

Online Booking Service for a Sport Centre

background | architectural | software design | software testing | demonstration

Group 06

<i>Baiyan Zhang</i>	2142564
<i>Chengze Liu</i>	2142808
<i>Fuyu Xing</i>	2143763
<i>Jinai Ge</i>	2035693
<i>Mingyuan Li</i>	2145618
<i>Yixuan Wang</i>	2145543
<i>Yue Wang</i>	2141845
<i>Zhenyang Zhao</i>	2142074

Group 6



CATALOG

C A T A L O G

01

Project background ☐

02

Architectural ☐

03

Software design ☐

04

Software testing ☐

05

Conclusion and future work ☐

06

Demonstration ☐

01

Project Background

- Problem Statement
- Aims of The Project
- Project Scope
- User Characteristics
- Assumptions and Dependencies
- Project Risks



Project background

—Problem, Objectives , Scope

Existed Problem

The manual booking process:

- time-consuming
- prone to errors and inefficiency

Project Objectives

- Provide an efficient platform for booking and managing sports activities.

Project Scope

- Supports user registration, activity cataloging, booking functionality, and administrative features.
- Accessible to all staff and students of the university.





Project background

- User Characteristics
- Assumptions and Dependencies ,Project Risks

Administrators

- Activity maintenance
- Booking monitoring
- personnel management
- Statistical report generation.

Users

- Browning sports activities
- Booking sports activities
- Manage sports activities

Assumptions and Dependencies

Project success hinges on:

- Human resources
- Technology infrastructure
- Stakeholder cooperation
- data protection and privacy rules.

Project Risks

Project risks include:

- developmental technicalities
- user adoption
- data security

Measure: Mitigation strategies for project success.

