JAVAEE PROJECT

博影(BOYING)票务娱乐平台

Instructor:Fan Hongfei

1851632 Shi Jiasheng
1751022 Li Cuiqi
1854081 Fu Cheng
1851486 Fang Hao

目录

CONTENTS

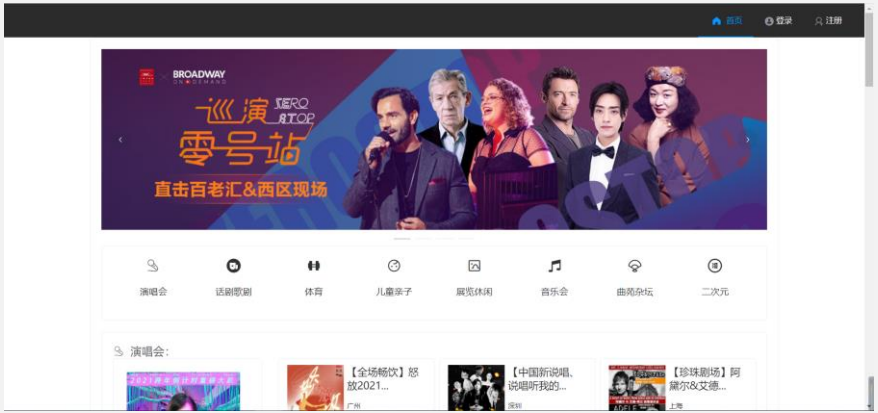
01

PART 01

Introduction

Introduction

Our project is an entertainment ticket marketing platform, with business covering concerts, dramas, musicals, sports events and other fields. The purpose is to allow users to safely, quickly and easily purchase tickets for various performances and improve user experience.The system is desperate to two parts,the first is user interface,the other one is administrator interface.



