

# **CPT202 Software Engineering Group Project**

## **Project Description**

### **Introduction**

Software Engineering Group Project is a project-based module without a final exam. You will work in a team of between seven and nine members to produce a moderately sized working software system. Your team will be given a vague specification and is expected to deliver a software, a presentation, and 3 reports as stated in the Assignment Papers, which meets the specifications before the due date. The size and type of the project are suitable for development in a modular format, and so teams are encouraged to devise program structures that allow various parts of the code to be developed independently by each team member. Being a team player means you are expected not only to apply the knowledge gained throughout the past semesters to specify, design, implement, test, and document your own code but also to cooperate with your teammates so that the whole project will be delivered on time with good quality.

### **Grouping**

Each team must be consisting of seven to nine members. Teams will be formed in two stages: Firstly, you will be given the option to choose your own team members. Students failing to form a team will then be randomly assigned to a team. Each team will then be randomly assigned to one of the 3 projects presented below. Students wishing to form their own team should register their team members in the Learning Mall.

### **General Guidelines**

The project descriptions are deliberately given in the form of simple customer specifications, which (as in the real world) are incomplete and often ambiguous, rather than a set of exact functional specifications. The team members should work methodically together (as the developers in a real-world software project would) to:

1. Analyse and formalize the customer specifications (at this stage, the various design choices and the software features can be subject to the team's creativity).
2. Design and decompose the functional and programmatic aspects of the problem and allocate constituent tasks to each team member. You are expected to use a top-down design which can then be modularised so that the tasks for each member can be clearly determined.
3. Implement the product with frequent meetings to report progress and decisions to each other and re-evaluate the agreed courses of action.
4. Implement test procedures, debug and correct the program. Each program module should be independently testable. Testing of each module and the program as a whole should be performed.
5. Finalise the deliverables.

The specifications are only basic, and most of the design choices should be made in your team meetings. The systems described within the different projects have a variety of different features, and the disambiguation of the customer specifications can be based on your logic and real-life experience.

If the team cannot implement all of the system features mentioned, it is better to have a few features fully working without run-time crashes than none of the required features working properly due to bugs or disrupting ripple effects between modules in the project. However, the corresponding marks deduction will be applied depending on the missing features.

If any team issues arise during the project, it is important to raise these with the module teaching staff as soon as they do so that resolutions can be found quickly. It is important to work with your team and to share the work accordingly. All team members must be responsible for carrying out the full development cycle (requirement study, UI, programming, database, testing, as well as report writing.) of the assigned modules.

**Deliverables**

See Assignment Papers for details.

## **Project A: Daily Reading Tracker**

(Supervisor: Chengtao Ji([Chengtao.Ji@xjtlu.edu.cn](mailto:Chengtao.Ji@xjtlu.edu.cn)))

### **TA & Groups:**

**Yifan Guan**([Yifan.Guan17@student.xjtlu.edu.cn](mailto:Yifan.Guan17@student.xjtlu.edu.cn)): Group(20, 36, 26, 29, 41)

**Yejuan Xie**([Yejuan.Xie21@student.xjtlu.edu.cn](mailto:Yejuan.Xie21@student.xjtlu.edu.cn)): Group(40, 38, 8, 15)

**Xiaolong Wu**([Xiaolong.wu24@student.xjtlu.edu.cn](mailto:Xiaolong.wu24@student.xjtlu.edu.cn)): Group(30, 28, 37, 13))

### **Overall Description:**

A web-based application designed to help users record and manage their daily reading activities. The application will feature user authentication, role-based access control, and administrative oversight to ensure a smooth and secure user experience.

### **Customer Specifications:**

The system to be implemented should provide the following functionalities:

#### **User Registration and Verification:**

- Allow users to register with an avatar, email, and password.
- Implement email verification for account activation.

#### **Role-Based Access Control:**

- Define two user roles: Admin and User.
- Admin can be pre-defined.

#### **Reading Activity Management:**

- Allow users to log their daily reading activities, including the title of the book/article, author, date, time spent reading, and personal notes.
- Enable users to edit or delete their logged entries.

#### **Resource Review and Management:**

- Admins can review user-submitted reading logs and manage inappropriate content.
- Admins can lock/unlock user accounts if excessive violations occur.

#### **Resource Listing:**

- The system lists all logged reading activities with options to view details.
- Users can view their reading history and track their reading progress over time.

#### **Search and Filter:**

- Implement search functionality for users to find specific reading logs.
- Provide filtering options based on date, title, author, and time spent reading.

### **System Users:**

The system should provide functionality for different end users listed below:

**System Administrators should be able to:**

- Review and manage user-submitted reading logs.
- Lock and unlock user accounts based on content violations.
- View and manage all user accounts and reading activities.

**Users who should be able to:**

- Register as a new user with email verification.
- Login to the system.
- Update their personal information, upload an avatar, and change passwords.
- Log daily reading activities, including editing or deleting entries.
- Search and filter their reading logs.
- View their reading history and progress.

**Additional Considerations:**

- **Security:** Ensure secure data handling and protection against vulnerabilities.
- **Usability:** Design an intuitive and accessible user interface. Consider usability properties such as providing help for users, accepting user feedback, and optimizing app performance.

