SUSTECH CS307 Fall 2025 Project Part II

> **Contributors**: Li Lingfei, Peng Jiaxin, Shi Jimao, Cao Li, Huang Yifan, 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 E-R graph** Base on the given database with permission management to meet the new requirements presented by **Sus**tainable **T**echnology for **C**ook(**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. **Write a report** and **give a presentation**.
4. Other advanced tasks described below.

Section 1 Introduction

This project is about a recipe website named **Sus**tainable **T**echnology for **C**ook(**SUSTC**), which is devoted to promoting a sustainable lifestyle of cooking by oneself. In Project II, the website, who's data has been clearly and accurately revised , is now hiring your team to build a new backend system for them. For those who are curious about the pre-processing step which leads to this dataset, see scripts here.

1.1 Connection with Database

As shown in `sustec-runner.compose.yml` and `sustc-runner.src.main.resources.application.yml`, you need to use user `sustc` with password `sustec` to connect the database `sustc` first.

1.2 Data Description of SUSTC

The data description is showed in `sustec-api.src.main.java.io.sustc.dto` (`sutec-api.....dto` for short) . In this project, we offer you the implemetation of build the database and load data base on the information as `sutec-api.....dto` present. Also, you may create or modified your own database by replace the implement class in `sustc-api.src.main.java.io.sustec.service.impl` (`sustc-api.....service.impl`).

In the version offered, we release a small-scale benchmark data for you to test locally, while in final test, we use the full-scale benchmark data. So you need to ensure that your implementation logic is correct, rather than just the correctness of the data.

1.3 Examples for Benchmark

We also provide some examples of benchmark tests, whose format and logic you can refer to, in `sustc-api.....service.impl`. Your inplemetations of the services in `sustc-api.....service` also are put in direction `sustc-api.....service.impl`. To raise the high performance, you can change the code of implementation we offered as you like, as long as the inteface remains unchanged.

1.4 Notice

You can learn the code's structure hierarchy step by step as follow:

- To achieve the implemetations of the service in `sustc-api.....service`, you only need to focus the codes in direction `sustc-api.....service` , rather than the whole hierarchy.
- To learn how the benchmark hierarchy works, you need to read extra codes in direction `sustc-runner.src.main.java.io.sustc.benchmark`, (`sustc-runner.....benchmark` for short).
- To create your own personalized tests, you need to be familiar with the codes in direction: `sustc-runner.....benchmark` and `sustc-runner.....command`
- Keep all the interfaces unchanged.

Section 2 Tasks

2.1 Group Introduction (shown in report)

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.

2.2 Database Design (15%, shown in report)

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

Then 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%, implemented in code. The high point shown in presentation and report)

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 code comment of `sustc-api.....service`.

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.

2.4 Advanced APIs and Other Requirements (15%+10%, shown in presentation and report)

Based on the APIs defined above, in order to encourage exploration, you can choose any advanced requirement as follow working well(include design, result, analysis and conclusion) to get full mark(15%) of this part. The more you do, the more marks you earn(another 10%).

- Encapsulate the features and implement a real back-end server instead of several independent scripts or programs. The server should provide socket-based communication or HTTP/RESTful web services;
- Apply database connection pools;
- Big data management;
- Page display design;
  - A usable and suiable GUI design or webpage design for data presentation.
  - Giving a wonderful input and output format display based on the command line.
- Proper use of user privileges, procedures, indexing, and views in a reasonable manner;
- Support high-concurrent with proper pressure tests.

Section 3 How to Test Your Program

Please follow the instructor on github link.

Section 4 How to Submit

Submit the report named “Report_sid1_sid2.pdf” in **PDF format** and **a .zip archive**(detail in github link)on the **Blackboard** website:

1. before 22:00 on 30th December 2025 Beijing Time (UTC+8), the presentation will be arranged on 16th week.
2. 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 teaching group of CS307.