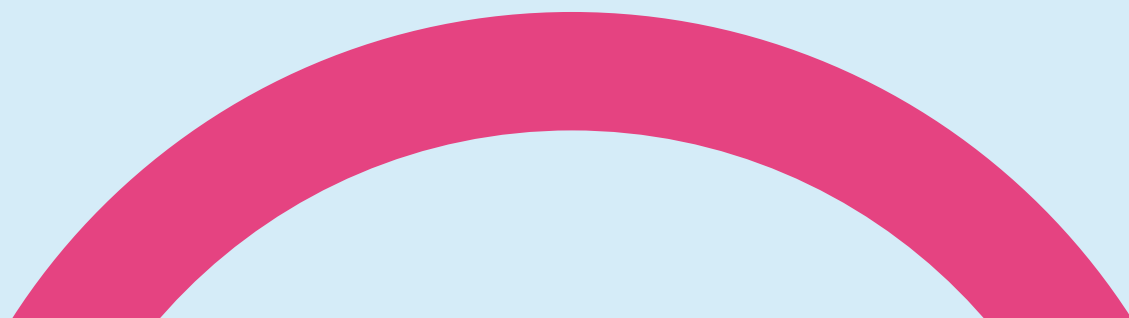
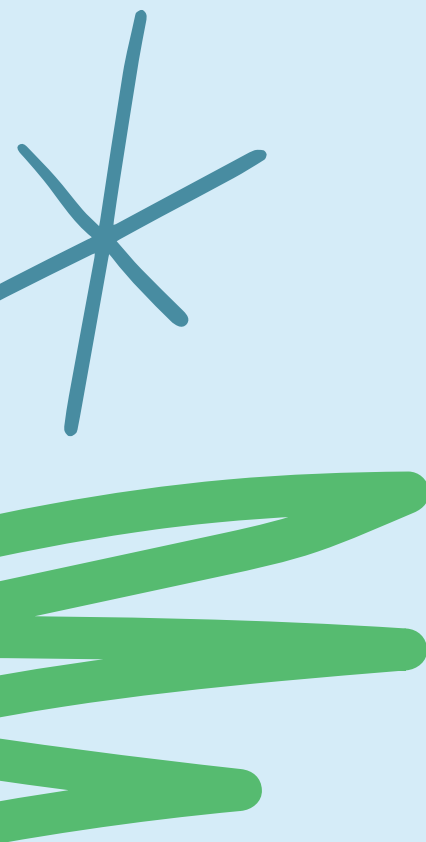


Anime Api on Go

Prepared by: Maratuly Temirbolat



Project description

The project name is anime api. The aim of the project is to investigate the backend part for user - anime handling. Mainly it is just like a common anime page which user can enjoy.

Services:

1. Anime Service (The service is aimed for user interaction with anime)
2. User Service (The service is aimed for user interactions between each other such as adding to friends or blocking)
3. Comment Service (The service is used for leaving comments in any anime or replying to them)

Front part:

The appearance of the site consists of about 5-6 pages. One of the example pages is represented on slide 5. To see all front part please visit "anime" folder on the github and see html files.

Server part:

Implemented functions:

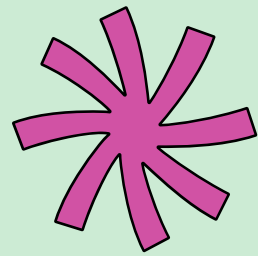
1. BlockUser (Block friend by ID and returning response error)
2. AddUser (Add user to friends by ID and returning response error)
3. ViewUserPage (View user page by ID and returning page's data)
4. HandleAnime (Interaction between current user and anime by action ID, where action means the option that user can do to anime)
5. RemoveAnime (Removing anime from the user's list)
6. ViewAnime (View anime page by anime ID and getting anime data)
7. CommentAnime and ReplyUserCommentAnime (Leave comments under each anime or reply to existed comment)

Client part:

Implemented clients:

1. **AnimeClient** (client that can handle all Anime Service functions: HandleAnime, RemoveAnime and ViewAnime)
2. **CommentClient** (client that can handle all Comment Service functions: CommentAnime and ReplyUserCommentAnime)
3. **UserClient** (client that can handle all User Service function: BlockUser, AddUser, ViewUserPage)

What can you do?



