Team members: Piotr Białek, Antoni Długosz

The project revolved around creating a service featuring artists and music bands, along with their albums, songs, and lyrics. Additionally, we aimed to offer users more than just the ability to read song lyrics, including:

- Adding annotations to lyrics, which users can then rate (+1, -1) and comment on.
- Posting reviews for songs, albums, aliases, bands, and artists, with a special flag for critic users.
- Fun facts about songs, albums, aliases, bands, and artists.
- Access to descriptions of songs, albums, aliases, bands, and artists.
- Access to tags related to a specific artist, band, album, song, or alias.
- Access to the members of a specific band.
- Access to the aliases of a specific artist.
- Easy and intuitive navigation through the graph of relationships between artists, bands, albums, songs, and aliases.

The main issue was the concern about various undesirable situations that could occur in the graph of relationships between artists, bands, aliases, albums, and songs (e.g., not wanting `Daria Zawiałow` to be part of Project `A`, with both `A` and `Daria` listed as performers of a particular song, etc.). We solved this problem by introducing five layers in the graph, namely: `artist, alias, project, album, song`

It turns out that the only remaining prohibition is: edges within the same layer, edges towards the previous layer, and the existence of two paths from a specific vertex to any other vertex that differ by at least one edge.

We believe that the database schema is quite intuitive and does not require additional comments.

We have left the creation of the most important things to the admins, and the functions for this purpose are:

- `create_entity (artist/alias/project/album/song, name, displayed count, description, link to image)`
- `create_user(username, plaintext password, is admin flag, is critic flag)`
- `grant_permission(user id, entity id)`

After granting permissions to a specific user, they can access `ip/login.php`, log in, and start adding albums to the artists they have permissions for, and songs to the albums they have permissions for (important: we assumed that if a user has permissions for a specific artist, they also have permissions for everything "underneath" that artist, such as all their albums, aliases, projects, or songs).

Other possibilities include:

- Linking an artist to a project.
- Adding annotations to songs.
- Adding tags to a given entity.
- Adding interesting facts to a given entity.
- Adding reviews to a given entity (note: anyone can do this, there are no requirements regarding permissions)
- Editing the description/image of a given entity.
- Editing one's own annotations.

We believe that describing all the application's possibilities at this point is quite exhausting and may not necessarily achieve the intended purpose, so we encourage you to watch a short (5-minute) demonstration of the program's capabilities. https://www.youtube.com/watch?v=jdBnKvHYRS8. It is worth mentioning that the delay in adding comments (visible at 1:22) was unintentional and has been resolved. The rest of the assumptions regarding the database and application remain unchanged from the project presentation in the classes.