NEU Study- Meeting Room Reservation System

README

Ying Tuo, Xi Shen, Yuqin Luo, Huixin Huang

Khoury College of Computer Sciences

Northeastern University, San Jose, CA 95138

Table of Contents

Overview

NEU Study - Meeting Room Reservation System is a system designed for Northeastern University students to reserve meeting rooms. When students want to go to the library to study or discuss with their classmates in a meeting room, students often need to go to several rooms to find an available one. This system provides a meeting room reservation function, which allows students to easily find available rooms and make reservations so that they do not need to waste their valuable time finding rooms. We used microservice structure, consensus algorithm (Raft), message queue, digital signatures (MD5) and other techniques to develop it as a distributed management system. This reservation system has great performance in highly available, reliable, resilient, easy to use, and efficient.

Configurations Settings

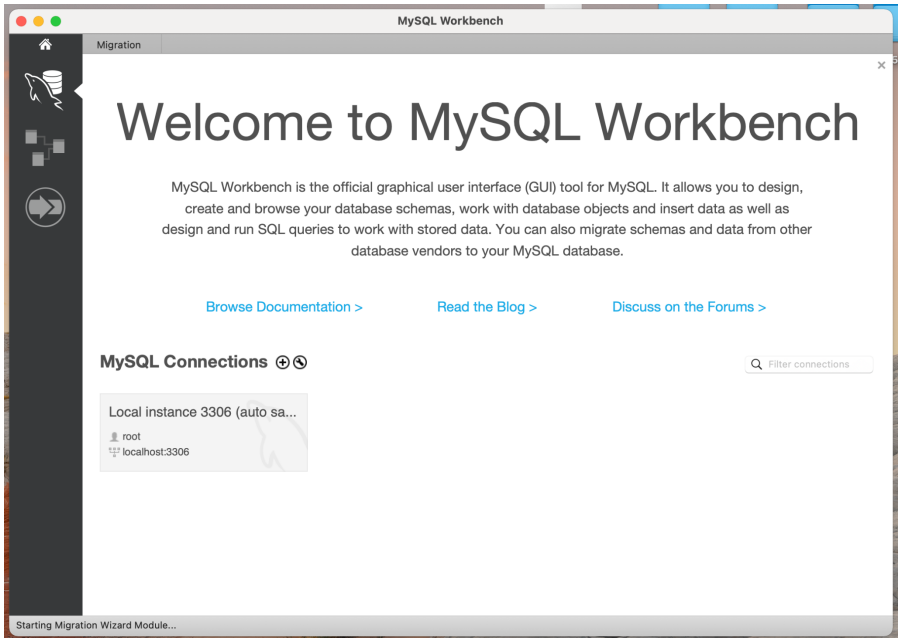
1. MySQL

Our MySQL database is running on localhost:3306, with the usernam root and no password.

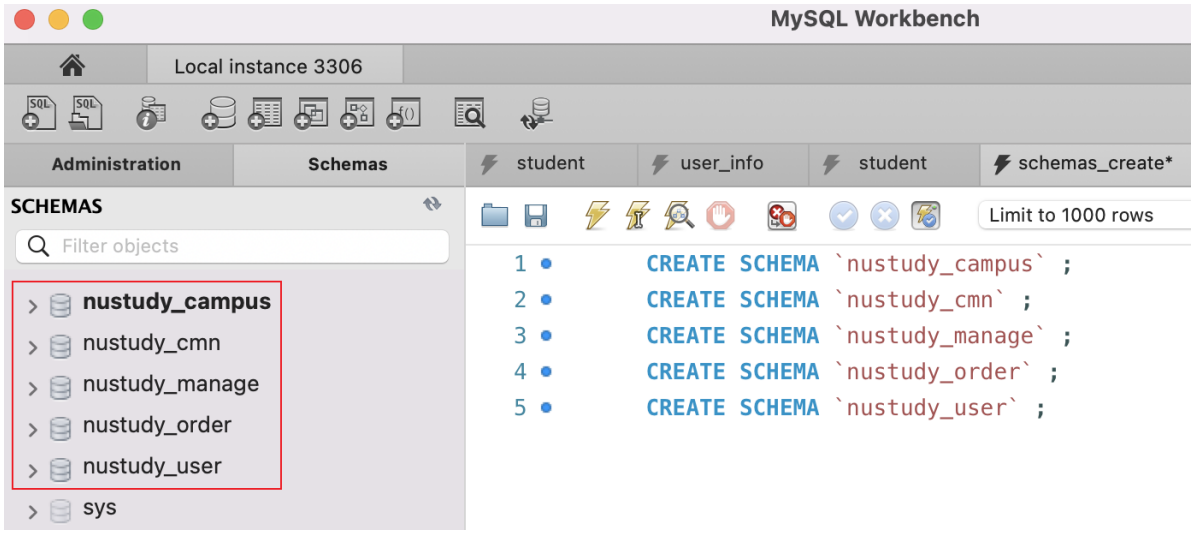
download MySQL and MySQLWORKBENCH.