02

PART 02

Function Realization

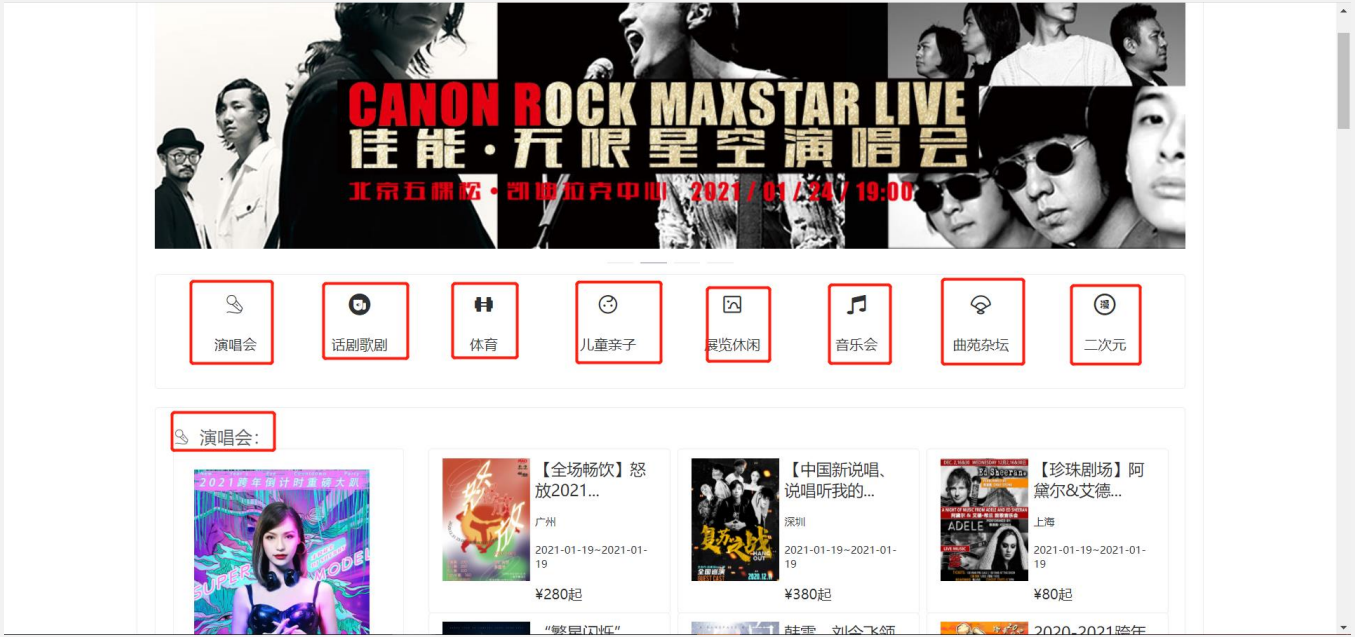
Overview of function realization

User interface function

Function	Remarks
Register	Use phone number to register an unique account
Login	User can login by three ways
Update personnel information	User can change some information except some that can’t be modified
Search for shows	User can search for shows by many kinds of conditions
Buy tickets	User can choose different tickets to buy,for different tickets,the prices are different
Manage orders	User can view all orders he has ordered and do some operations to them.
Refund tickets	User can refund tickets

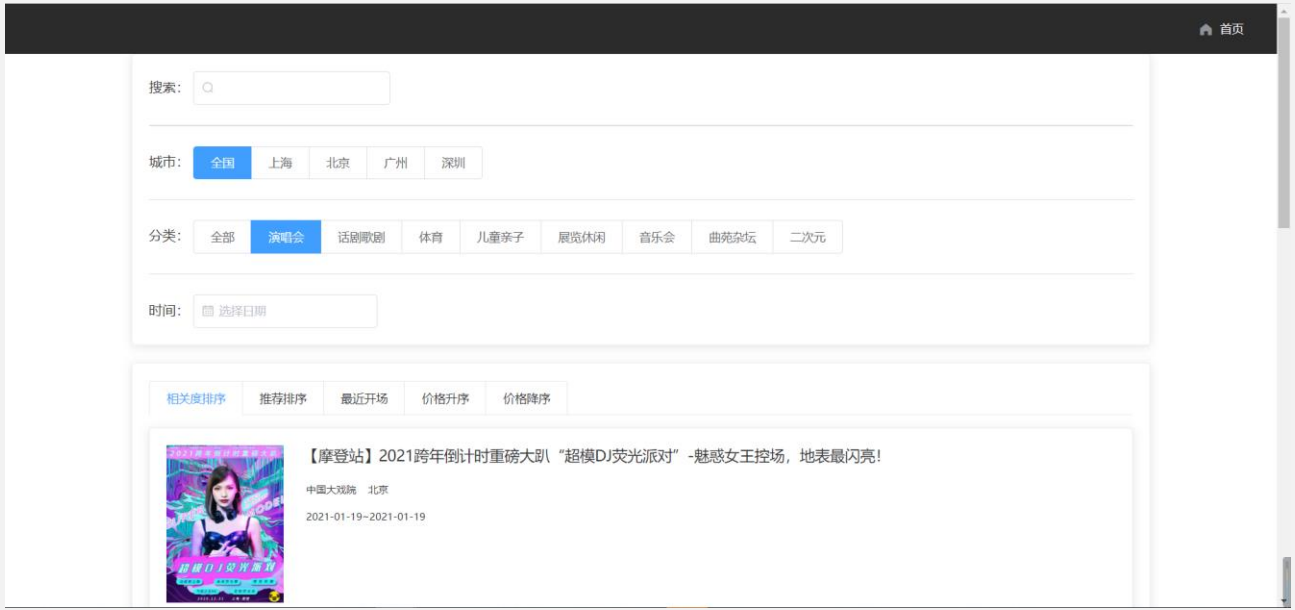
Search for shows

User can click any category to jump to the search page.Click different category will get a different initial search show page for user



Search for shows

Here we can enter search name to search show in the search bar.In the next line, we can choose different cities to filter the search result. Also we can choose different categories ,time and different sort condition to change the search result



