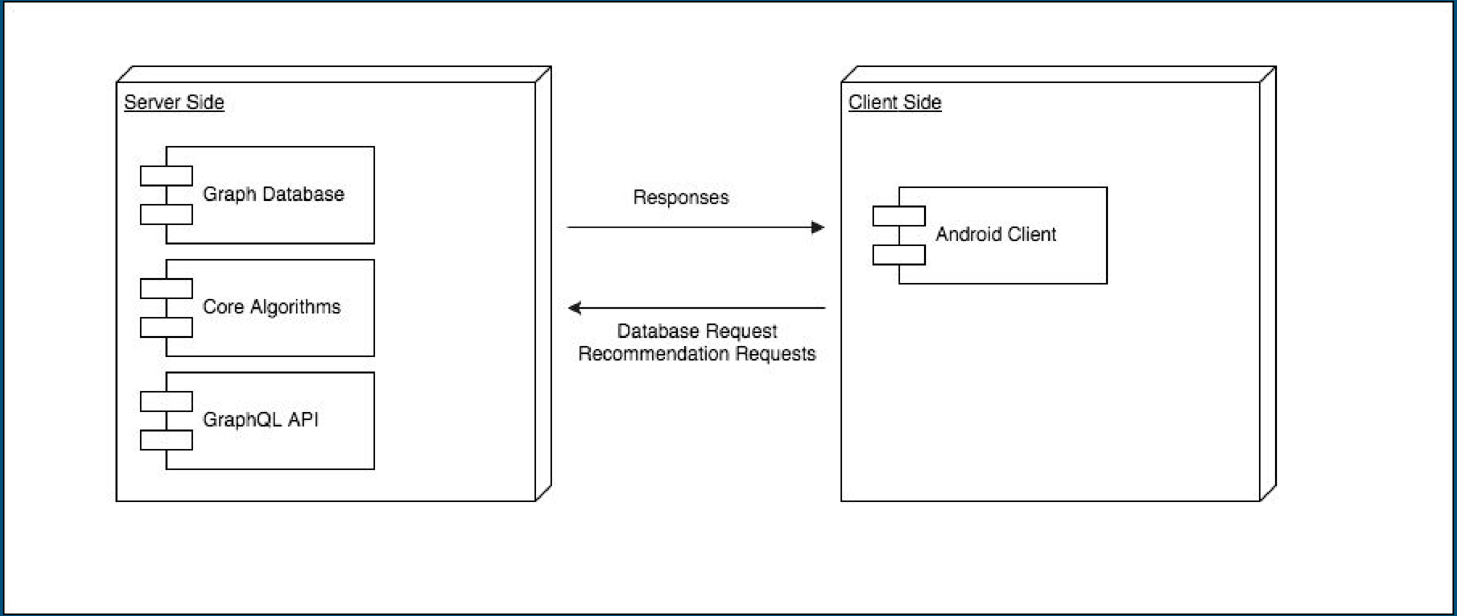
Bright Future
Bowerbirds

This Time Tomorrow
The Kinks

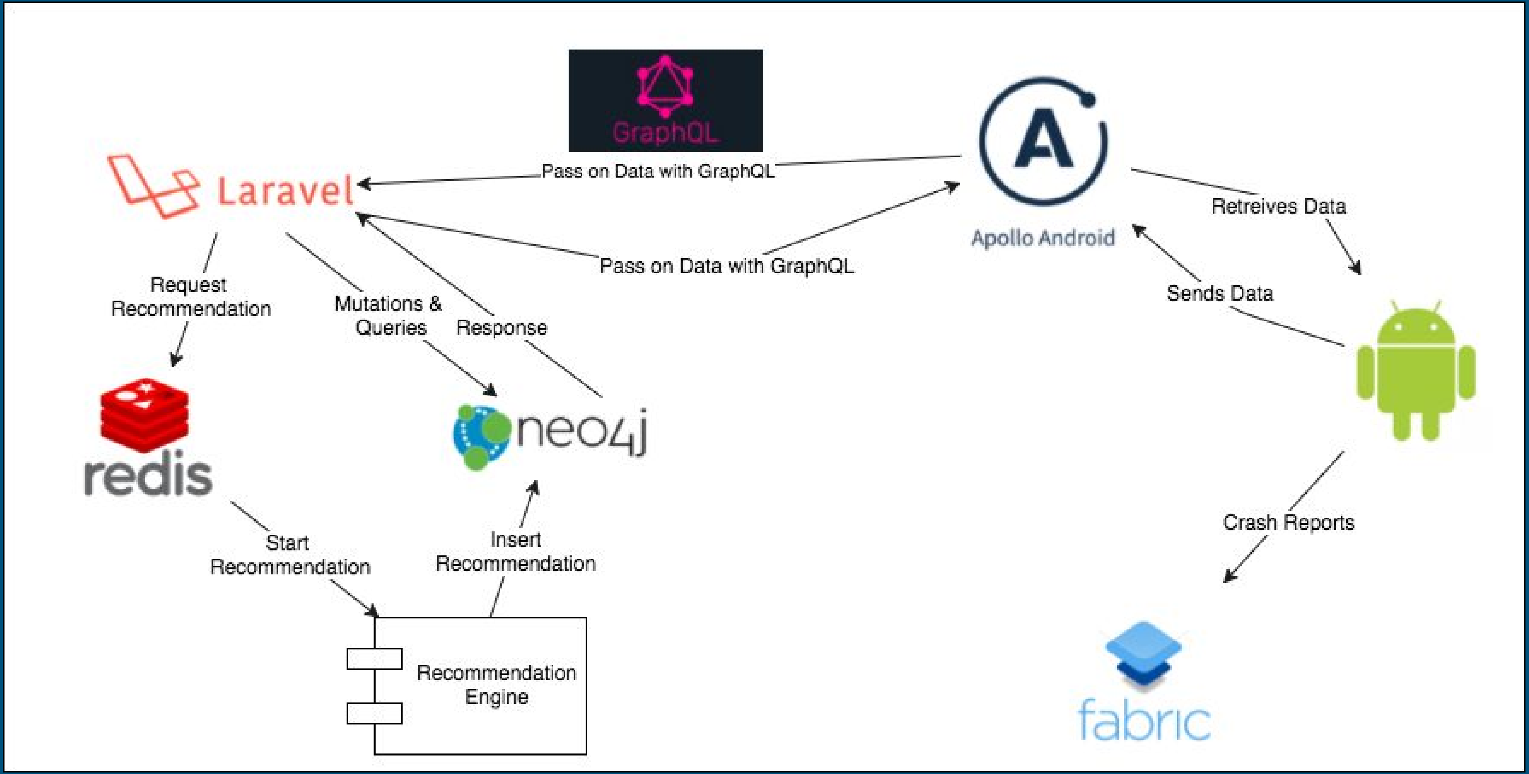
How It Works

Application Design and Infrastructure

Server Side & Client Side



Application Workflow