Run MySQL at default port 3306.

Create a local instance with address: localhost:3306. Set username as root, and password [empty].



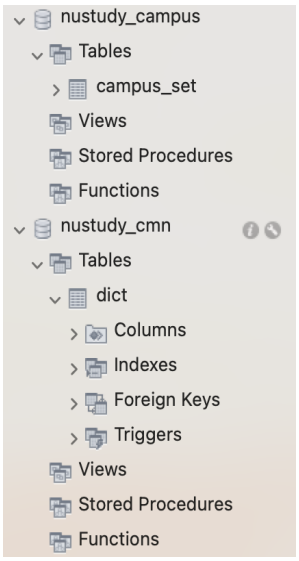
Open `prePropuateData/mysql_preData/schemas_create.sql` file in the package, run the following MySQL codes to Create MySQL database schemas, and we can get five schemas.

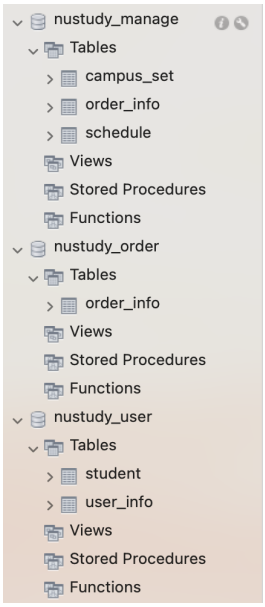


Create all tables (under `prePropuateData/mysql_preData folder)

Run nustudy_campus.sql in schema nustudy_campus
Run nustudy_cmn.sql in schema nustudy_cmn
Run nustudy_manage.sql in schema nustudy_manage
Run nustudy_order.sql in schema nustudy_order
Run nustudy_user.sql in schema nustudy_user

After running all sql in our mysql_preData file, the DB should be like:

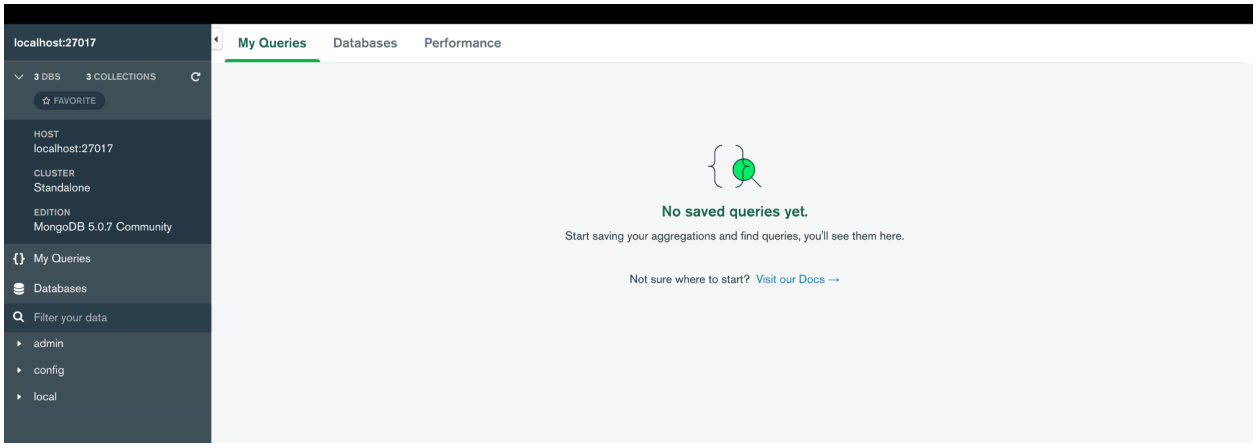




For now, we have set our MySQL database.