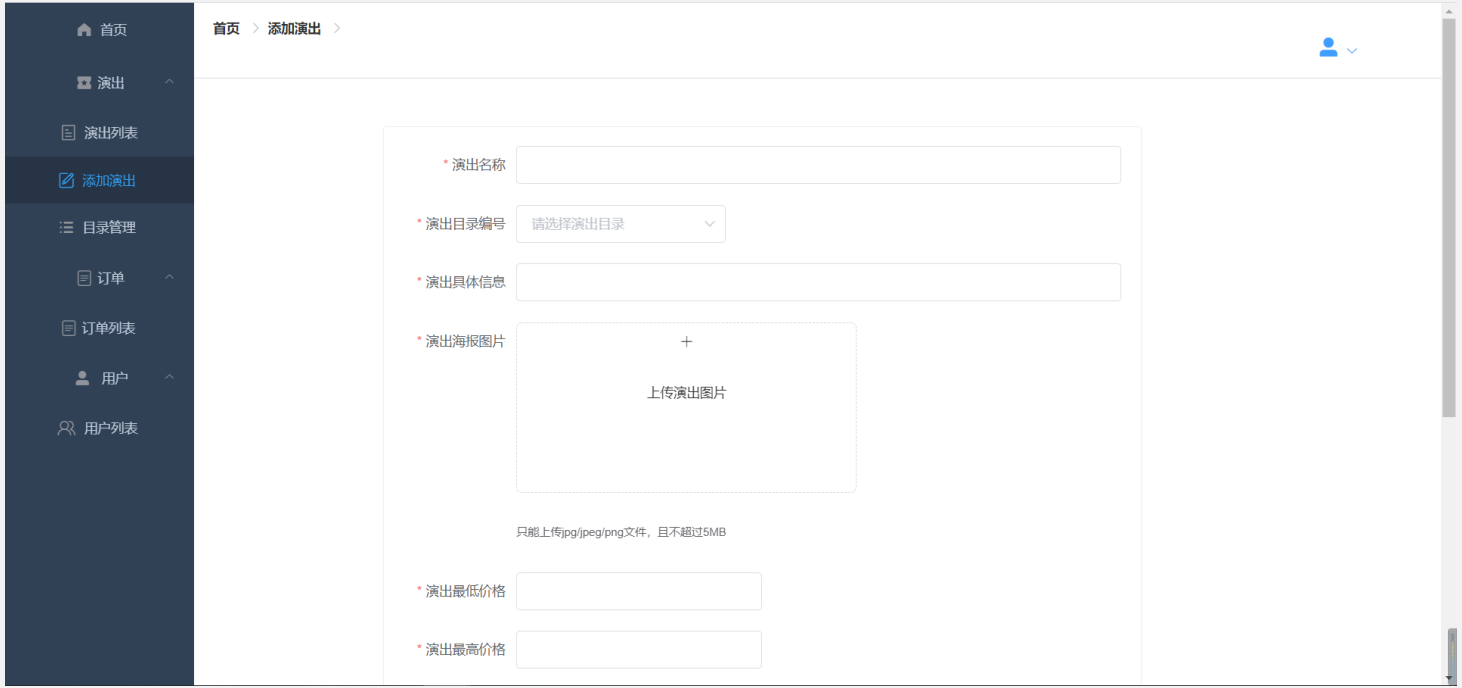
Overview of function realization

Administrator interface function

Function	Remarks
Login	Admin can login by only one ways
View statistics information of platform	Admin can view some information of platform like order and user information statistics
Manage shows	Admin can view platform shows and delete shows
Add new shows	Admin can enter new show’s information to add new shows to platform
Manage show category	Admin can view ,edit , delete and add category
Manage orders	Admin can manage all users’ orders
Manage users	Admin can manage all users on the platform

Add shows

Here we need to finish this form to add a new show, conditions with star mark must be given. Here you should enter show name, poster, category, price and other information, then you can add this new show to database and reflect to user interface