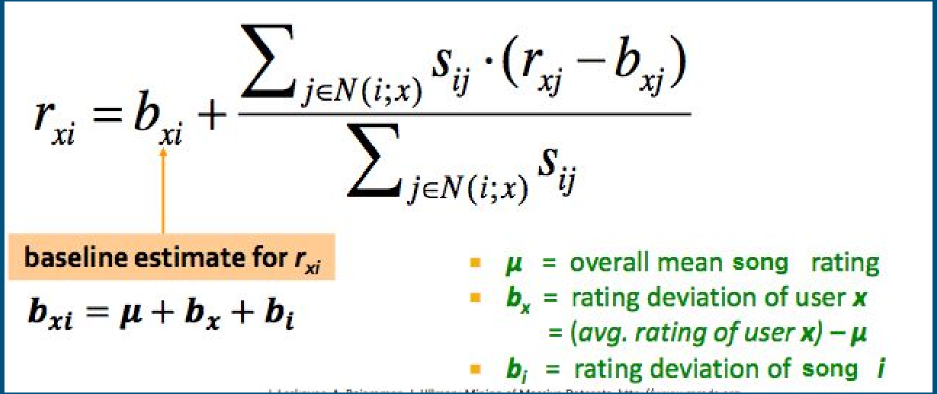
Data & Core Algorithms

Our Dataset and Gathering Process

- ~15.000 Users
- ~900.000 Tracks
- ~240.000 Albums
- ~90.000 Artists
- ~35 million Relations between Nodes
  - Based on the last.fm users
  - 6 Fetcher Machines

