Oblivion Sound

Computer Engineering Course
DataBases Class

Project Description

Our goal is to make a complete and functional database for an hypothetical application named *Oblivion Sound*, for the time we have designed a visual entity-relationship diagram and its attributes. We will be using SQL as the programming language for the creation of the actual database, where we will manage the data of a music streaming service implementing the needed functionalities for it to be usable and secure.

Our Team

André

E-mail:
andrelouro@student.dei.uc.pt
Student Number:
2021232388

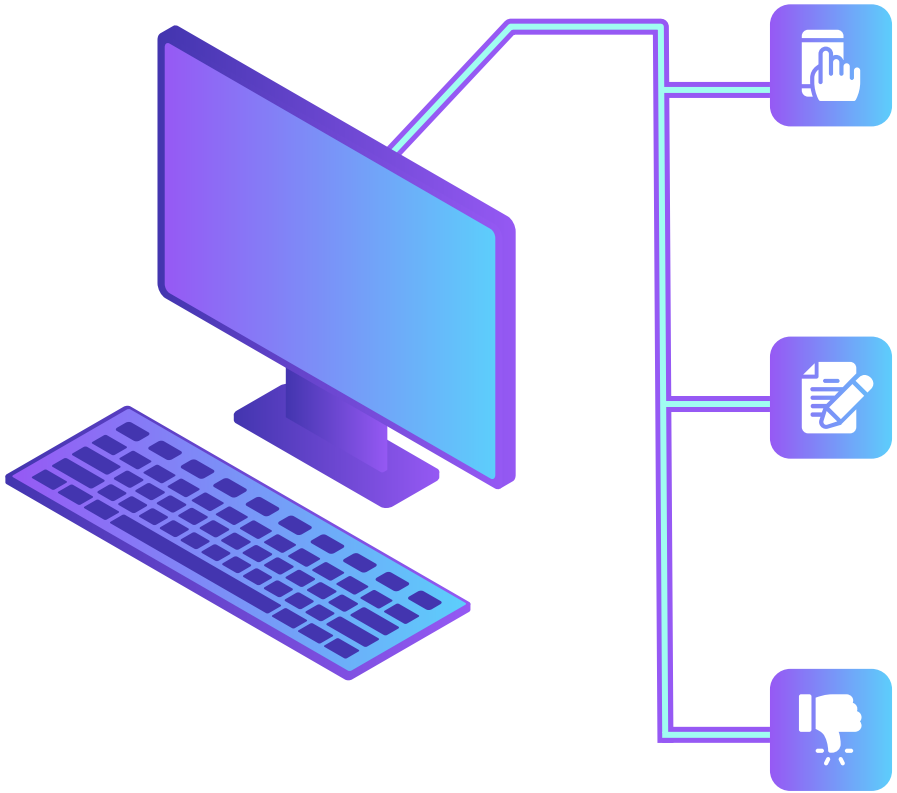
Joás

E-mail:
joassilva@student.dei.uc.pt
Student Number:
2021226149

Filipe

E-mail:
filiperodrigues@student.dei.uc.pt
Student Number:
2021218054

Data migration process infographics



Insert

Examples of data to insert:

- Artist and Consumer Accounts;
- Songs and Albums;
- Playlists;
- Pre-Paid Cards;
- Comments;

Update

Examples of data to update:

- Artist and Consumer Accounts;
- Playlist visibility, name and songs;
- Pre-Paid Cards;
- Consumer Subscription;

Delete

Examples of data to insert:

- Artist and Consumer Accounts;
- Songs, Albums and Playlists;
- Pre-Paid Cards;

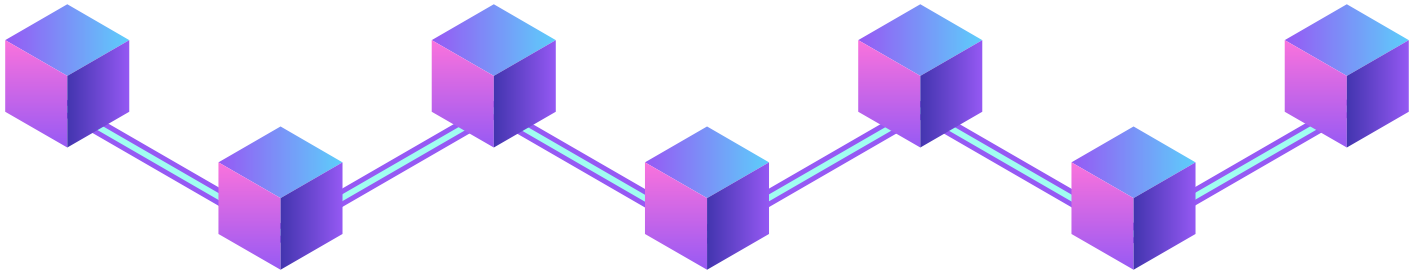
Transactions

- Every insertion/deletion/update operation;
- Subscription plan payment;
- Pre-paid card redeeming/usage;
- The update/creation of the top 10 most listened songs playlist;

