Title: Booking Management System

The app will feature two distinct GUI interfaces. Admin users will have the ability to add, create, and edit events, while all other users will be able to browse existing events and purchase tickets. Everything will be stored within a MySQL database stored locally on our computer (Creation Script Included).

Problem Statement:

We're solving the problem of inefficient and manual event booking processes since lots of venues and organizations mostly rely on phone calls or emails to manage reservations and seating selections. That can lead to scheduling conflicts, double bookings, and lack of proper communication. Our system makes the booking process easier to manage, and more accurate.

Primary users would be admin users like a venue manager to manage the overall systems, and general users like guests viewing the events and registering for upcoming events.

This app matters since it showcases an event booking experience for organizers and attendees. It helps to reduce overhead for administration, improve scheduling accuracy, and gives a convenient way to secure events.

Core Features & Functionality:

Admin User Story:

Upon opening the app you will be introduced with the login screen. For admins specifically there is one shared account only. Once entering the account username and password correctly you will be brought to the admin screen. The admin screen will have a list of all existing event cards. Each event will have a small button on the allowing the admin to edit the description, the name, or even the date of this event. There will be a button that allows the admin to create a new event. Once clicking this button it will put the event at the bottom of the list. This event will be a default blank event which cannot be seen by the public. To make this event public the admin will need to change the name of the event, add a description and set a valid date for the event. There will only be one event per day. Similarly to editing an event, a small button on the event card allows the admin to delete this event. Upon deleting it all information stored in the database will be deleted.

General User Story:

Upon opening the app you will be introduced with the login screen. For anyone that is not an admin they can log in using their username and password. If the user has no account there will be two different buttons on this screen to help them out. One being a create account option, which will have a popup allowing them to make an account. While the other button will be a guest option. The guest option will just allow them into the app without signing in. Once out of the login screen they will be on the homepage for the users. There will be a list holding all of the event cards. In this page there will be a few different options:

- Filtering - Users can filter this list by specific dates they had in mind
- My Tickets - This will allow signed in users to see the tickets they have purchased
- View Events- Clicking on an event card will bring the user to the events page. Within this event page the user will be able to read all the information on the event. On the bottom of this page will be a seating chart.
    - Seating Chart - Here the user will be able to click on the specific seat they want, allowing them to buy that seat. If another user has already purchased that seat it will be greyed out (or something similar) to all other users.
- Checkout Button - This will bring the user to a new screen where they can purchase all tickets they added to their cart
- Sign Out - Signs the user out

Frontend:

- JavaFX
    - Will be used to format the programs front end
    - Seating/Onclick methods adding them to a cart
        - Seats will change color once taken (think of movie theater)
    - Drop Down Menu
    - For admin/event creators different menu options to create events (time only for now, where will be a stretch goal)
    - Menu for general users
    - Menu for admin users

Backend:

- MySQL DB
    - Users
    - Events
    - Seat
    - User Seat Join Table
    - Seat Event Join Table
    - Something that shows what seats have been sold for each event (unsure atm)

Web API:

- Paypal
- Calandar
- Google Maps

APIs will be used to display the event dates and possible location. The PayPal API would be used for the checkout screen.

Product BackLog:

- Database Creation Script – **Sprint 1**
    - Connecting to the DB
- Log In Screen – **Sprint 1**
    - Sign In
    - Create Account
- Account Creation – **Sprint 1**
    - Username
    - Password (simple hashing function)
- Admin Screen – **Sprint 1**
    - Creating an event
    - Viewing all events
    - Editing events
    - Delete events (no refunds)
    - Sign out button
- General Screen
    - List of events / Home Screen – **Sprint 1**
        - Scroll list - Upcoming shows first
        - Filter Functionality (Date)
        - Sign out button
    - Event page
        - Event Description
        - **Seating**
            - **All the stuff that goes with this**
        - Take me to checkout button
        - Google Calendar API???
        - Google Map API???
    - Checkout Screen
        - Steal CC numbers
        - List of seats in cart
        - For guests have a fake function to send an email
        - Paypal API????
        - Conformation screen (simple text box saying it was purchased)
    - My Ticket section

Stretch Goals:

- Different seating prices
    - Possible use of factory method
    - Price preview
- Event locations
- Personal account rewards
- Different kinds of event

Design Pattern:

- Strategy pattern will be used menu system and more
- Factory pattern for seats, and users

Team Roles & Responsibilities:

Nate C-Fossa - Admin Screen
Beau Desrosiers - Database Creation Script
Kiara Matos Luna - Log In Screen

https://github.com/BDesro/SE_Final-Project_Event-Booking-System

Timeline & Milestones:

Sprint 1: For sprint 1 we hope to have most of the GUI interfaces all set up, with basic integration with our database. As shown above in our product backlog we have highlighted who will be responsible for what during sprint 1. While we know this may be a little less than expected we wanted to make sure we will be able to accomplish everything we can in this sprint. So if we were to finish our assigned part early, we have outlined a couple other parts we can start working on. While this isn't typically how scrum would work we thought it would be slightly more manageable for us to break it down like this.

Sprint 2: In sprint two what we will hopefully accomplish is displaying the information that has been stored in our database properly. This will be accomplished with the seating chart for each event, and my ticket section. We will also create the checkout option. After sprint 1 we hope that our app will almost be like a very good outline of what it should look like, but with little to no functionality.

Risks & Challenges:

What could go wrong:

- Visually it may not look the best
- Managing the seat objects and the layout of the seats (colors, pricing, updates)
- Making sure UI components are properly bound to information from the database

The best way to reduce the risk of stuff going wrong would be proper communication on what components are being worked on or need to be improved. Test cases that go beyond easy testing would also be helpful in making sure any exception is handled (never know what the user might try to do wrong). If there is lots of duplicate code, refactoring would be the best course for readability, reducing the complexity and debugging, and makes our code more efficient when executing it.