 Cairo University,

Faculty Of Engineering,

Dep. Of Computer Engineering,

2nd year.

**Software final project**

Submitted to: Dr. Ahmed Kaseb

Back-end team 3

Team member ID

Aya Samir 9180351

Bahaa Eldeen Mohamed 9180390

Dina Alaa Ahmed 9180534

Nada Abdelmaboud 9181513

Link to the source code Git repository:

Source : <https://github.com/MohmedMonsef/Marioplex-Backend>

API documentation : <https://github.com/MohmedMonsef/apidoc>

In different repo so that front & android know last updates

Distribution of the final tasks:

In phase three ,we send more than 50% around 80% and we has done all features in Spotify and we asked TA and he told us in phase 50% we only be evaluated on 50% , and the rest will be evaluated in this phase . which we write (new) in feature we consider it in final phase, or we have done after 50% phase. (old) which is in 50% phase.

Aya Features:

Album: - album listeners statistics per day/month/year (new)

Track: - track listeners statistics per day/month/year (new)

Player: - create Queue by array of tracks are not in same album or playlist like (playing track from radio ratio in Spotify)

- add track to queue (old)

- skip next (old)

- skip previous (old)

- shuffle user playback (old)

- repeat user playback (old)

- do some edits in player functions

User: - login (old)

- promote to Premium (confirm done by Nada) (new)

- become free (new)

- update Premium information (new)

- get user player data (new)

Playlist: -create playlist (new)

-get playlist (new)

-delete playlist (new)

-follow/unfollow playlist (new)

Artist: - update artist information (new)

- delete album (new)

Home page: - get popular albums (old)

- get popular artists (old)

- get popular playlists (old)

- get recently playing (albums, playlists, artists) (new)

- get new releases albums (old)

Browse: - get playlists of genres (new)

Search: - add to recently search (new)

-delete from recently search (new)

- get recently search (new)

Other: - initialize half of API documentation before start project (old)

- do environment variable (new)

- collection library/player/browse/Home (new)

- unit test library/player/browse (new)

- update in API documentation

- store volume, current timestamp in playing track and repeat track flag when logout

-Unprotected Authorization for Home Page & Artist routes

Bahaa Eldeen Features:

Track: - get and stream track audio file (new)

- upload track audio file with multiple quality and supports encryption (new)

- upload track binary data in google drive for the free 15GB (new)

- edit track (new)

- get track info, audio features, get multiple tracks (old)

- get tracks related to certain track (new)

- like and unlike track (old)

- delete track (new)

- track like statistics per day/month/year (new)

Image: - upload image file for all entities in database (new)

- upload image binary data in GridFS (new)

- update image file for all entities in database (new)

- get and stream image binary data for all entities (new)

- delete image from all entities (new)

Player: - pause/resume track (old)

- get queue (old)

- add to recently playing (old)

- get recently playing tracks (old)

- prototype for set player instance and update player, but it gets changed heavily (old)

User: - follow/unfollow users (new)

- get followers (new)

- get users whom user follow (new)

Facebook: - login with Facebook (old)

- sign up from Facebook info if user was not in database (old)

Other: - added migration to mongoose (old)

- added rate limit to all routes to avoid bots and automation to server (new)

- default image when there is no image for any entity (new)

- user/track/image unit test (new)

- UML and initial schema (old)

- integrating google API (new)

- using EME API to play encrypted tracks in the web

- collection user/track/image (new)

- update in API documentation

-Unprotected Authorization for Track & User routes

Nada Features:

Notifications: -Artist upload track (new)

-Artist upload album (new)

- user view profile (new)

- user follow playlist (new)

- user follow user (new)

- Notifications History (new)

Artist: - get artist (old)

- get several artists (old)

- get Artist albums (old)

- get Artist Tracks (old)

- get Related Artists (old)

- get Top tracks in a specific country (old)

- create album (old)

- get artist as user (old)

Playlist: -Add track/s to playlist (old)

-remove track/s from playlist (old)

-reorder track/s in playlist (old)

-update playlist details (old)

-get current user playlists (old)

-get any user public playlists (old)

-toggle collaboration status (old)

-toggle public status (old)

-get list of user deleted playlists (old)

- Restore deleted playlists (new)

Library: -check saved track (old)

-check saved album (old)

User: - Forget/Reset password (new)

- claim user to artist (new)

- logout (new)

others: - sendmail with node mailer (old)

- collection artist/playlist (new)

- unit test artist/playlist (new)

- update in API documentation

- Confirmation (signup – premium – update info) (new)

- Playable Tracks (Available markets check) in premium/free

- store Offline Notifications when user is logged out & get them once he logged in again (new)

-Unprotected Authorization for Artist & Playlist routes (new)

Dina Features:

Album: -Likes statistics (per day – month -year)

User: - sign up (old)

- get user profile public (old)

- get user profile private (old)

-remove user account (new)

-update user account information (new)

-user follow artist/unfollow artist (new)

-check if user follow artist (new)

-get artists followed by a certain user (new)

Search: -search for album, artist, track, playlist, profile (old)

Artist: - follows statistics Day/Month/Year (new)

- get album by id (old)

-get albums (old)

-get albums tracks (old)

-follow/unfollow album (old)

-get saved albums (old)

Category: -get categories (new)

-get category by id (new)

-get category playlists (new)

Others: - data base collection (old)

- update in API documentation

-Unprotected Authorization for Album & Search routes (new)

- collection Album & Search (new)

- unit test Album & Search (new)

Challenges the team faced:

- work is purely remotely.

- all the labs were cancelled so a lot of guessing.

- some major concepts like migration was not fully understood.