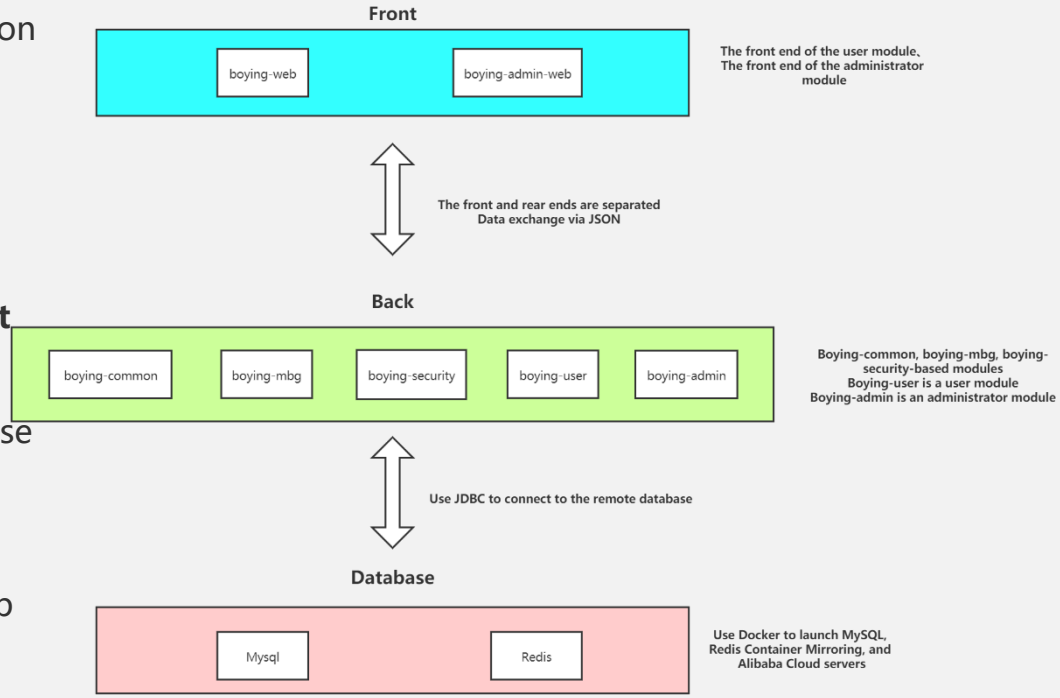
03

PART 03

Architectural and Component Design

The overall structure of the project

- The boying project is a front-end separation project
- the front-end is developed by **Vue**
- The back-end is developed by **SpringBoot**
- the database uses the relationship database **Mysql** and the memory database **Redis**
- The front-end mainly includes boying-web (user module) and boying-admin-web (administrator module)



The project deployment address

- Project address (user interface):
http://47.103.203.188:8080/boying-user/index.html

- Project address (management page):
http://47.103.203.188:8080/boying-admin/index.html

- Back-end Swagger interface address (user module):
http://47.103.203.188:8000/swagger-ui.html

- Back-end Swagger interface address (administrator module):
http://47.103.203.188:7000/swagger-ui.html

The Front-end

User front

```
├─assets
|   ├─config about the images and the aliyun's api
|   ├─css:total css of the web
|   └─img:the images of the web
├─components: some componets of the web
├─icons
|   └─iconfont
├─router: router of the web
├─store
├─utils
└─views: the pages of the web
```

admin front

```
├─assets
|   ├─config
|   └─css
├─components
├─icons
|   └─iconfont
├─router
├─store
├─utils
└─views
```

The back-end project structure

The back-end mainly includes boying-common (universal code module), boying-mbg (mybatis generator module), boying-security (security authentication module), boying-user (user module), boying-admin (administrator module)

```
boying
    ├── boying-admin The background management module
    ├── boying-common Encapsulate the project's common code module
    ├── boying-mbg MyBatisGenerator automatically generates database operation
code
    ├── boying-security SpringSecurity permission authentication common code
module
    ├── boying-user Ticket purchase module for the show at the front desk
```

