CSCI 370 Project

Student-Professor Appointment Booking System (SPABS)

John Lamprecht

Proposal

What and Who:

The purpose of this project is to develop a Student-Professor Appointment Booking System (SPABS) designed for a university setting. This system aims to facilitate the scheduling of one-on-one meetings between students and professors, enhancing the academic experience by providing a convenient and efficient method for booking appointments. The target audience includes university students, professors, and academic administrators who are involved in the scheduling of academic advising, tutoring, or research meetings.

The student-Professor Appointment Booking System (SPABS) is designed for the academic domain within a university setting. It addresses the scheduling challenges faced by students, professors, and academic administrators in arranging one-on-one meetings for academic advising, tutoring, or research discussions.

The primary audience for SPABS includes university students seeking academic support, professors offering guidance and mentorship, and academic administrators responsible for facilitating academic interactions.

The problem:

The current method of scheduling one-on-one meetings between students and professors, primarily through email or on a first-come, first-serve basis during scheduled office hours, often lacks flexibility and efficiency. This approach frequently results in students being unable to meet with professors and administrators due to conflicting classes and other time-oriented factors. Additionally, students may have to book appointments at times that are not convenient for them, leading to missed appointments and reduced access to academic support. Professors, conversely, face challenges in managing their schedules and ensuring they are available for all students who need to book appointments.

This project aims to address these issues by introducing a system that allows students and professors to book appointments. The system will ensure that double bookings do not happen and the out of office hours booking process is streamlined. This innovative system not only improves the scheduling process but also enhances the overall academic experience by ensuring that students and professors can meet at mutually convenient times, thereby facilitating more effective academic advising, tutoring, and research meetings.

Primary objectives, goals, and methods:

The primary objectives of this project are to:

1. Develop a user-friendly interface for students and professors to book appointments, ensuring that double bookings and out of office hours bookings are handled and tracked.
2. Ensure secure and efficient communication between the client and server, with the server acting as a single application that communicates with an Oracle database to manage appointments.

The project will be implemented using C++ for both the client and server components, with the server and client acting as a single application that communicates with an Oracle database. The development process will involve designing the system architecture, implementing the frontend and backend logic, testing the system, and deploying it for use.

Expected outcomes and implications:

The expected outcomes of this project include:

1. A functional Student-Professor Appointment Booking System that allows students to book one-on-one meetings with professors by selecting multiple preferred time slots.
2. Improved scheduling efficiency and reduced missed appointments due to inconvenient times.
3. Enhanced academic support and advising for students by ensuring they have access to professors' expertise.

The implications of this project are significant for both the university and its students. It will contribute to a more efficient and effective scheduling process, leading to better academic outcomes. Furthermore, it will serve as a practical demonstration of C++ application development and database interaction.

Implementation

Database schema

Entities:

- Users
- Appointments

Relationships:

- A student and professor can book multiple Appointments.
- An appointment is associated with one TimeSlot.

Schema:

- Users

  userID (Primary Key) char(9)

  FirstName varchar(65)

  LastName varchar(65)

  Email varchar(75)

  Password varchar(60)

  UserType varchar(20)

- Appointment

  AppointmentID (Primary Key) number autogenerated

  bookerID (Foreign Key) char(9)

  bookeeID (Foreign Key) char(9)

  TimeSlot timestamp(6)

  AppointmentDate DATE

  Status varchar(10) (e.g., Pending, Confirmed, Cancelled)

  Length char(2)

ER diagram:

Sample data

- Users

  734552694, Kristofer, Nuth, Nuth.Kristofer@viu.ca, NE7R0936y5E, STUDENT
  646979538, Doro, Lapidus, Lapidus.Doro@viu.ca, K18MFp4DRl1, PROFESSOR

- Professor

  ProfessorID: 123789456, FirstName: Huizhu, LastName: Liu, Email: Huizhu.Liu@viu.ca

  ProfessorID: 987321654, FirstName: David, LastName: Wessels, Email: davidwessels@viu.ca

Interface design

- Entry screen

  Enter your oracle username>>
  Enter your oracle password>>
  Successfully connected to the database.
  Appointment ID trigger dropped successfully.
  Appointment sequence dropped successfully.
  Appointment table dropped successfully.
  Users table dropped successfully.
  Users table created successfully.
  Appointment table created successfully.
  First dummy user inserted successfully.
  Second dummy user inserted successfully.
  Dummy appointments inserted successfully.
  Enter your username>> 123456789
  Enter your password>>
  Logging in...

- Welcome screen

  welcome John Doe what would you like to do?

  view - view appointments

  book - book an appointment

  back - go back

  help - show help

  quit - exit the application

- View appointments screen

  Current appointments:
  Appointment ID: 1 With: Jane Smith Time: 15-FEB-24 12.00.00 PM -07:00 Date: 15-FEB-24
  Status: Confirmed length: 15

  Appointment ID: 2 With: Jane Smith Time: 22-MAY-24 02.30.00 PM -07:00 Date: 22-MAY-24
  Status: Pending length: 30

  Appointment ID: 3 With: Jane Smith Time: 10-MAR-24 04.45.00 PM -07:00 Date: 10-MAR-24
  Status: Confirmed length: 45

  confirm <ID> - confirm appointment
  cancel <ID> - cancel appointment
  back - go back
  help - show help
  quit - exit the application

- Book appointments screen

  Who would you like to book an appointment with or type 'CANCEL' to cancel>> Jane

  Enter the date of the appointment (YYYY-MM-DD) >>2024-05-22

  Enter the time of the appointment (HH:MM) or CANCEL to cancel>> 15:00

  Enter the length of the appointment (15, 30, 45, 60 mins) or CANCEL to cancel>> 60

  Creating booking...

  Booking created successfully.

  Booker: John Doe Bookee: Jane Date: 2024-05-22 Time: 15:00 Length: 60 Status: pending

- Quit screen

  COMMAND>>quit
  Quitting application...

Implementation

- The base functionality of viewing, booking, confirming, and cancelling appointments has been implemented.

- Stretch goals:
  - The functionality for an admin user who can do all this, create new users and new appointments, confirm, and cancel appointments for anyone has not been implemented.
  - The functionality to download a report of your appointments has not been implemented.

- Test data used for demonstrating:

  - Users:
    - 123456789, John, Doe, Doe.John@viu.ca, password123, STUDENT
    - 987654321, Jane, Smith, Smith.Jane@viu.ca, password456, PROFESSOR

  - Appointments:
    - 1, Jane Smith, 12:00, 2024-02-15, Confirmed, 15
    - 2, Jane Smith, 15:00, 2024-05-22, Pending, 30
    - 1, Jane smith, 16:45, 2024-03-10, Confirmed, 45

Testing

- Test cases:
  - Viewing appointments as John and Jane.
  - Booking a new appointment as John with Jane and vice versa.
  - Confirming appointments as John and Jane.
  - Canceling appointments as John and Jane.