02

Architectural Design

- System Architecture
- High-level Database Design

- Frontend System
- Backend System



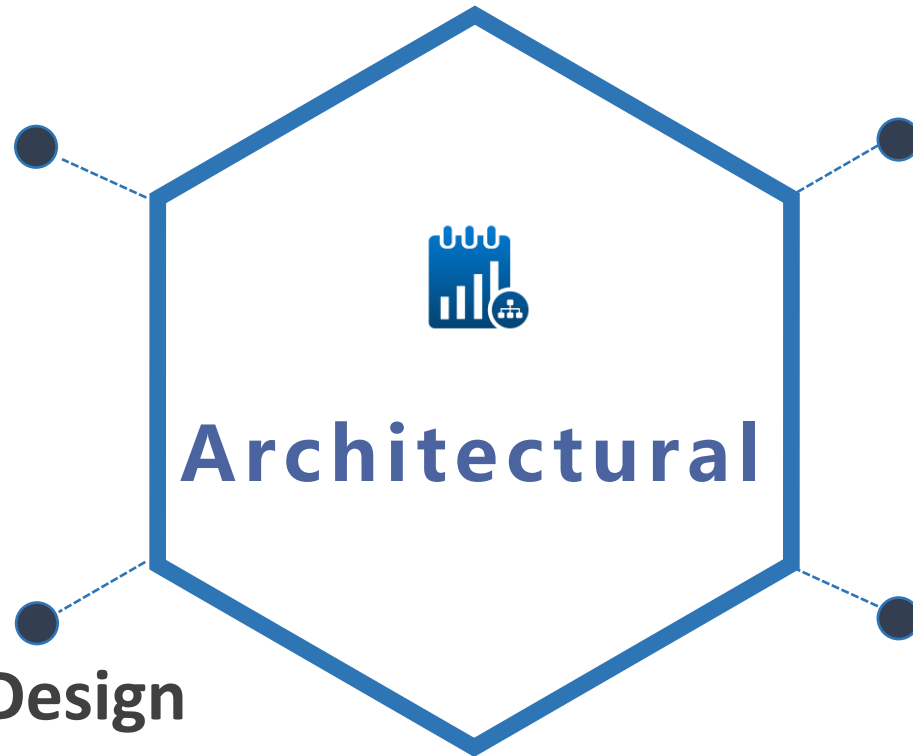
Architectural Design

System Architecture

- MVC

High-level Database Design

- ERD



Frontend System

- HTML/CSS/JavaScript
- jQuery
- bootstrap

Backend System

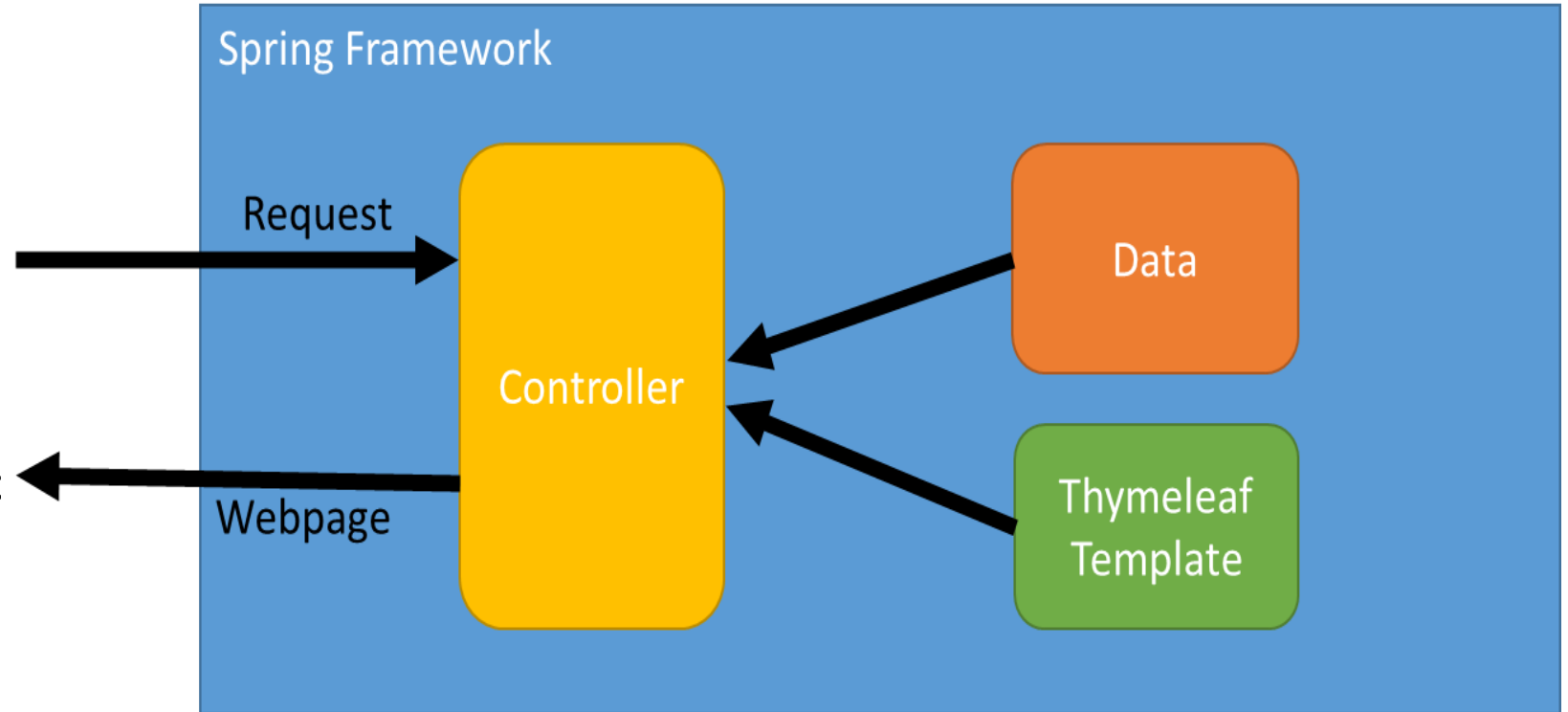
- Spring Boot
- Spring Data JPA & MyBatis Plus
- MySQL
- Supercor

Architectural Design —System Architecture

Model-View-Controller

BENEFITS

- Clear code structure
- Separation of concerns
- Ease of maintenance and scalability
- Separation of business logic and user interface
- Improve code readability maintainability
- Facilitate team collaboration



Architectural Design

—Frontend, Backend System

Frontend

HTML/CSS/JavaScript

Three foundation of web pages

Bootstrap

Open source front-end framework

jQuery

A fast, compact JavaScript library

Backend

Spring Boot

An open source Java-based framework for creating microservices.

Spring Data JPA & MyBatis Plus

A library that automatically connects the data access layer (repository) to the JPA implementation.

MyBatis Plus is an enhancement to MyBatis that simplifies CRUD operations.