Item-to-Item Collaborative Filtering

- Uses 15.000 Last fm users and their activities
- Ratings are generated based on ‘Loved Tracks’ and ‘Top Tracks’
- Utility matrix: User X Track Ratings (15.000 X 900.000)
- Caching similarities between tracks(track feature vectors)
- Cosine Similarity
- Rating Estimation
- Caching Matrices



$$r_{xi} = \underset{\uparrow}{b_{xi}} + \frac{\sum_{j \in N(i;x)} s_{ij} \cdot (r_{xj} - b_{xj})}{\sum_{j \in N(i;x)} s_{ij}}$$

baseline estimate for r_{xi}

$$b_{xi} = \mu + b_x + b_i$$

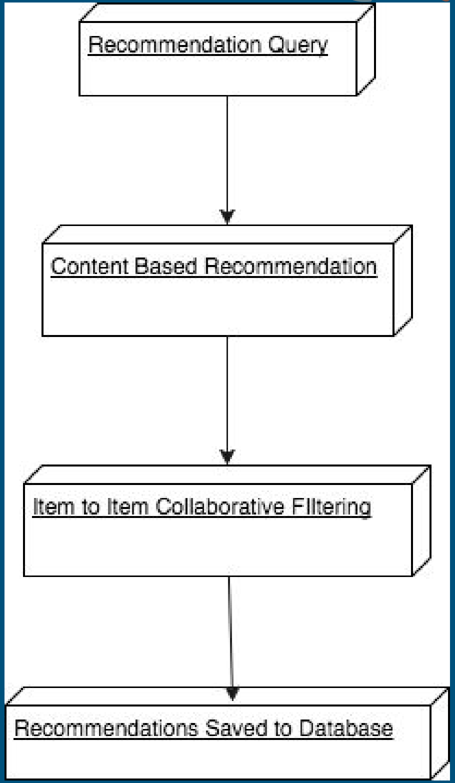
- μ = overall mean song rating
- b_x = rating deviation of user x = (*avg. rating of user* x) – μ
- b_i = rating deviation of song i

Content Based Recommendation

- Based on features obtained from spotify
- Features: Danceability, energy, speechiness, acousticness, instrumentalness, liveness, valence, tempo
- Recommendations based on nearest neighbors
  - Euclidean Distance
  - Weighting with Rating

Ensemble Of Algorithms

- Requested query ⇒ Content Based
- Content Based ⇒ Item-to-Item
- Item-to-Item ⇒ Database as Recommendation Node
- Allows related but personalized recommendations



Recommendation Query

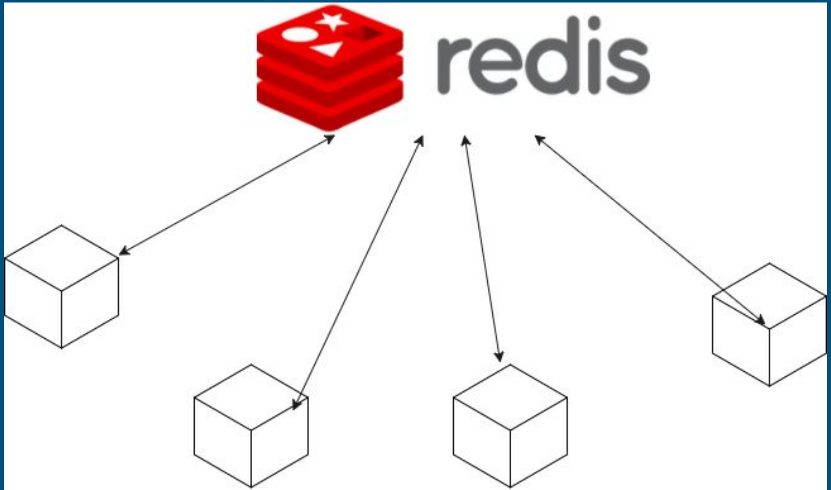
Content Based Recommendation

Item to Item Collaborative FIltering

Recommendations Saved to Database

Redis Task Queues

- All recommendations are kept in Redis Queue
- Multiple worker machines can fetch requests in parallel
- Easily Scalable since it is Distributed



Demonstration