2. MongoDB

Download MongoDB and run it on default port 27017.



3. Nacos

Download Nacos(https://nacos.io/en-us/docs/quick-start.html), and start it and end it in the bin directory

Start Nacos: sh startup.sh -m standalone End Nacos: sh shutdown.sh

```
xs@chinasahis-MacBook-Pro ~/Projects/cs6650-final-project/nacos/bin$ sh startup.sh -m standalone                    main
/opt/homebrew/Cellar/openjdk/17.0.2/libexec/openjdk.jdk/Contents/Home/bin/java  -Xms512m -Xmx512m -Xmn256m -Dnacos.standalone=true -
cp .:/Users/xs/Projects/cs6650-final-project/nacos/plugins/cmdb/*.jar:/Users/xs/Projects/cs6650-final-project/nacos/plugins/mysql/*.
jar -Xlog:gc*:file=/Users/xs/Projects/cs6650-final-project/nacos/logs/nacos_gc.log:time,tags:filecount=10,filesize=102400 -Dnacos.ho
me=/Users/xs/Projects/cs6650-final-project/nacos -Dloader.path=/Users/xs/Projects/cs6650-final-project/nacos/plugins/health -jar /Us
ers/xs/Projects/cs6650-final-project/nacos/target/nacos-server.jar  --spring.config.location=classpath:/,classpath:/config/,file:./,
file:./config/,file:/Users/xs/Projects/cs6650-final-project/nacos/conf/ --logging.config=/Users/xs/Projects/cs6650-final-project/nac
os/conf/nacos-logback.xml --server.max-http-header-size=524288
nacos is starting with standalone
nacos is starting, you can check the /Users/xs/Projects/cs6650-final-project/nacos/logs/start.out
```

```
xs@chinasahis-MacBook-Pro ~/Projects/cs6650-final-project/nacos/bin$ sh shutdown.sh
The nacosServer(40530) is running...
Send shutdown request to nacosServer(40530) OK
xs@chinasahis-MacBook-Pro ~/Projects/cs6650-final-project/nacos/bin$
```

4. RabbitMQ

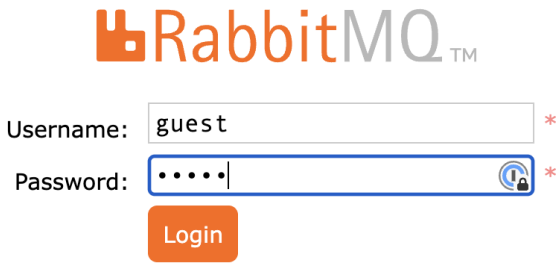
Firstly, you should make sure the RabbitMQ has been installed in your machine:

https://rabbitmq.com/download.html

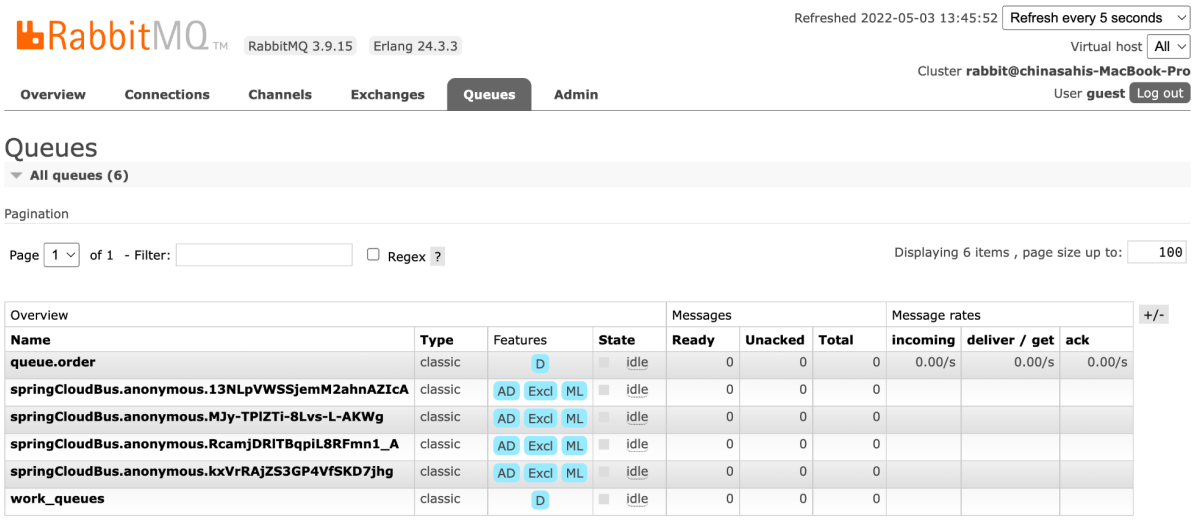
Please also make sure you have also installed erlang and socat as the dependencies of RabbitMQ