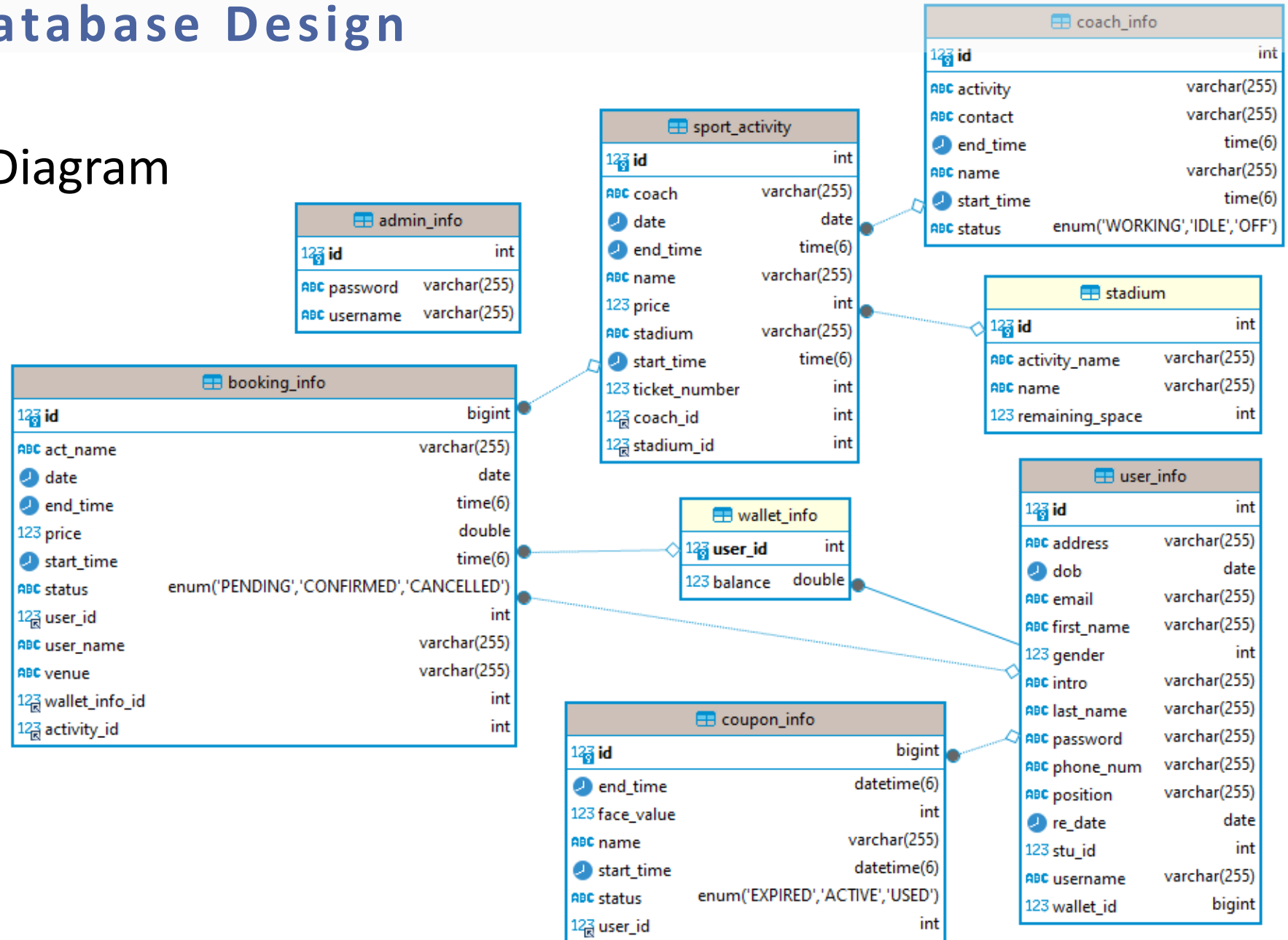
MySQL

A widely used open source relational database management system

Architectural Design

—High-level Database Design

Entity-Relationship Diagram



03

Software Design

- Software Modules
- High-level Process Flow
- Software Support Services
- Coding Structure and Convention
- Software Configuration and Production Environment





Software Design

—Software Modules

System Modularization——Code Logicalization

- ⑩ Enhanced Maintainability, Scalability , Reusability
- ⑩ Reduced Coupling between Modules
- ⑩ Improved Code Readability

Module Characteristics

- Distinct Function
- Independent Operation
- Interaction through Defined Interfaces

Examples:

Login Function and Registration Function ——Mutual independence

Booking Sports Activity Module —— No need to pay attention to the payment module

Software Design

—High-level Process Flow

Basic system functions

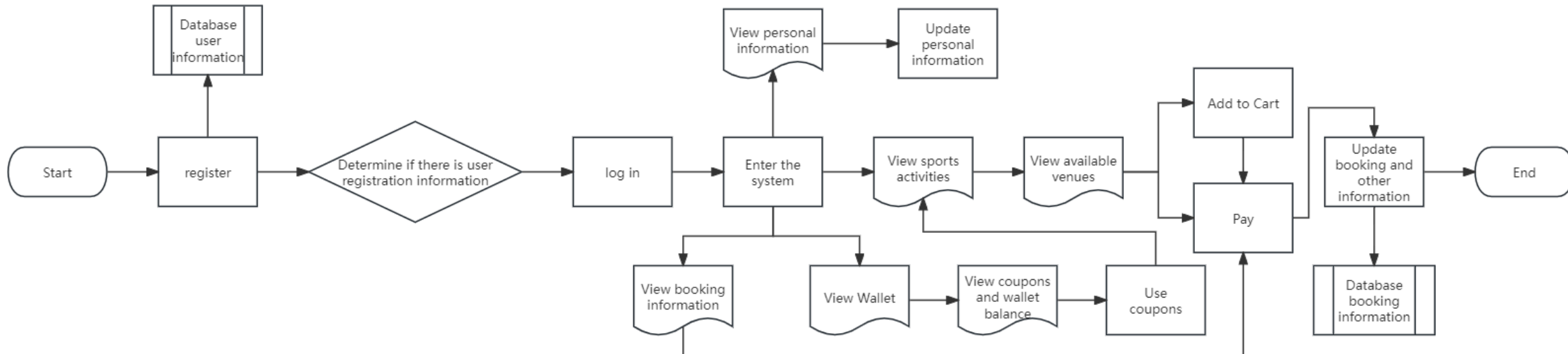
Users:

- Registration and login
- Browsing, booking, paying for sports activities
- Viewing and modifying personal information
- Viewing wallets

Administrator:

- Login
- Manage sports activities, venues, coaches,
- Manage reservations, user information, coupons
- View reports

Flow chart





Software Design

—Software Support Services

Database

- Creation, configuration, monitoring, and maintenance of a database
- Using spring-boot as a framework.
- Utilizing Mybatis-plus

Security

- Implementing strict password security measures during user registration.
- Password reset to prevent theft and ensure compliance with guidelines.

Webpage Navigation

- Web navigation service
- Scrolling banner
- Low account balance alert
- Agreement consent reminder

Software Design

—Coding Structure and Convention

File Organization

- Using GitHub: Team member uploads the contributions and merges them into the main branch following peer review and approval.

Indentation

- Uniform indentation
- Predefined bracket positioning
- Standardized comment placement.

Declarations

- Corresponding classes are created in the controller and model layers.

Statements

- Conditional statements
- Encapsulating related statements within functions.

Naming Conventions

- Using camelCase for naming variables, functions, and front-end file names
- Using a standard class naming strategy

Programming Principles

- MVC architecture (model, view, and controller).
- Correspond to the modal, mapper, and controller folders.



Software Design

—Software Configuration and Production Environment

- Development Environment
- Dependency
- Dependency Configuration
- Production Environment

04

Software testing



Software Testing

—Unit Testing

For 'CoachController' class:

- ✓ testViewCoachInfo()
- ✓ testDeleteCoachInfo()
- ✓ testAddCoachInfo()
- ✓ testUpdateCoachInfo()
- ✓ testGetCoachByName()

Similar unit tests have been performed for:

- | | |
|---------------------------|---------------------------|
| ❑ AdminLoginController | ❑ SportActivityController |
| ❑ AdminUserInfoController | ❑ StadiumController |
| ❑ BookingController | ❑ UserController |
| ❑ CoachController | ❑ UserInfoController |
| ❑ CouponController | ❑ UserLoginController |
| ❑ ReportController | ❑ WalletController |

```
✓ CoachControllerTest (com.group6.bo. 772 ms)
  ✓ testGetCoachByName() 772 ms
    Tests passed: 1 of 1 test - 772 ms
    Creating a new SqlSession
    SqlSession [org.apache.ibatis.session.defaults.DefaultSqlSession@52bcd7c9] was not registered for synchronization because synchronization is not active
    JDBC Connection [HikariProxyConnection@399902409 wrapping com.mysql.cj.jdbc.ConnectionImpl@7f6b57f2] will not be managed by Spring
    ==> Preparing: SELECT * FROM coach_info WHERE name = ?
    ==> Parameters: Mingyuan.Li(String)
    <== Columns: id, activity, contact, end_time, name, start_time, status
    <== Row: 1, shuttlecock, 181298390, 17:12:00, Mingyuan.Li, 09:00:00, IDLE
    <== Total: 1
    Closing non transactional SqlSession [org.apache.ibatis.session.defaults.DefaultSqlSession@52bcd7c9]
    2024-05-11T15:02:00.240+08:00 INFO 87373 --- [booking4sportcentre] [ionShutdownHook] j.LocalContainerEntityManagerFactoryBean : Closing JPA EntityManagerFactory for pe
    2024-05-11T15:02:00.244+08:00 INFO 87373 --- [booking4sportcentre] [ionShutdownHook] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Shutdown initiated...
    2024-05-11T15:02:00.252+08:00 INFO 87373 --- [booking4sportcentre] [ionShutdownHook] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Shutdown completed.
    Process finished with exit code 0
```