1. Users can access all animes that exist in database (retrieve data about each anime)

2. Users can Like animes.

3. Users can Leave their rating to anime.

4. Users can add anime to their lists (Currently watching, will watch, through away, etc.).

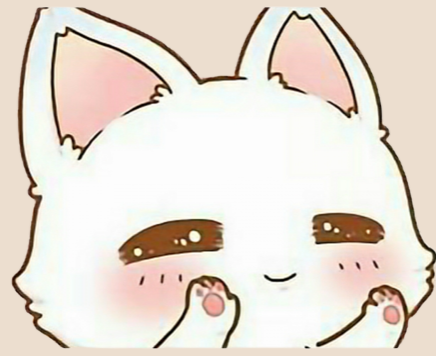
5. Users can add each other to friends as followers

6. Users can retrieve info about other's user's page data.

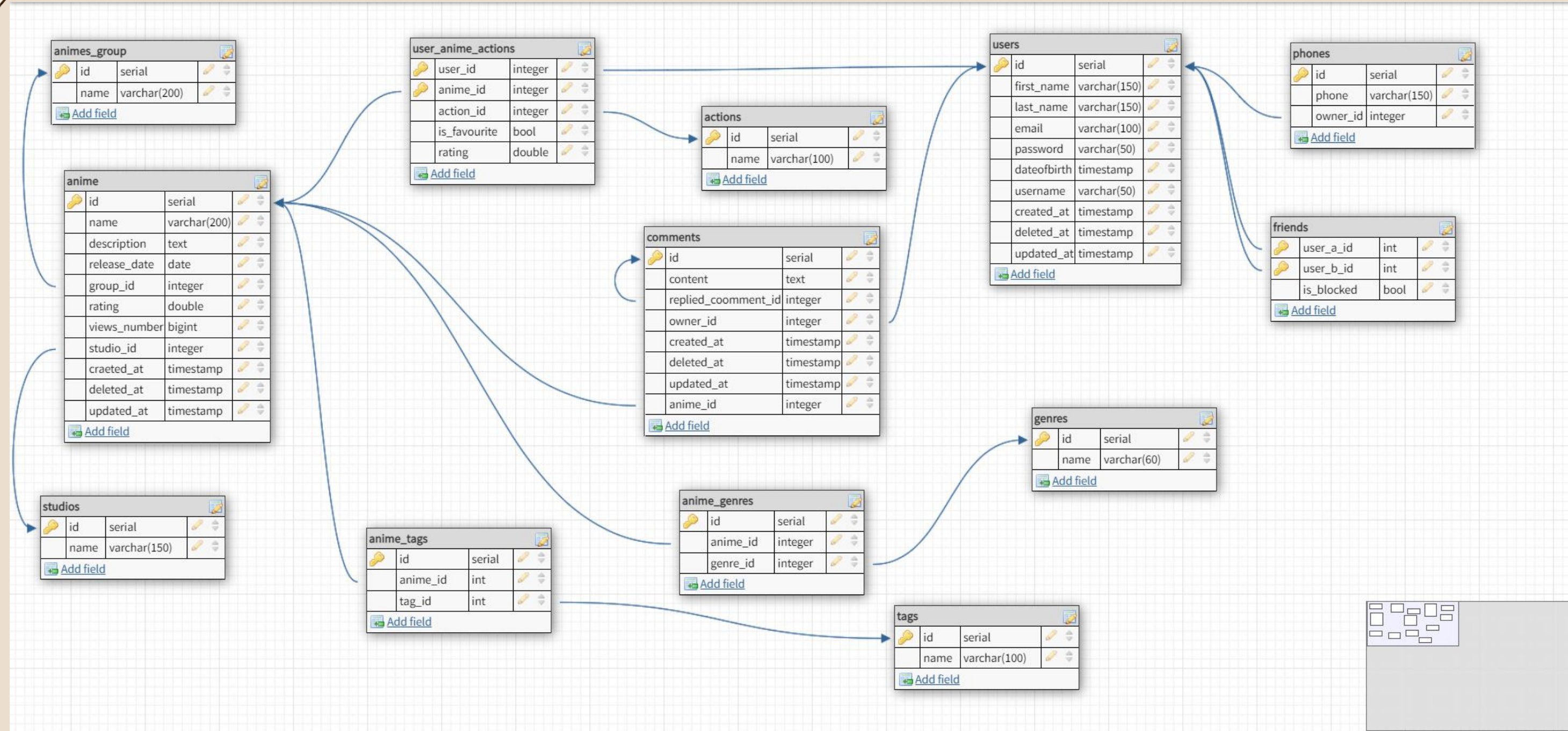
7. Users can block each other but only if they are already friends.