RabbitMQ will run on default port 5672, and its management port is 15672.

Run RabbitMQ: brew services start rabbitmq Then go to localhost:15672, and log in with username guest, and password guest.



Message queue is used for updating the available number of schedules.



Run and Test

1. Run back-end firstly, we provide two ways to run our code:

1. Run each jar package separately

   You can run jar files in the folder jars one by one.

```
java -jar service_gateway-0.0.1-SNAPSHOT.jar

java -jar campus_manage-0.0.1-SNAPSHOT.jar

java -jar service_campus-0.0.1-SNAPSHOT.jar

java -jar service_cmn-0.0.1-SNAPSHOT.jar

java -jar service_order-0.0.1-SNAPSHOT.jar

java -jar service_user-0.0.1-SNAPSHOT.jar
```

Hint: when running the jar files failed with noticing the port is already in use, please use

lsof -i :port -> to get the PID.

And run kill -15 PID

to kill the process runing on the port.

2. Or you can run the run.sh to start all jars together at jars directory by script.

```
jars > sh run.sh

ct/jars$ sh run.sh
```

If you want to exit, you can command+C, and **kill** all ports.

```
>lsof -i : 8201  -> get its PID1

>  kill -15 PID1

>lsof -i : 8202 -> get its PID2
```

> kill -15 PID2

>lsof -i : 8203 -> get its PID3

> kill -15 PID3

>lsof -i : 8206 -> get its PID6

> kill -15 PID6

>lsof -i : 80 -> get its PID0

> kill -15 PID0

>lsof -i : 9998 -> get its PID8

> kill -15 PID8

```
xs@chinasahis-MacBook-Pro ~/Projects/cs6650-final-project/jars$ lsof -i:8202          main
COMMAND   PID USER   FD   TYPE             DEVICE SIZE/OFF NODE NAME
java    83695   xs   13u  IPv6 0x6566929662853fc3      0t0  TCP *:8202 (LISTEN)
xs@chinasahis-MacBook-Pro ~/Projects/cs6650-final-project/jars$ kill -15 83695        main
```

2. Run front-end

1. Run NEU Study Central Server Management website

/finalproject/nustudy_frontend/ nustudy_central_management_site/

At its directory, run: npm run dev

```
corteydeMacBook-Pro:nustudy_central_management_site cortey$ npm run dev
```

Then in brower run localhost:9528

2. Run NEU Study Room Reservation System

At the directory nustudy-site, run: npm run dev

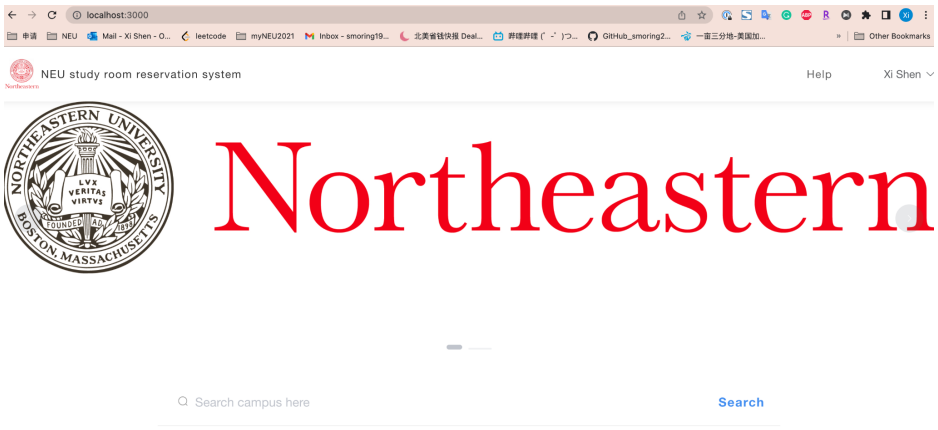
/finalproject/nustudy_frontend/ nustudy_student_site/

```
corteydeMacBook-Pro:nustudy_student_site cortey$ npm run dev
```

