Fall 2023 CS213 Project Part II

> Contributors:
>
> Topic Design: SUN Kebin
>
> Contributors: He Zean, Leng Ziyang, Tang Yulei, Lin Huiyan, Zhang Haoming
>
> Review: Yu Shiqi, WANG Weiyu
>
> This project description is extended from the one of Fall 2022 CS213/CS307.

General Requirement

- It is a personal project. Each student should do it separately and submit one report for the project.
- You should submit the report and the source code before the deadline. **All late submissions will receive a score of zero**.
- **DO NOT copy** ANY sentences and figures from the Internet and your classmates. Plagiarism is strictly prohibited.
- The number of pages for your report should be **between 4 and 16**. Reports less than 4 pages will receive a penalty in score, however, ones with more than 16 pages will NOT earn you more score.

After finishing Project I, it is not difficult for you to get an overview about the advantages of database management system (DBMS) against ordinary file I/O, especially for their performances. In this project, you are required to continue the works done in Project I and finish the following general tasks:

1. Design a new database with permission management to meet the requirements presented by **Synchronized User-generated Subtitle Technology Company (SUSTC)**.
2. Correctly implement the APIs described below which will be used to communicate with your database that include operations requested by different roles in SUSTC. Since the APIs are defined in Java, no other programming language is allowed.
3. Other detailed tasks described below.

Section 1 Background

This project is about the structure of a fictional Danmu video website – Synchronized User-generated Subtitle Technology Company (SUSTC). After decades of development, it has built an ecosystem around users and creators who continuously produce high-quality content including videos and Danmu comments.

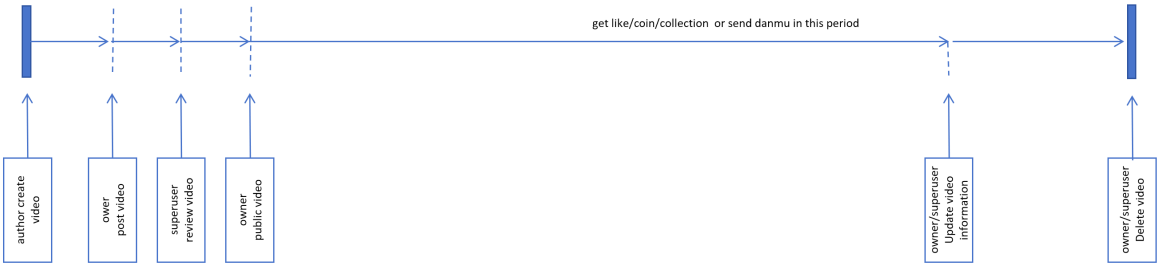
Usually, a video is post by users known as “Uploaders”(also owner). Then, the videos will be sent for review to check if it conforms to standards by users known as “Reviewers.” After a successful review, the Uploader can publish this video at their designated time. The users can then watch the video, during which they can interact with each other through Danmu beyond time and space. Of course, except for sending Danmu messages, the users can also do a series of operations, such as giving coins, liking videos, making videos their favorites.

1.1 Entities of SUSTC

You may create or modified your database as `io.sustc.dto` present.

User: This entity is an important subject which could perform many operations. Each user could own multiple identities. It means that a user can be an owner, an administrator, or an audience, which could be distinguish by different privilege or some additional information.

Video: This entity is a work created or be post by a user, hence it is closely related to its owner. During a video's whole lifetime, it may go through the following stages as shown in the following figure.



Danmu: When an audience watch a video, he/she may send Danmu messages. These Danmu messages can also be changed display mode or be liked.

1.2 Data Description

Attribute information in this part is same as comments in `io.sustc.dto` of github link

`class UserRecord`

mid: the unique identification number for the user

name: the name created by the user

sex: include but not limited to biological sex

birthday: the birthday of the user

level: user engagement evaluated according to system decision criteria.

