

Meeting Room Booking System

Overview

The Meeting Room Booking System is a Java application designed to manage meeting rooms and bookings within an organization. It provides functionalities for users to check room availability, book meeting rooms, and manage bookings. Administrators have additional privileges such as adding new rooms and managing user accounts.

Features

User Authentication

- Users can log in with their credentials or register as new users.
- Administrators have special privileges compared to regular users.

Room Management

- Administrators can add, edit, or delete meeting rooms.
- Each room has attributes such as room ID, capacity, and availability.

Booking Management

- Users can view available rooms for a specified date and time.
- Users can book meeting rooms for specific time slots.
- Bookings are stored in the system and checked for conflicts.

User Interface

- The application provides a command-line interface for user interaction.
- Users navigate through menus to perform various actions.

How it Works

- **Login/Register:** Users start by logging in with their credentials or registering as new users.
- **Main Menu:** After login, users are presented with a main menu where they can choose between admin or user functionalities.
- **Admin Actions:** Administrators can manage rooms, view bookings, manage users, and log out.
- **User Actions:** Regular users can view available rooms, book rooms, view their bookings, and log out.

Implementation Details

- The application is implemented in Java using object-oriented programming principles.
- It consists of multiple classes such as RoomManager, BookingManager, ConflictResolver, UserInterface, User, Room, Booking, and Main.

- User input is handled using the Scanner class for interactive command-line interaction.
- Date and time handling are done using Java's Date class.
- The project demonstrates basic user authentication, input validation, and system navigation.

Setup and Execution

- Import the project into an IDE like Eclipse or IntelliJ IDEA.
- Set up the Java build path and ensure the project compiles successfully.
- Run the Main class to start the application.
- Follow the on-screen instructions to interact with the system