

Part 0: We are planning to use Java. Some of the members use macOS and some others use Windows. We have talked to Prof. Kaiser and she said she was fine with it. For API, we will use JDBC for communication with database, OAuth 2.0 for login, and potentially sphinx4, a voice recognition API.

Schedule++

Advanced Software Engineering Project Proposal

Team:Scoreoverflow

Cindy Le(xl2738), Henry Xing(hx2209), David Wan(dw2735), Zichuan Wang(zw2395)

Synopsis:

Schedule++ is a multi-platform desktop software that automates the scheduling of public facility usages (e.g., university classrooms). Administrators can input a list of events (e.g. class sessions) with their respective time and location requirements. The system will assign a room and time slot for each of them while optimizing for these constraints. Individual users can also view available rooms and book venues for private events on this platform. Additional features include specifying the equipments (e.g. projector) available in a classroom and prioritizing certain user groups (e.g. professors) for room booking. The software can also potentially add a voice control function for accessibility concerns. The system will be a smart and integrated tool for universities and other facility providers.

Target Users:

- Educational institutions (e.g., universities and high schools)
- Middle to large companies with more than several meeting rooms and people to use them
- Public event space providers or sports stadiums with available courts

User Categories:

- Normal users (e.g., students and normal employees): They can book individual events in available rooms
- High priority users (e.g., professors and managers): They can book events like the normal users. They also have priority booking over normal users, able to book rooms that were already taken by normal users.
- Program supervisor (e.g., member of the administration office): They are very similar to High priority users, but they can schedule events in batch.
- Administrator (e.g., Building maintenance personnel): They input the facility information including operating hours and available equipments. They can optionally shut down a facility for maintenance purposes.

Use Cases:

< 0 >: As an administrator, I want the ability to add a new room or change existing room availability so that I can be sure the information in the system is update to date. My conditions of satisfaction are I can add a new room with its availability and equipment information (e.g., projectors) to the system and I can change the availability/facility information of a room.

< 1 >: As a normal user, I want the ability to book a room or cancel my booking. My conditions of satisfaction are I can search for rooms whose availability and facility information matches my request and I can cancel my booking.

< 2 >: As a high priority user, I want the ability to book a room with high priority or cancel my booking. My conditions of satisfaction are I can search for rooms whose availability and facility information matches my request, I can cancel my booking and I can book a room even if it has been booked by a normal user (under certain constraints).

< 3 >: As a program supervisor, I want the ability to schedule meeting places and venues in batch. My conditions of satisfaction are I can input a list of meeting requests with specified time/location/equipment/capacity preferences and schedule all of them without pairwise conflict and satisfy the most requirements.

< 4 >: (Potentially) As a user with disability, I want the ability to book rooms with speech input. My condition of satisfaction is I can get get information and book meeting places with voice control.

< 5 >: (Potentially) As a user, I want the ability to see booked rooms and contact him/her to negotiate. My condition of satisfaction is I can get contact information of the person who booked my desired room.

Workflows/ Usage Scenarios:

Case users:

1. Users sign up with uni/employee ID
2. Select room size, priority, and other preferences
3. Get lists of rooms
 - a. normal users can only see unbooked rooms
 - b. high priority users can see both unbooked rooms and rooms booked by normal users
 - c. Same user groups can see each others' booked rooms for negotiation
4. Select the room the user wants to book and confirm the time period and location

5. Voice input for differently abled people (Optional)
6. If a high priority user overwrites the room (book a room that a normal user has booked), it will automatically send a email to notify the normal user

Case Administrator:

1. Sign up with uni/employee ID
2. Input new classroom information and available technologies
3. Change existing classroom information and available technologies as necessary

Case program supervisor:

1. Users sign up with uni/employee ID
2. Import a list of events with associated room size, priority, and other preferences to request rooms
3. Get the room assignments for these events, automatically scheduled by the system, notify any normal users if a room is overwritten
4. Voice input for differently abled people (Optional)

Technologies

Development Framework: Java

Static Analysis tool: PMD and IDEs (Eclipse, IntelliJ)

Unit testing tool: Junit

Build tool: Maven

Data Storage: MySQL, jOOQ/JDBC, (potentially) online storage platform: OCI

Continuous integration tool: Travis CI

API: JDBC, OAuth 2.0, sphinx4 (Optional)