Group 11 weekly report

Data Science and Big Data Technology Class 1
Group member：汪嘉仪 姚慧琳
Date：2023.07.3

▼ 1. Week 1 Understand the framework and project framework construction (2023.05.29-2023.06.04)

- 1. About framework understanding and selection
- 2. Construct E-R diagram according to function
- 3. Project initialization

▼ 2. Week 2 Client back-end development Front-end interface design (2023.06.05-2023.06.11)

- 1. Member module development
- 2. Add public to module common
- 3. Front-end environment construction

▼ 3. Week 3 transfer algorithm design and implementation and passenger management development (2023.06.12-2023.06.18)

- 1. Passenger function module development
- 2. Transfer station function development
- 3. Package code template

▼ 4. Week 4 management terminal module development and page design (2023.06.19-2023.06.28)

- 1. Create the required data table
- 2. Basic data management functions
- 3. Daily information management and update

- 4. Management interface development

1. Week 1 Understand the framework and project framework construction (2023.05.29-2023.06.04)

- ☑ Understand the Springboot framework
- ☑ Conceive project functions, design E-R diagram
- ☑ Project initialization
    - ☑ Added member module
    - ☑ Add http test module
    - ☑ Add gateway module configuration route forwarding

1. About framework understanding and selection

Our group compared some different back-end frameworks. Considering the Springboot framework can quickly build projects, powerful application monitoring, automatic configuration and other advantages, we chose to use this microservice framework.

2. Construct E-R diagram according to function

According to the required functions, we draw the entity-relation diagram of the database table design of the project, and check whether it is redundant and whether the design is reasonable.

3. Project initialization

- Use the idea editor for project development, create a large `ticket-system` project, create a member submodule, and modify the log to facilitate cooperative development.
- Add a http test file to test whether the project can run normally.
- Add a gateway module and configure routing and forwarding to facilitate subsequent authentication on the gateway and reduce the number of interactions between the client and each microservice.

2. Week 2 Client back-end development Front-end interface design (2023.06.05-2023.06.11)

- ☑ member module development
    - ☑ Create a member table according to the E-R diagram
    - ☑ Integrate the persistence layer mybatis to access the database
    - ☑ SMS verification login function
- ☑ Add common module common

  - ☑ Add exception handling
  - ☑ Add tool class
  - ☑ AOP print request parameters
- ☑ Create a CLI section to build a front-end environment

1. Member module development

- Add the table creation statement of the member module, and add entity classes according to the data table. Compared with UUID, self-incrementing ID and other algorithms, finally use the snowflake algorithm to generate the data table ID
- Integrate the persistence layer mybatis, connect to the database and access it successfully
- Encapsulate request parameters and return results, integrate axios to initiate requests, add axios interceptors, and use local variables to save member information. Add route interception login, use JWT single sign-on. When logging in, verify whether the mobile phone number is 11 digits and whether the verification code is correct

2. Add public to module common

Store public configuration files, tool classes, AOP, interceptors, etc. in this module. Add exception handling code, which can handle exceptions uniformly or custom exceptions.

3. Front-end environment construction

Use Vue CLI to create web modules, integrate Ant Design Vue, and perform CLI multi-environment configuration. After successfully logging in, jump to the main interface that has been built, and add sidebar and navigation bar components to the user interface.

3. Week 3 transfer algorithm design and implementation and passenger management development (2023.06.12-2023.06.18)

- ☑ passenger module development
  - ☑ Build the passenger table according to the E-R diagram
  - ☑ Added passenger function
  - ☑ Delete, edit passenger function
  - ☑ Add pagination function
- ☑ Relay station function development
  - ☑ Add related table creation statement
  - ☑ Transit algorithm design and implementation
  - ☑ Add front-end interface
- ☑ Package code template

1. Passenger function module development

According to the E-R map, build the passenger table, develop the functions of the controller and other layers, realize the functions of adding, querying, and deleting passengers, and add the refresh function to ensure the consistency of front-end and back-end data. Use pagehelper to realize the function of query paging
When adding a passenger, you need to enter an ID number, and encrypt sensitive information like this when it is transmitted from the front end to the back end

2. Transfer station function development

Design two different algorithms to realize the transfer station function: Dijkstra algorithm and brute force traversal method to find the path with the shortest time. Set transfer conditions: at least 20 minutes between two trains for passengers to transfer. Build train schedules, city tables, etc. based on the E-R map, and develop the front-end interface, use the sub-table nesting method to display the information of each train, and realize the ticket purchase function

3. Package code template

In order to improve development efficiency, the commonly used CRUD codes are packaged into ftl files; templates such as controller, service, queryreq, and queryresp are added, and DbUtil is used to read table field information and add entity class generators

4. Week 4 management terminal module development and page design (2023.06.19-2023.06.28)

- ☑ Add the required data tables such as seat table and carriage table
- ☑ Basic data management
    - ☑ Add basic train number, seat, station management functions
    - ☑ Optimize related functions, such as automatic pinyin recognition, calculation of stop time, etc.
- ☑ Daily information management, update
    - ☑ Integrated quartz to generate timing tasks
    - ☑ Complete the development of daily train numbers, seats and other functions
    - ☑ Add the function of automatically generating train numbers and seats according to the specified date
- ☑ Management interface development

1. Create the required data table

According to the E-R diagram, add table creation statements for data tables such as carriage table and seat table to facilitate subsequent development

2. Basic data management functions

Realize operations such as addition, deletion, and editing of stations, train number information, carriage information, seat information, etc. Optimize related functions, such as introducing pinyin to automatically recognize the pinyin and initials of the input station name; adding functions to automatically calculate the stop time, etc.

3. Daily information management and update

Use springboot to complete the timing task test, integrate quartz to complete the automatic generation function, and you can specify to generate data in the next few days. Add the manual generation function, and you can choose the specific date for generating information. Test it, the test found that the automatically generated data can be successfully written to the database

4. Management interface development

- Add components such as drop-down box components, design the layout of different interfaces, and complete the development of the query interface for basic data and daily data
- Design the welcome interface design of the management terminal, and successfully realize the front-end and back-end interaction