

Scenario for Event Center Website

Actors:

1. **Admin:** Manages the system, the events, the users, and approves new users.
2. **Event Manager:** Creates events and manages Event Manager booths
3. **User:** Accesses events, purchases events, requests booth forms, and views the history of event purchases.

Use Cases:

Admin:

1. **Manage Users:** Admin A1 logs in and adds a new event manager, M1, to the system. A1 also reviews and approves a new user, U1, who has registered.
2. **Manage Events:** Admin A1 manages event E1.
3. **View System Statistics:** A1 views current user details, event details and can delete users or event.

Event Manager:

1. **Create Event:** Event Manager M1 logs in and creates a new event, E1, and assigns an Event Manager booth request form, B1, to event E1.
2. **Manage Event Manager Booth:** M1 manages Event Manager booth request B1 by approving or denying it for event E1.

User:

1. **Search for Events and Purchase an Event:** User U1 logs in, searches for an event, E1, and purchases it from Event Manager M1 based on M1's availability.
2. **Ticket Purchase History:** After the purchase, U1 can check his purchase ticket history

Scenario Flow:

1. **Admin A1 Logs In:**
 - Adds Event Manager M1 to the system.
 - Approves new user U1's registration.
 - Views statistics showing one Event Manager and one User.
2. **Event Manager M1 Logs In:**
 - Creates an event E1 and booth B1.
 - Updates the user booth request for events.
3. **User U1 Logs In:**
 - Views and accesses event E1.
 - Purchases an event from Event Manager M1.
 - Completes the purchase and views purchase history.
4. **Admin A1 Checks Purchase Event:**
 - Sees U1 purchase history.
 - Reviews event details.
5. **Data Storage and Retrieval:**
 - All actions by A1, M1, and U1 are recorded in the system's database.
 - Each login, event creation, event purchase, booth request, and ticket purchase history are stored and can be retrieved for review.

Implementation:

- **Admin Component:** Allows the admin to manage users, events, and event managers, and to view statistics.
- **Event Manager Component:** Enables event managers to create events/booths and manage booth requests.
- **User Component:** Allows users to view events, purchase events, request booths, and view ticket purchase history.
- **Backend Integration:** Ensures the Spring Boot, Maven with PostgreSQL application is properly communicating with the backend (APIs for user management, Event Manager management, purchase, booth request, feedback submissions).
- **State Management:** Uses Redux to manage the state related to users, events, and feedback.
- **Styling:** Utilizes styled components in a clean and modular fashion.
- **Frontend :** Used React with tailwind css