Then in brower run localhost:3000

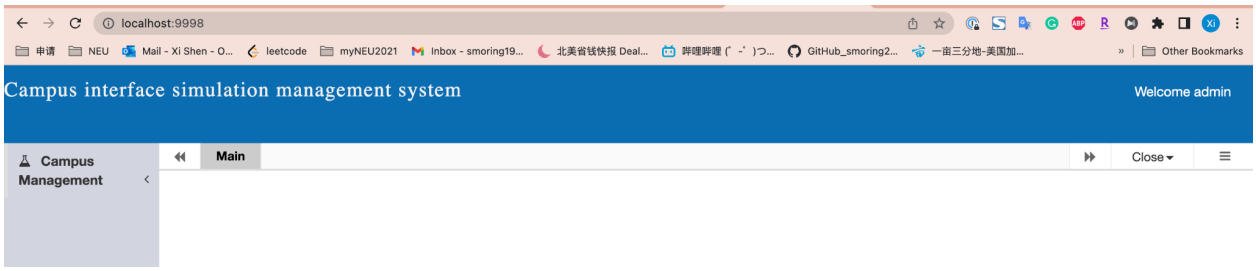
You can use any email of below as the username and **any** password to login.

tuo.y@northeastern.edu shen.x@northeastern.edu huang.huix@northeastern.edu luo.yuqin@northeastern.edu



Run NEU-Silicon Valley Campus

After running the campus_manage of the back-end, Then in brower run localhost:9998

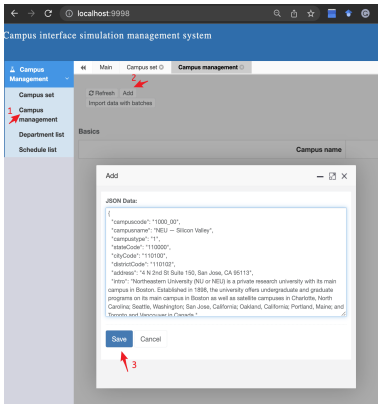
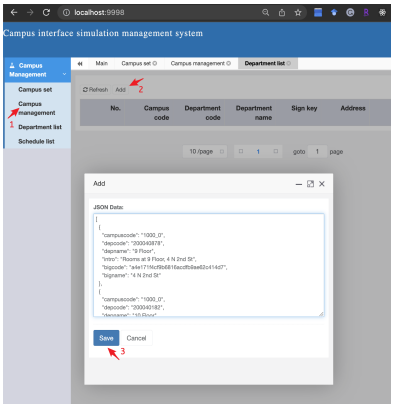
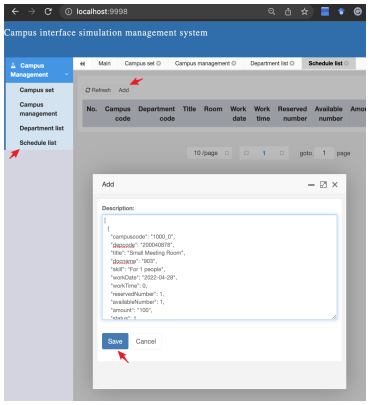


For now, this whole project has run successfully.