Concurrency Conflicts

- When creating an account, the same username can’t be used twice;
- Multiple comments at the same time can cause the comments to be out of order or to be with the wrong content;
- Simultaneous subscriptions, pre-paid cards, playlist, song or album creation or update

Development Plan



Planned Tasks

- Insertion of the necessary data using SQL;
- Postman management;
- Correction of errors;
- Preventing conflicts and data corruption;

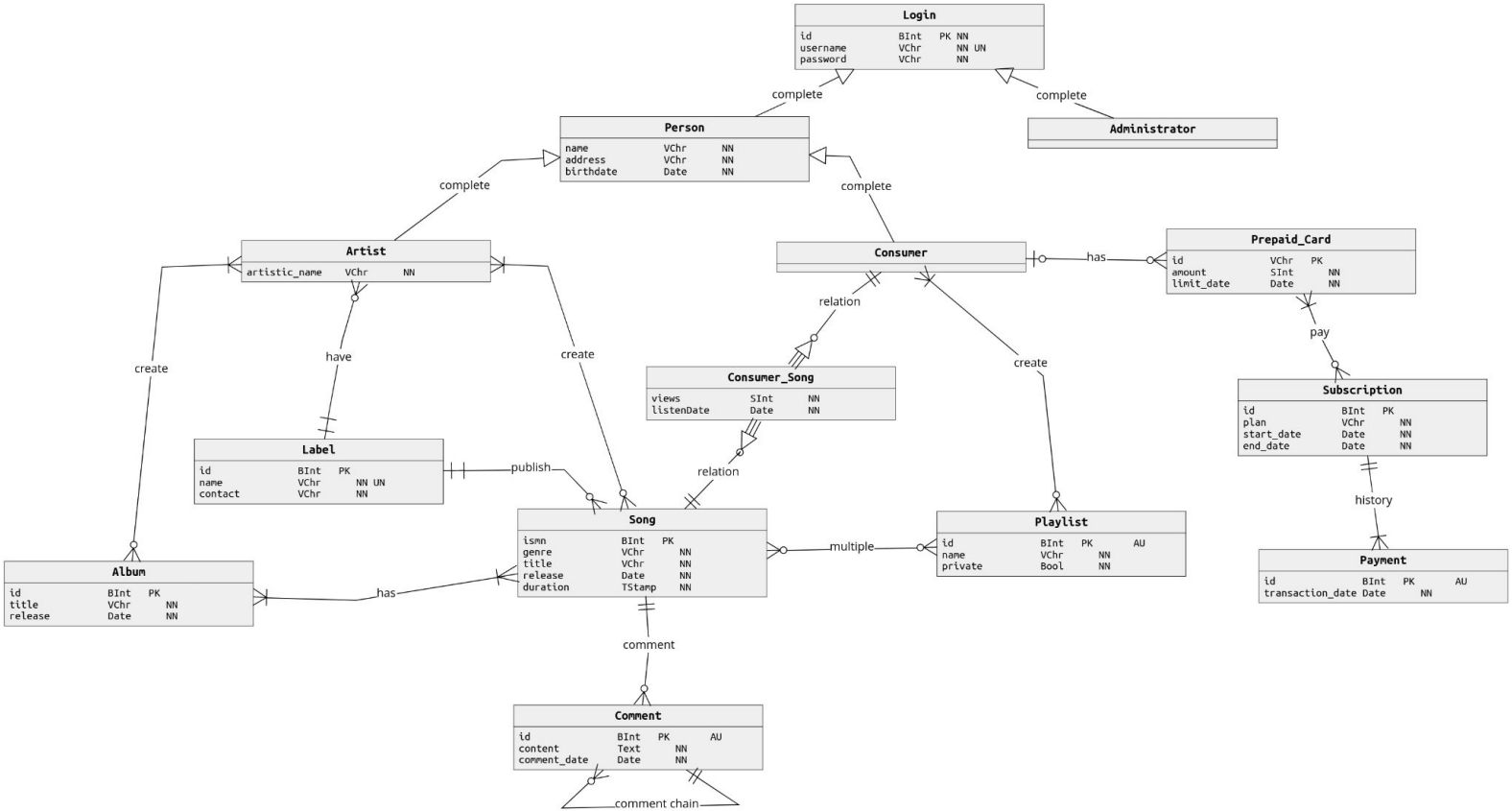
Work Division

Every team member will work equally in each task.