8. Users can leave their comments to anime or reply to already existed ones.





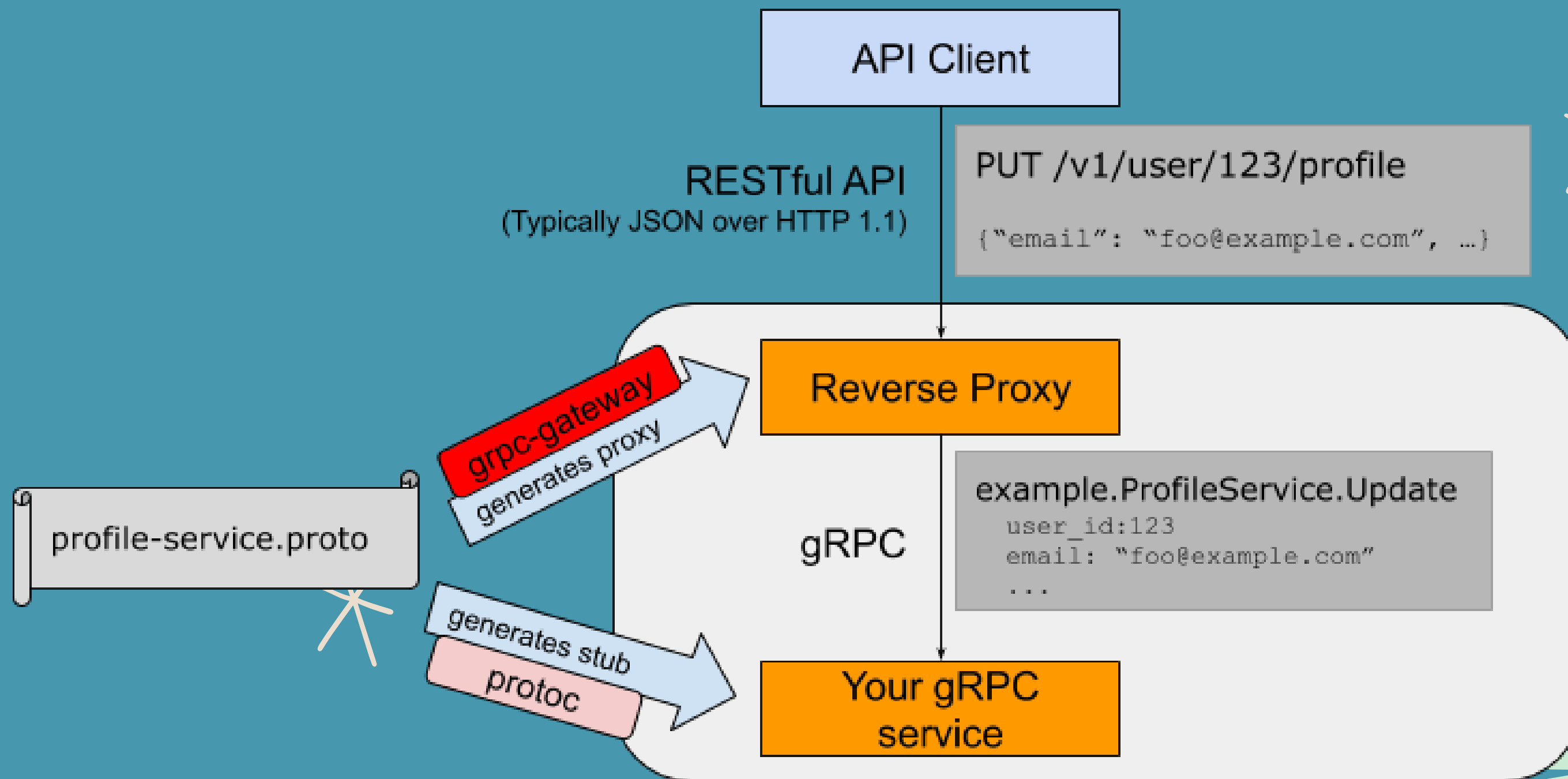
Project Database



Number of tables: 14 (1 extra table for types)