Software Testing

— — Integration Testing

AdminUserInfoControllerTest:

Add, verify, modify, re-verify, and delete a user.

BookingControllerTest:

Create booking, confirm creation, update details, confirm booking, verify balance adjustment, delete booking.

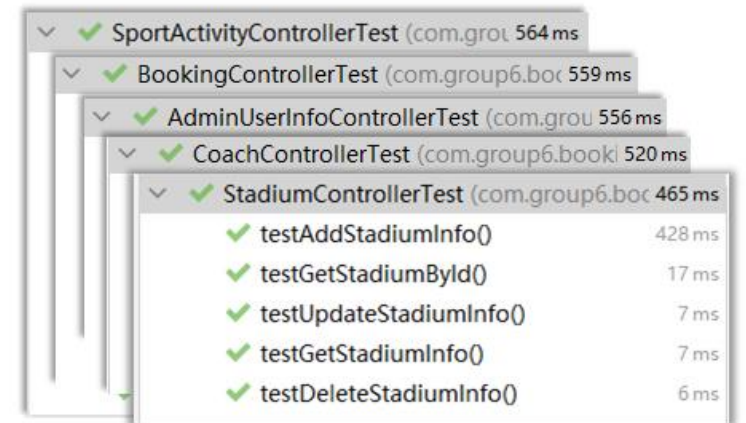
SportActivityControllerTest:

Add coach and stadium, create activity, verify activity addition, modify ticket count, re-verify activity, update details, delete activity.

CoachController and StadiumController:

Add, delete, modify, and check operations.

The testing results showed that our controllers passed all testing correctly.



✓ SportActivityControllerTest (com.gro	564 ms
✓ BookingControllerTest (com.group6.boc	559 ms
✓ AdminUserInfoControllerTest (com.grou	556 ms
✓ CoachControllerTest (com.group6.bookl	520 ms
✓ StadiumControllerTest (com.group6.boc	465 ms
✓ testAddStadiumInfo()	428 ms
✓ testGetStadiumById()	17 ms
✓ testUpdateStadiumInfo()	7 ms
✓ testGetStadiumInfo()	7 ms
✓ testDeleteStadiumInfo()	6 ms

Software Testing

—System Testing

Result:

Test Category	Functional Integrity	Performance	Security	Compatibility
Results	All functions performed as expected	Response time within limits	All security principles are valid	Compatible with all tested browsers

1.Functional Integrity: Validates all user-facing functions in both user and administrator systems.

2.Performance: Assesses front-end and back-end response time.

3.Security: Focuses on login security measures.

4.Compatibility: Software are tested on multiple browsers (Edge, Safari, Firefox).

Software Testing

—Acceptance Testing

Test Result of Four Real Users:

Test Category	Total Tests	Tests Passed	Tests Failed	Pass Rate	Test Coverage
Results	31	31	0	100%	100%

1. Administrator Operations:

Including login, logout, adding, modifying, and deleting entries in the system's master files, as well as viewing orders, user information, and statistical reports.

2. User Operations:

Including registration, login, logout, updating personal information, booking activities, and updating bookings.

05

Conclusion and Future Work



Conclusion and Future Work

CONCLUSION

- ⑩ **Designed and Implemented** Online Booking System
- ⑩ **Adhered to Software Engineering Standards**
- ⑩ **Covered Requirements, Design, Testing, Deployment**

FUTURE WORK

- ⑩ Shopping cart **centralized payment function**
- ⑩ **Optimize** the performance of the backend

06

Demonstration



THANK YOU

Group 6