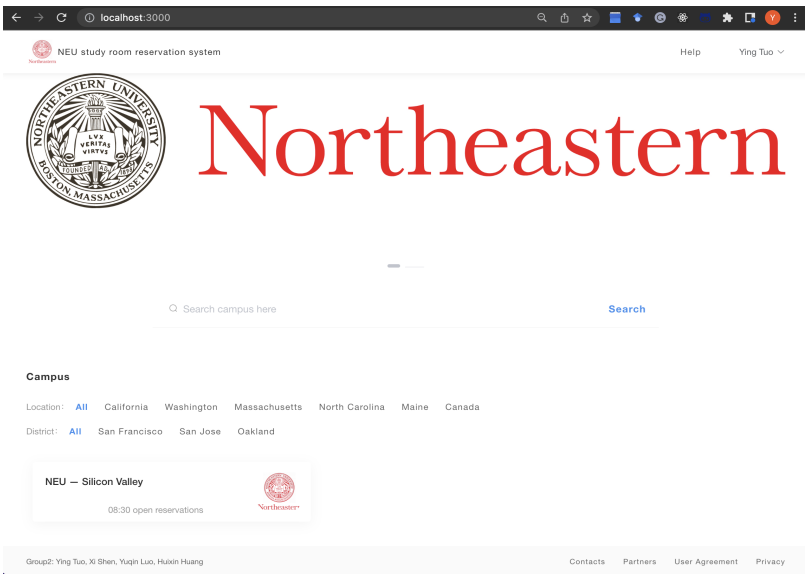
3. Start To Test

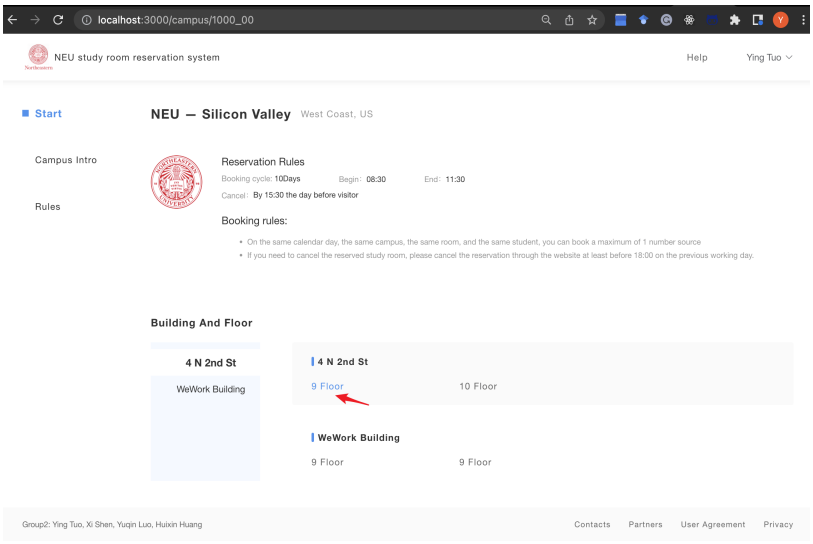
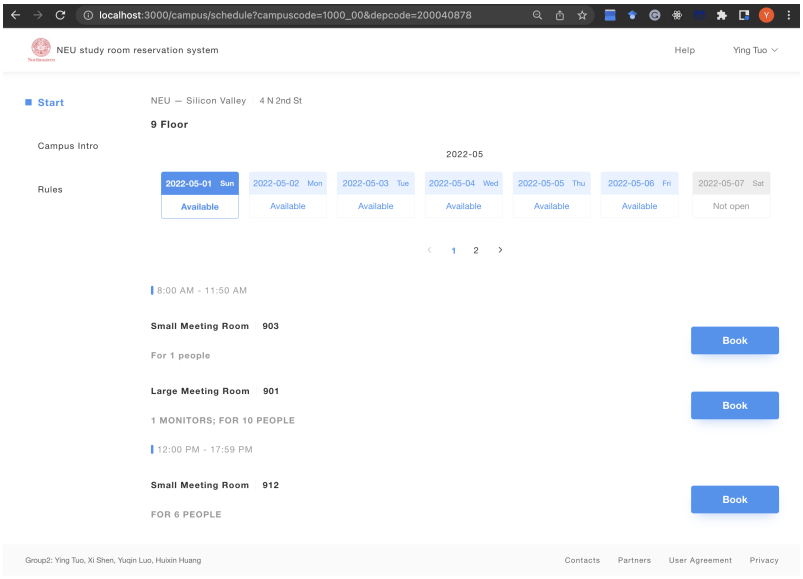
Push dummy data to the MongoDB database

http://localhost:9998/ Campus management:

Campus	Department	Schedule
		

And the data will show on the reservation system.

Start to book	

Choose room	
Book	

Submit & Review