SUSTECH CS307 Fall 2024 Project Part II

> **Contributors**: Liu Tengfei, Li Zhaohan, Liu Jinrun, Zhou Xingyu, Gong Mingdao, Wang Zhong-Qiu, Wang Weiyu
>
> The final interpretation of this project belongs to the course group.

General Requirement

- This is a group project with same group members as Project I. Each group should finish the project independently and submit only one report. Generally, it is NOT allowed to change teammates grouped in Project I. Also **no contribution ratio**.
- 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 tasks:

1. Design a new database with permission management based on the one you finished in Project I to meet the new requirements presented by **Scholarly Universe System to Check (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 advanced tasks described below.

Section 1 Backgroud

This project is about the structure of a publication management system named Scholarly Universe System to Check (SUSTC), which is similar to Google Scholar. The provided data contains basic information about articles, and it is complete (that is, all the cited articles of an article can be found in the provided data). The data is an excerpt of the full NIH NLM PubMed 2024 Baseline. For those who are curious about the pre-processing step which leads to this dataset, see scripts here.

1.1 Entities of SUSTC

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

1.2 Data Description

Attribute information in this part is same as comments in `io.pubmed.dto`

`class Article`

id: Article's main id, pmid.

authors: List of article's authors.

title: Article's title.

keywords: List of article's keywords.

journal: Journal of this article.

pub_model: Article's pub_model.

references: List of article's references to other articles.

article_ids: List of article_ids.

publication_types: List of article's publication types;

grants: Grants awarded to this article.

created: `java.util.Date` of the creation date of this article record.

completed: `java.util.Date` of the completion of this article.

`class ArticleID`

ty: Type of article's ID.

id: Id of this type of article ID.

`class Author`

fore_name: Forename of the author, if there has one.

last_name: Lastname of the author.

initials: Intials of the author, if there has one.

collective_name: Collective name of the institution when the author is not a person.

`Grant`

id: Id of grant.

acronym: Acronym of the awarding institution.

country: Country of the awarding institution.

`class Journal`

id: Id of the journal.

title: Title of the journal.

country: Country of the journal registered in.

issn: The ISSN of the journal.

issue: `JournalIssue` tells where this article sits in its journal.

`class JournalIssue`

volume: Volumn of the journal that contains the article.

issue: Issue of the journal that contains the article.

`class PublicationType`

id: Id of the publication type.

name: Name of the publication type.

1.3 Notices

This case of Sustc has a few of entities and simple relation. According to the operations, you should be care about the timing sequence of the operations. Also 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 your group and workloads:

1. Names, student IDs, and the lab session of the group members.
2. The contributions and the percentages of contributions for each group member. Please clearly state which task(s)/part of the task(s) is/are done by which member in the group.
3. If you failed to link a task or its sub-tasks to one of the group members, we will NOT count the score for the part you miss (since we do not know who accomplished this task).

You group should submit one version to Blackboard, and check the files carefully. No late submissions will be accepted.

Section 2 Tasks

2.1 Group Introduction

The 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%).

Also 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 javadoc of the services, see README.md for more info.

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, and no exceptions shall be thrown by the APIs.

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.

In this part, it is mainly judged by benchmark, but we recommend detailing the special design you group made in the report.

> Hint: Because many APIs depend on citation count of articles, it's recommended to create *an additional temp table* that stores the citation count of each articles. By doing so you only need to maintain the correctness of these intermediate data in this temp table during initial populating, updating and deleting, and all citation-related computation can be omitted during most of the ordinary API handlers, thus improves the system performance.

2.4 Advanced APIs and Other Requirements (15%)

Based on the APIs defined above, you can easily implement a complete backend/server system that receives and deals with the requests from the frontend. It is not mandatory that you implement a GUI frontend, however, it must include the login, logout, permission control and other basic functionalities specified above. You are free to use any programming language(s) with any network communication protocol(s) to implement the frontend and the backend after finishing the basic requirements in Java.

Section 3 How to Test Your Program

Please follow the instructions in README.md.

Section 4 How to Submit

Submit the report named “Report_sid1_sid2.pdf” in **PDF format** and **a .zip archive** (detail in README.md) on the **Blackboard** website. There are two submission batches for you group to choose:

1. before 23:30 on 22th December 2024, Beijing Time (UTC+8), the presentation will be arranged on 15th week. The score will multiplied by 1.1;
2. before 23:30 on 29th December 2024, Beijing Time (UTC+8), the presentation will be arranged on 16th week.
3. It's suitable to delegate any member of the group to present the report, if the non-participating member provides the letter of authorization.

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 course group of CS307.