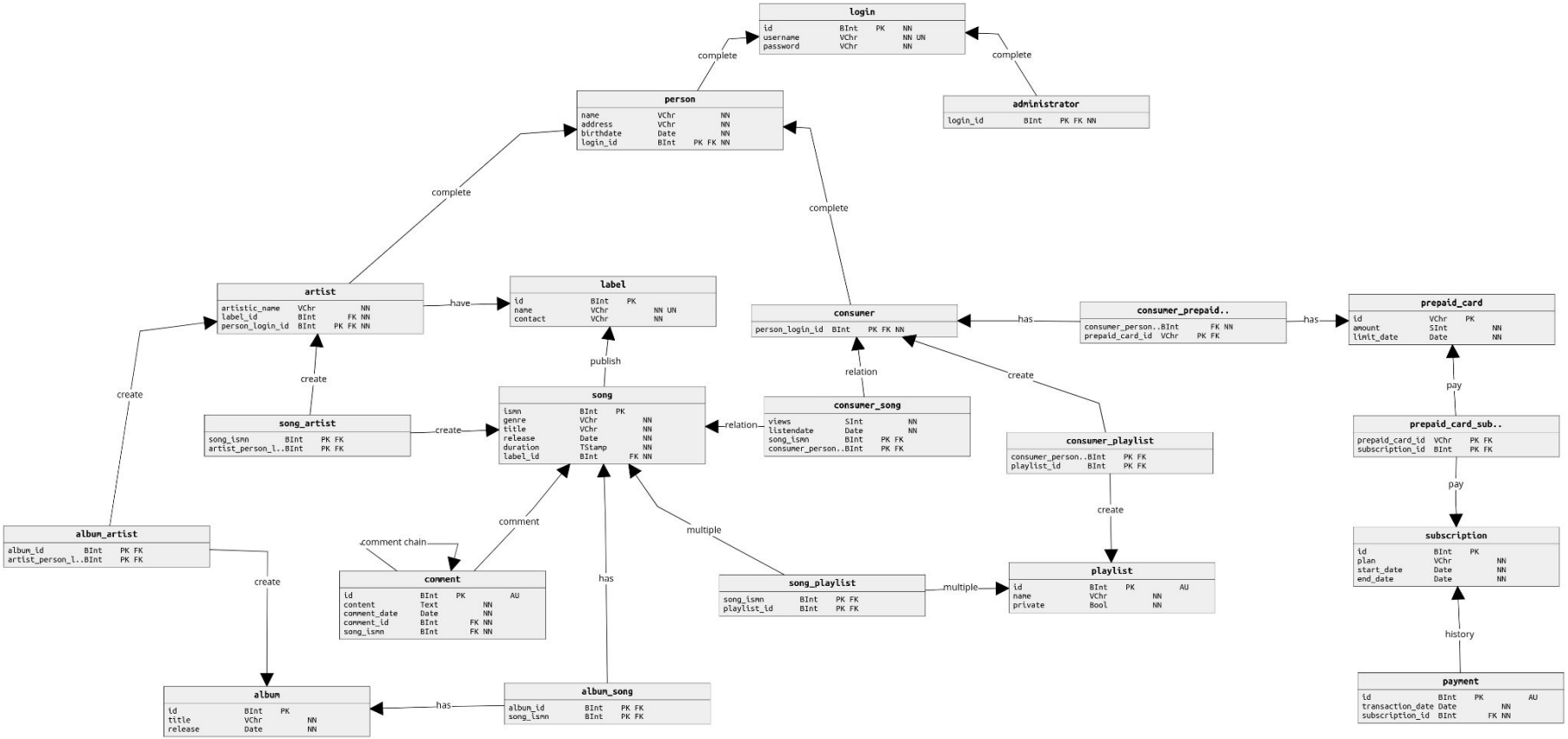
Timeline

Weekly 6 hours of work for each team member.

Conceptual ER



Physical ER


login
id BInt PK NN
username VChr NN UN
password VChr NN
complete
complete
person
name VChr NN
address VChr NN
birthdate Date NN
login_id BInt PK FK NN
administrator
login_id BInt PK FK NN
complete
complete
artist
artistic_name VChr NN
label_id BInt FK NN
person_login_id BInt PK FK NN
have
label
id BInt PK
name VChr NN UN
contact VChr NN
consumer
person_login_id BInt PK FK NN
has
consumer_prepaid..
consumer_person.. BInt FK NN
prepaid_card_id VChr PK FK
has
prepaid_card
id VChr PK
amount SInt NN
limit_date Date NN
create
create
publish
relation
create
pay
song_artist
song_ismn BInt PK FK
artist_person_l.. BInt PK FK
create
song
ismn BInt PK
genre VChr NN
title VChr NN
release Date NN
duration TStamp NN
label_id BInt FK NN
relation
consumer_song
views SInt NN
listendate Date NN
song_ismn BInt PK FK
consumer_person.. BInt PK FK
consumer_playlist
consumer_person.. BInt PK FK
playlist_id BInt PK FK
prepaid_card_sub..
prepaid_card_id VChr PK FK
subscription_id BInt PK FK
pay
create
comment
comment chain
multiple
pay
album_artist
album_id BInt PK FK
artist_person_l.. BInt PK FK
create
comment
id BInt PK AU
content Text NN
comment_date Date NN
comment_id BInt FK NN
song_ismn BInt FK NN
has
song_playlist
song_ismn BInt PK FK
playlist_id BInt PK FK
multiple
playlist
id BInt PK AU
name VChr NN
private Bool NN
subscription
id BInt PK
plan VChr NN
start_date Date NN
end_date Date NN
history
album
id BInt PK
title VChr NN
release Date NN
has
album_song
album_id BInt PK FK
song_ismn BInt PK FK
payment
id BInt PK AU
transaction_date Date NN
subscription_id BInt FK NN