## **Project B: Music Resource Management**

(Supervisor: Soon Phei Tin([Soon.Tin@xjtlu.edu.cn](mailto:Soon.Tin@xjtlu.edu.cn)))

### **TA & Groups:**

**Yang Xiang**([Yang.Xiang19@student.xjtlu.edu.cn](mailto:Yang.Xiang19@student.xjtlu.edu.cn)): Group(7, 21, 24, 11, 6)

**Sisi Shen**([Sisi.Shen24@xjtlu.edu.cn](mailto:Sisi.Shen24@xjtlu.edu.cn)): Group(14, 2, 12, 45)

**Haosu Xu**([Haosu.Xu23@student.xjtlu.edu.cn](mailto:Haosu.Xu23@student.xjtlu.edu.cn)): Group(19, 42, 31, 18))

### **Overall Description:**

A web-based application for users to manage personal music resources, featuring user authentication, role-based access control, and administrative oversight.

### **Customer Specifications:**

The system to be implemented should provide the following functionalities:

#### **User Registration and Verification:**

- Allow users to register with an avatar, email, and password.
- Implement email verification for account activation.

#### **Role-Based Access Control:**

- Define two user roles: Admin and User.
- Admin can be pre-defined.

#### **Music Resource Management:**

- Enable users to create, read, update, and delete (CRUD) music categories.
- Allow users to upload music files (mp3) with metadata (title, artist, album, etc.).
- Enable tagging and categorization of music files.

#### **Resource Review and Management:**

- Admins review user-uploaded resources and block unauthorized/prohibited content.
- Admins can lock/unlock user accounts if content is blocked excessively.

#### **Resource Listing:**

- System lists all music resources with options to view details.
- Users can play music files on the details page.

#### **Search and Filter:**

- Implement search functionality for users to find music resources.
- Provide filtering options based on categories and tags.

### **System Users:**

The system should provide functionality for different end users listed below:

**System Administrators** should be able to:

- Review and block user-uploaded music resources.
- Lock and unlock user accounts based on content violations.
- View and manage all music resources and user accounts.

**Users** who should be able to:

- Register as a new user with email verification.
- Login to the system.
- Update their personal information, upload avatars, and change passwords.
- Create and manage personal music categories.
- Upload, tag, and categorize music files.
- Search and filter music resources.
- View resource listing and detailed information about their music resources.
- Plays music.

### **Additional Considerations:**

- **Security:** Ensure secure data handling and protection against vulnerabilities.
- **Usability:** Design an intuitive and accessible user interface. Consider usability properties such as providing help for users, accepting user feedback, and optimizing app performance.

## **Project C: An online meeting booking system**

(Supervisor: Nanlin Jin([Nanlin.Jin@xjtlu.edu.cn](mailto:Nanlin.Jin@xjtlu.edu.cn)))

### **TA & Groups:**

**Dingding Chen**([Dingding.Chen18@xjtlu.edu.cn](mailto:Dingding.Chen18@xjtlu.edu.cn)): Group(1, 16, 33, 50, 25)

**Xinyi Zeng**([Xinyi.Zeng24@student.xjtlu.edu.cn](mailto:Xinyi.Zeng24@student.xjtlu.edu.cn)): Group(22, 5, 10, 49)

**Tong Zou**([Tong.Zou24@student.xjtlu.edu.cn](mailto:Tong.Zou24@student.xjtlu.edu.cn)): Group(32, 17, 9, 23))

### **Overall Description:**

A web-based application for users to manage the meeting room resources, featuring user authentication, role-based access control, and administrative oversight.

### **Customer Specifications:**

The system to be implemented should provide the following functionalities:

#### **User Registration and Verification:**

- Allow users (students who have a university ID card number) to register with an avatar, email, and password.
- Implement email verification for account activation.

#### **Role-Based Access Control:**

- Define two user roles: Admin and User.
- Admin can be pre-defined.

#### **Meeting room management:**

- Allow admins to enter the catalog of the meeting rooms, including an image of the meeting room, the room number, the capacity (number of seats), any IT facilities such as interactive screen/projector and speaker(s), and the location (for example in SD building, EE building).
- Admins to maintain the updated information about bookings, including time slots, dates, the reasons for booking, the account who has booked meeting rooms, and the room availability.

#### **Resource Review and Management:**

- Admins can review the updated information about bookings and the admins can reject any booking.
- Admins can lock/unlock user accounts if they book the meeting rooms excessively.

### **Resource Listing:**

- The system lists all available meeting rooms with options for viewing details.
- Users can view the details of the available meeting rooms and the bookings they have made.

### **Search and Filter:**

- Implement search functionality for users to find meeting rooms.
- Provide filtering options based on capacity, location and/or IT facilities.

### **System Users:**

The system should provide functionality for different end users listed below:

**System Administrators** should be able to:

- Review, add, delete meeting rooms.
- Lock and unlock user accounts based on overbooking.
- View and manage all meeting room bookings, and user accounts.

**Users** who should be able to:

- Register as a new user with email verification.
- Login to the system.
- Update their personal information, upload avatars, and change passwords.
- Manage their room booking, including deleting or updating an existing booking, before the start of the meeting time.
- Search meeting rooms on the capacity, **IT facilities**, and/or the location.
- Make a new room booking.
- View the booking they have made.

### **Additional Considerations:**

- **Security:** Ensure secure data handling and protection against vulnerabilities.
- **Usability:** Design an intuitive and accessible user interface. Consider usability properties such as providing help for users, **accepting user feedback**, and optimizing app performance.