The back-end Technical selection

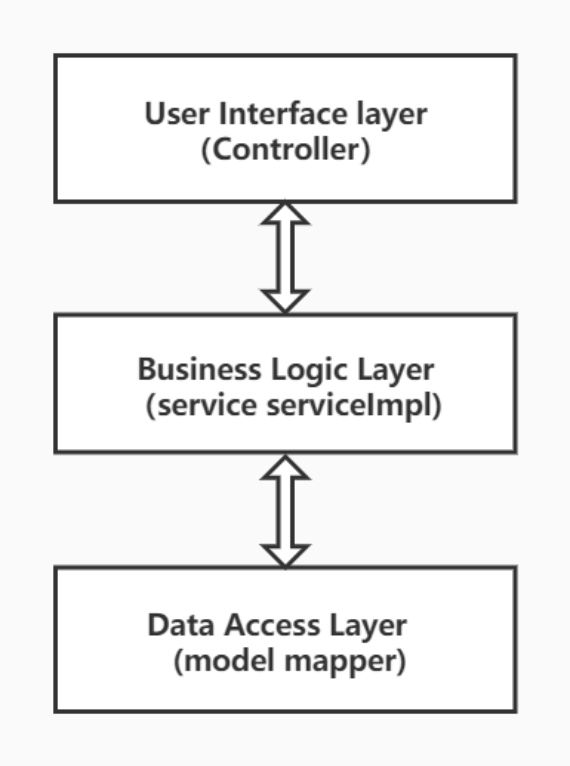
Technology	Version	Description
JDK	1.8	Java Development Kit JDK
Spring Boot	2.3.0	Containers and MVC frameworks
Spring Security	5.1.4	Certification and authorization framework
MyBatis	3.4.6	ORM framework
MyBatisGenerator	1.3.3	The data layer code is generated
PageHelper	5.1.8	MyBatis physical pedding plug-in
Swagger-UI	2.9.2	API Document production tools
Redis	5.0	Distributed caching
Docker	18.09.0	Apply the container engine
Druid	1.1.10	The database connection pool
JWT	0.9.0	JWT login support
Lombok	1.18.6	Simplify object encapsulation tools
MySQL	5.7.30	A related database

Three-tier architecture

- Boying back-end is a project based on SpringBoot

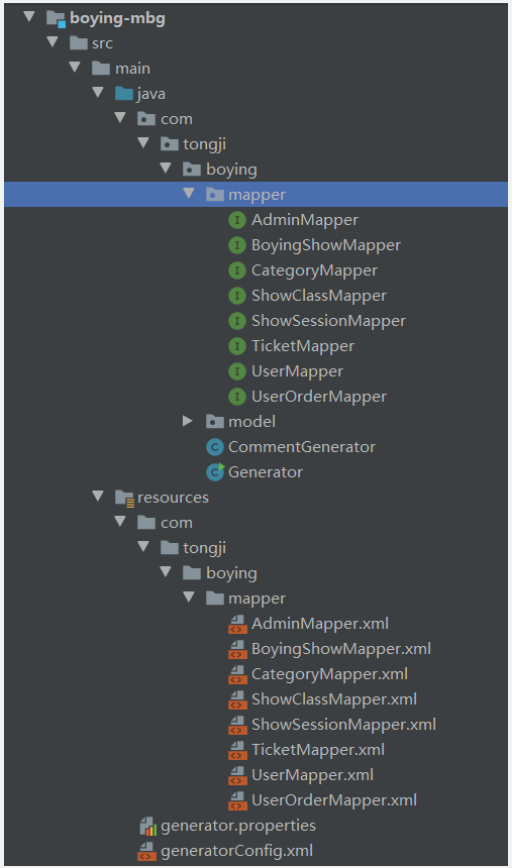
- The boying back-end project is developed based on the idea of a three-tier architecture

- Data Access Layer（mapper), Business Logic Layer(service), User Interface layer(controller).



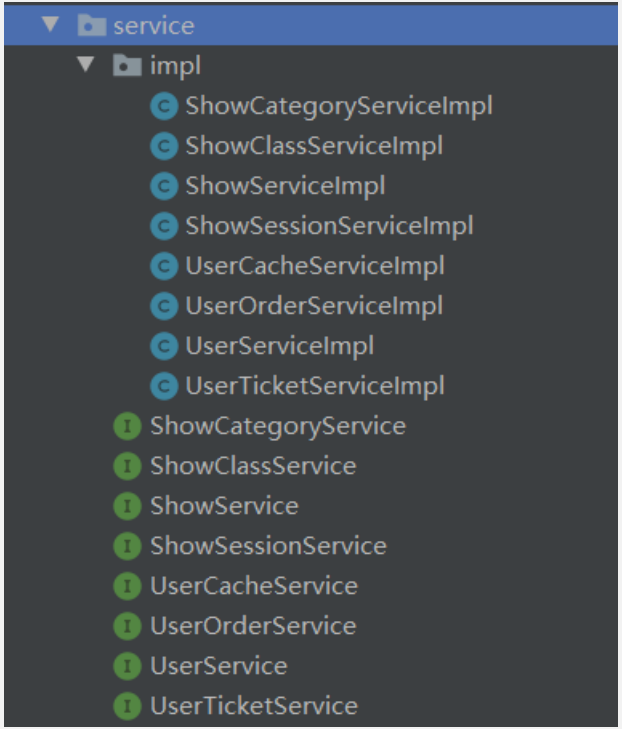
Data Access Layer

Data Access Layer is used for database access (additions and deletions). We used Mybatis and encapsulated Data Access Layer in a boying-mbg sub-module for unified management of the Data Access Layer code.