sign: the personal description created by the user

following: a list holding all the users' mids that this user follows

identity: a value in `{"user", "superuser"}` indicating the user's role

password: The password used when login by `mid`

qq: OIDC login by QQ, does not require a password

wechat: OIDC login by WeChat, does not require a password

`class AuthInfo`

mid: The user's mid

password: The password used when login by mid

qq: OIDC login by QQ, does not require a password

wechat: OIDC login by WeChat, does not require a password

`class UserInfoResp`

mid: The user's `mid`

coin: The number of user's coins

following: a list holding all the users' `mid`s that this user follows

follower: The user's follower `mid`s

watched: The videos' `BV`s watched by this user

liked: The videos' `BV`s liked by this user

collected: The videos' `BV`s collected by this user

posted: The videos' `BV`s posted by this user

`class RegisterUserReq`

password: The password used when login by `mid`

qq: OIDC login by QQ, does not require a password

wechat: OIDC login by WeChat, does not require a password

name: the name created by the user

sex: include but not limited to biological sex, with type `RegisterUserReq.Gender`

birthday: the birthday of the user

sign: the personal description created by the user

Gender: the gender type enum class

`class VideoRecord`

bv: the unique identification string of a video

title: the name of video created by video owner

ownerMid: the `mid` of the video owner

ownerName: the `name` of the video owner

commitTime: the time when the owner committed this video

reviewTime: the time when the video was inspected by its reviewer

publicTime: the time when the video was made public for all users

duration: the video duration

description: the brief text introduction given by the owner

reviewer: the `mid` of the video reviewer

like: a list holding the `mid`s of the users who liked this video

coin: a list holding the `mid`s of the users who have given this video their coins

favorite: a list holding the `mid`s of the users who favored this video

`class ViewRecord`

mid: the users' `mid`s who watched this video

timestamp: last watch time stamp in seconds since the video starts

`class PostVideoReq`

title: The video's title

description: The video's description

duration: The video's duration (in seconds)

publicTime: The video's public time. When posting a video, the owner can decide when to make it public. Before the public time, the video is only visible to the owner and superusers. If the video is already published (when this DTO is used in `io.sustc.service.VideoService.updateVideoInfo(AuthInfo, String, PostVideoReq)` update a video), this field should be ignored.

`class DanmuRecord`

bv: The danmu's video `BV`

mid: The danmu's sender `mid`

time: The danmu's display time (in seconds) since the video starts

content: The danmu's content

postTime: The danmu's post time

likedBy: The users' `mid`s who liked this danmu

1.3 Notices

This case of SUSTC has a few of entities and simple relation. According to the operations, you should be careful about the timing sequence of the operations and the impact of an operation on states.

1.4 Important Notice

To make sure that your work is not in vain, your report must contain the basic information of yourself(Names, student IDs).You should submit at least one version to Blackboard, and check the files carefully. We will grade the last submission version. No late submissions will be accepted.

Section 2 Tasks

2.1 Personal Information

The personal information(Names, student IDs) should be written clearly in report

2.2 Database Design (15%)

As mentioned above, please make a new **E-R diagram** (5% out of 15%). Please follow the standards of E-R diagrams.

Then design the tables and columns. You shall generate the database diagram via the tool in Datagrip “Show Visualization” and embed a snapshot or a vector graphics into your report (5% out of 15%).

Besides, you shall briefly describe the design of the tables and columns, and submit the database user creation and privilege descriptions (5% out of 15%).

2.3 Basic API Specification (70%)

To provide basic functionality of accessing a database system, you are required to build a backend library using Java which exposes a set of application programming interfaces (APIs). The detailed specification for each API is described in github link.

Note that the APIs are defined in a series of Java interfaces. Hence, you are not allowed to define your own API instead of predefined ones nor allowed to use any programming language other than Java. Please refer to the actual Java interface file if there were any conflict between this description and the provided Java interface. If not stated particularly, all the input parameters are IMMUTABLE.

Furthermore, it is NOT permitted to use file I/O to manipulate the data provided by the APIs to improve the performance. You MUST use JDBC to interact with the database to manipulate the provided data.

The score of the API part is mainly judged by automatic benchmark, but we recommend detailing the special design you made in the report. It is important to make sure your code to run completion, and there is a base score.

2.4 Advanced APIs and Other Requirements (15%)

Base on this basic API, you can open explore on one or several of the ways to make your code more faster, more strong, more security, or other optimization design. Well organized experiments should be detailed in your report.

Section 3 How to Test Your Program

Please follow the instructor on github link

Section 4 How to Submit

Submit the report named “Report_sid.pdf” in **PDF format** and **a `.zip` archive**(detail in github link) on the **Blackboard** website. You should submit all above before 23:59 on 31 December 2023, Beijing Time (UTC+8), the presentation will be arranged on 16th week.

Section 5 Disclaimer

The characters, businesses, and events in the background of this project are purely fictional. The items in the files are randomly generated fake data. Any resemblance to actual events, entities or persons is entirely coincidental and should not be interpreted as views or implications of the teaching group of CS213.