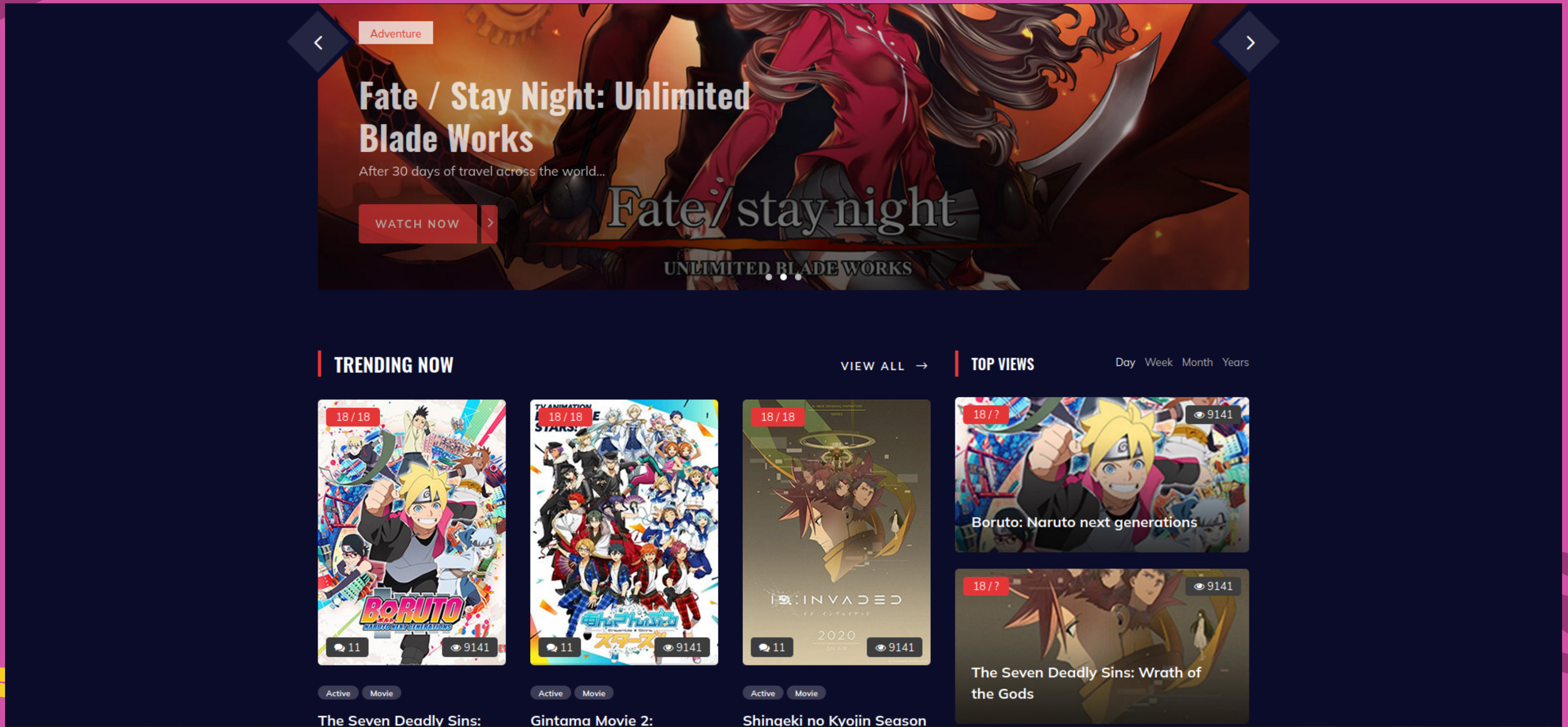


* REST IMPLEMENTATION



GRPC GATEWAY

Front part (see github for more)



Thank you for
your attention!