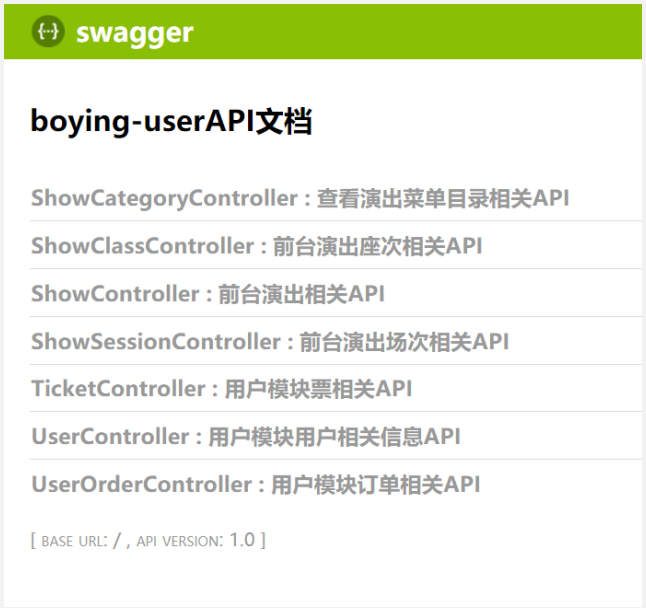
Business Logic Layer

Business Logic Layer Is used for business processing and data transmission, Data Access Layer from the processing processing, and data to the User Interface layer. We used a separate approach between the service interface and the service implementation, so that the business logic would be easier to scale.



User Interface layer

Primarily responsible for presenting the data processed by the Business Logic Layer to the user. And we use Swagger for easy back-end debugging and front-end use.

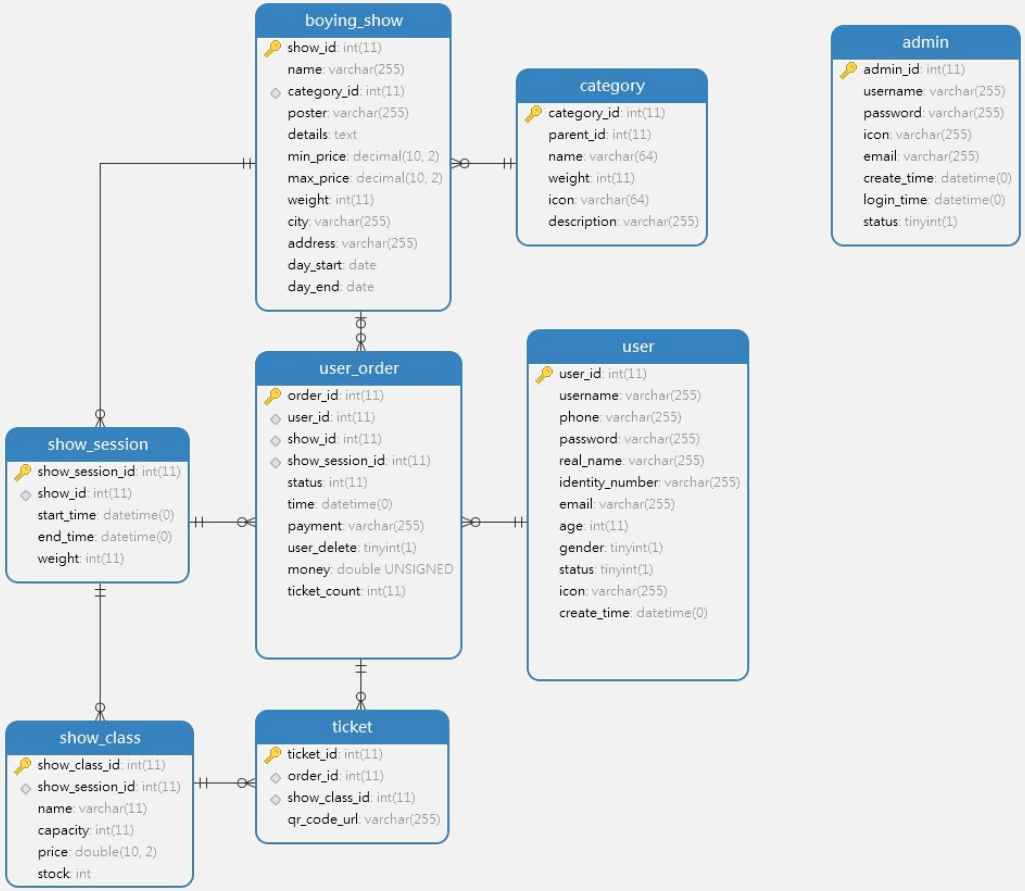


04

PART 04

Database Design

Database design

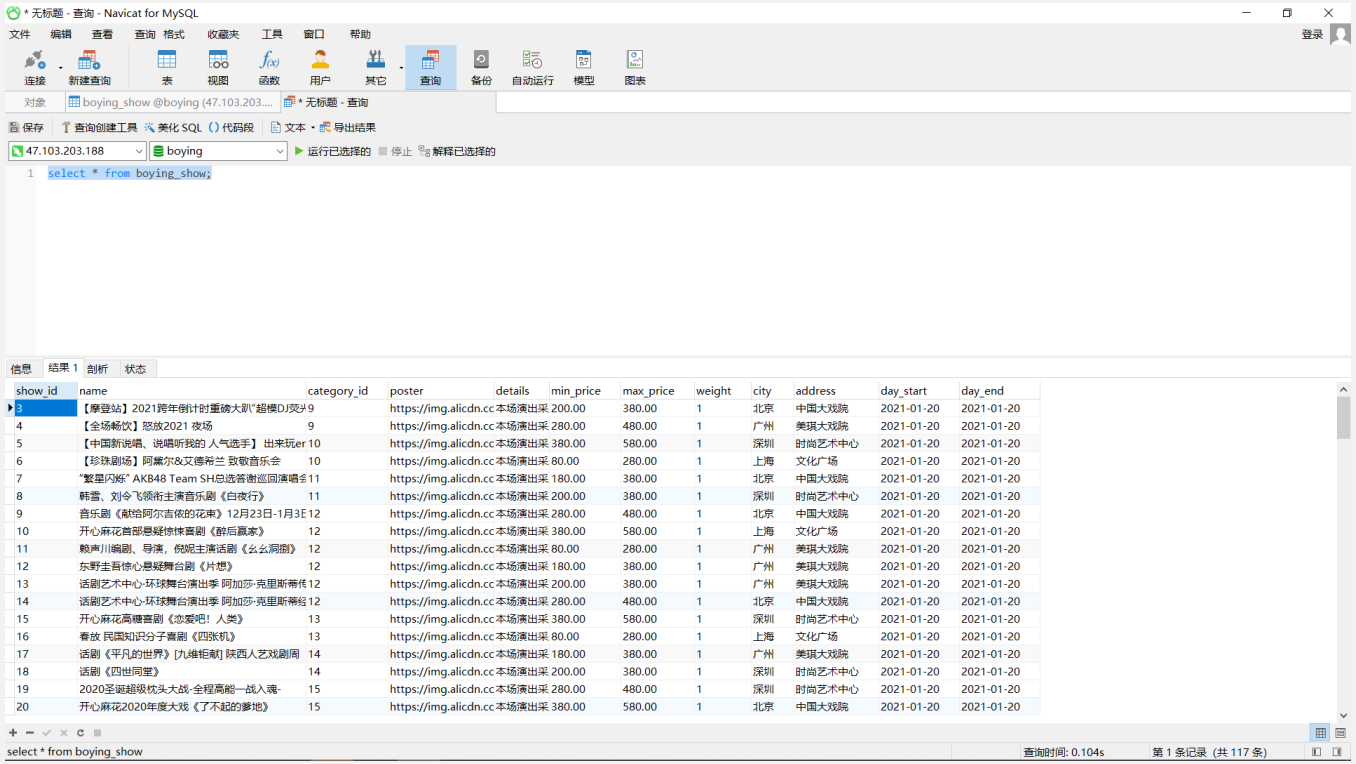


user_order Table

user_order

PK/FK	Field	Type	Comment
PK	order_id	int(11) NOT NULL	
FK	user_id	int(11) NOT NULL	The user Id that belongs to
FK	show_id	int(11) NULL	Owned by the show Id
FK	show_session_id	int(11) NOT NULL	Id of the show session
	status	int(11) NOT NULL	Pending viewing, completed, returned orders (1,2,3)
	time	datetime NOT NULL	The time the order was submitted
	payment	varchar(255) NULL	The payment method for the order
	user_delete	tinyint(1) NOT NULL	Whether the order is visible to the user, that is, whether the user deleted the order
	money	double(10,2) unsigned zerofill NULL	The total amount of the order
	ticket_count	int(11) NULL	The total number of tickets

Navicat for MySQL



05

PART 05

Collaboration Among members

Collaboration among team members

Github Collaborative Development

GitHub is used for version management during the development of the system

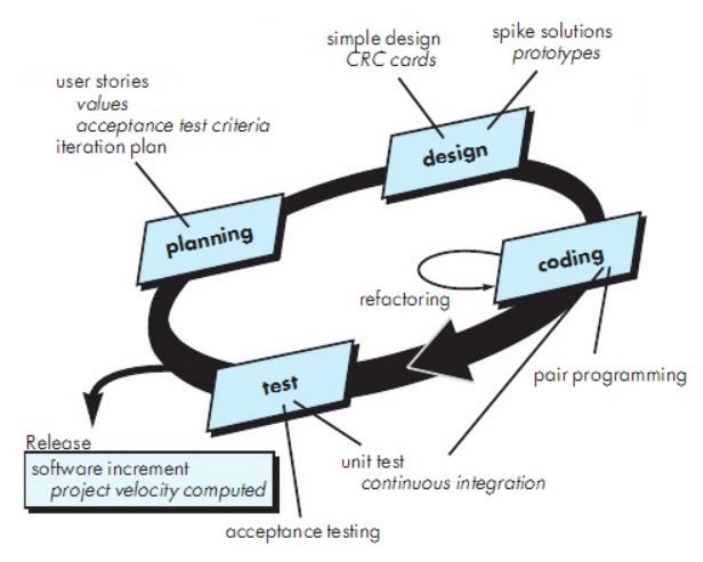
We also use git GUI like Sourcetree and Github desktop to manager the version of the codes better

Separation of frontend and backend

The project is separated into frontend and backend and in order to quickly develop a runnable project, we develop frontend and backend at the same time.

Collaboration among team members

Agile development



Before we release a software increment, we

1.Planning
Gather the requirements and determine what to do.

2.Design
Provide the implementation guidance for the requirement.

3.Coding
Do the coding, cooperate by Github.

4.Test
Debug and ensure the project meet the requirement.

Github development specifications

1. The commit is made uniformly under the branch of the develop.

2. Before each commit (preferably before writing code), pull the last version of code to avoid code conflicts.

3. Each submission should describes where the changes have been made.

4. Follow a uniform code style (Alibaba Java Coding Guidelines)

06

PART 06

Process of the development

Process of the development

1. Determine the topics of the project, implementation function points, table design, business logic and other information by team discussion.
2. Design the database table structure with Navicat-15 for MySQL.
3. Build the initial project environment for the SpringBoot-Mybatis environment.
4. Integrate Swagger, Redis and other development tools.
5. Install docker on the cloud server and download the Redis, Mysql image to start the container.
6. Build the different modules of the project.
7. The frontend is projected through Vue and the API is viewed through the backend Swagger interface and interacts with the backend with Axios.
8. Once the project is debugged and completed, it is deployed uniformly on the Alibaba Cloud.

07

PART 07

Problems and solutions encountered

Front and Back Aspects

- To log in with Swagger, be sure to configure the Bearer token
- Alibaba Cloud OSS CORS issues require cross-domain-related configuration in the Alibaba Cloud console
- Ports such as 7000, 8000, 8080 need to be configured for the server, otherwise they cannot be accessed

Cache Aspects

If you modify the database directly, the data in the Redis cache may still exist, so you need to manually empty the cache after you modify the database data

```
# Connect to the cloud server command line
docker exec -it redis redis-cli -a "redis"
flushall
```

THANKS!