1. Project Recap

For our project we are working on a ticket management system. There will be halves of our project. One half that will be for admins of the application which will be for event creation, and the other half for users. The users will be able to buy tickets to their event. Within our first sprint cycle no major changes have happened to the idea of our project. Fundamentally it will work the same as before.

2. User Stories

Story 1:

As an admin, I want to create an event for a music concert. The admin will access the admin screen, click a "Create Event" button, and a default event will be generated. The admin can then provide a name, description, and date. The event can be saved without being published until it's ready.

Priority: High in Sprint 1

Time Estimate: 7 days

Assignee: Nate

1. Add button to make an event
2. Edit the attributes of the event
3. Add button to make event public
4. Connect backend properly
5. Validate backend database insertion

Story 2: As a user, I want to log in and see a list of available events so that I can select one and purchase tickets for it. After typing in my username and password, I want to be able to press “Log in” and be immediately brought to the general events screen so I can select my event.

Priority: low-medium for sprint 1

Time Estimate: 1-2 Days

Assignee: Beau (following login by Kiara)

Breakdown:

1. Create general screen controller/fxml.
2. Integrate events from the database with JavaFX’s TableView
3. Integrate the switch from login screen to general screen on successful user login.

Story 3:

As a user, I want the choice to log in as a guest, create an account through a separate screen, or, as an admin, to directly access the admin screen so that I can choose how to access the system based on my role.

Priority: High in Sprint 1

Time Estimate: 3-5 days

Assignee: Kiara (goes to related Screens by Nate and Beau)

Task Breakdown:

1. (Day 1) Create User class to have new User instantiations when an account is created and to keep track of user info for future integration

2. (Day 2-3) Create a LoginPage with buttons to log in as general user, create an account (separate screen), text fields to submit username and password, and message label to be displayed when incorrect username or password is added
    a. Create relevant methods for checking password using PasswordUtils (Beau)
3. (Day 4) Work started on CreateAccountScreen class with JDBC, message for success or failure, and verifying each email has ONE unique account
4. (Day 5) Refactor with scene switches to go to appropriate screens when button is clicked

3. Requirements

Identify the functional and non-functional requirements for Sprint 1

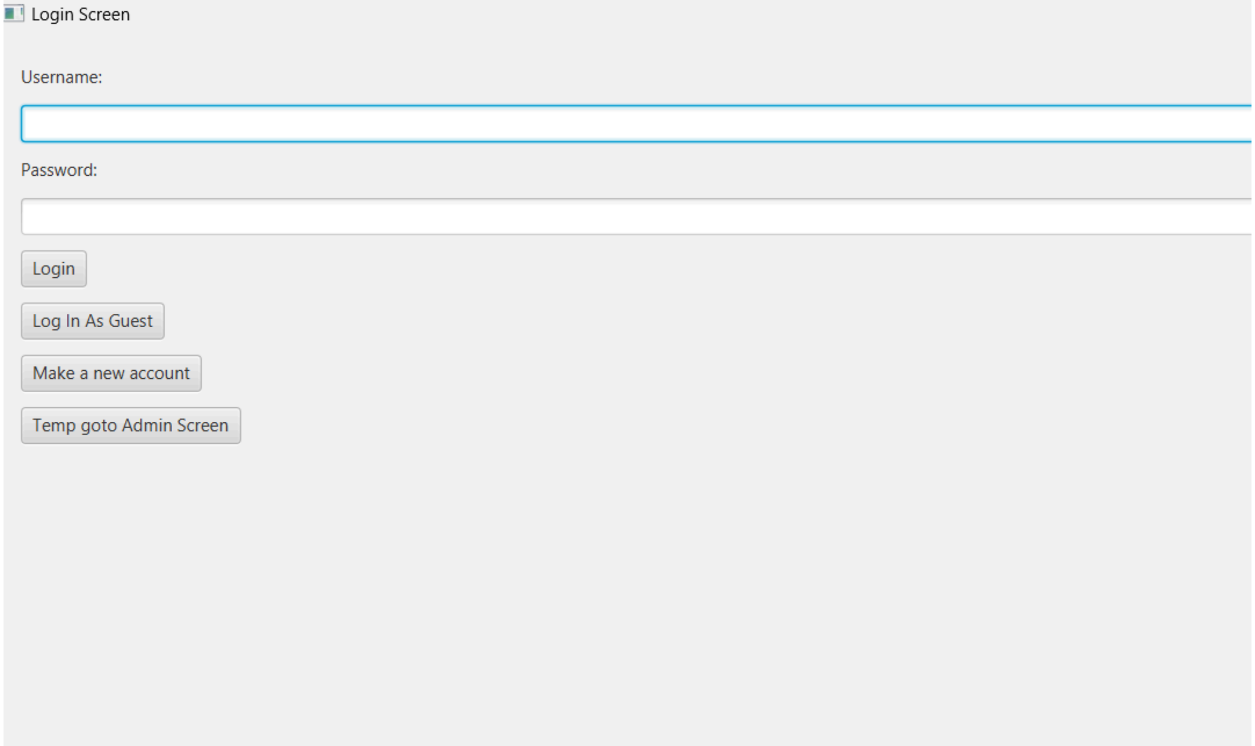
Functional

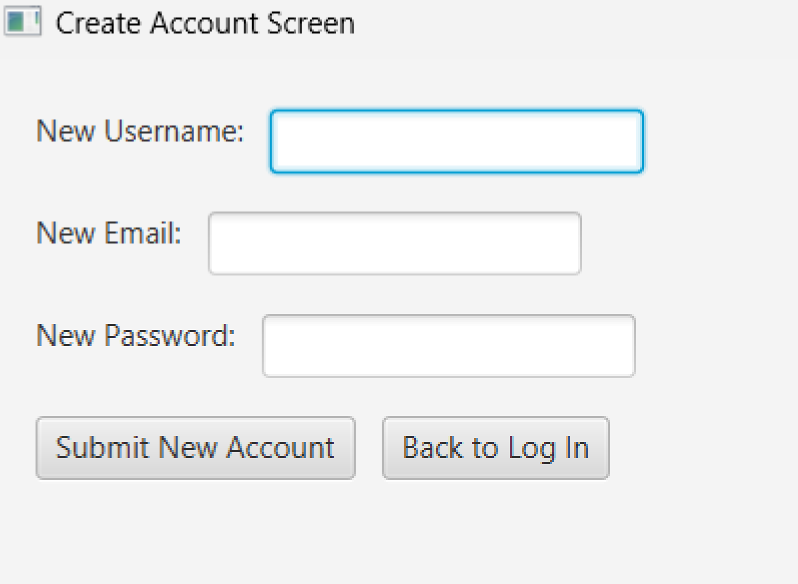
- The system must allow users to register and log in.
- There must be two different roles: Admin and User
- Admin must be able to perform basic CRUD operations on events in the database.
- Users must be able to login and see a list of events made public by the admin.

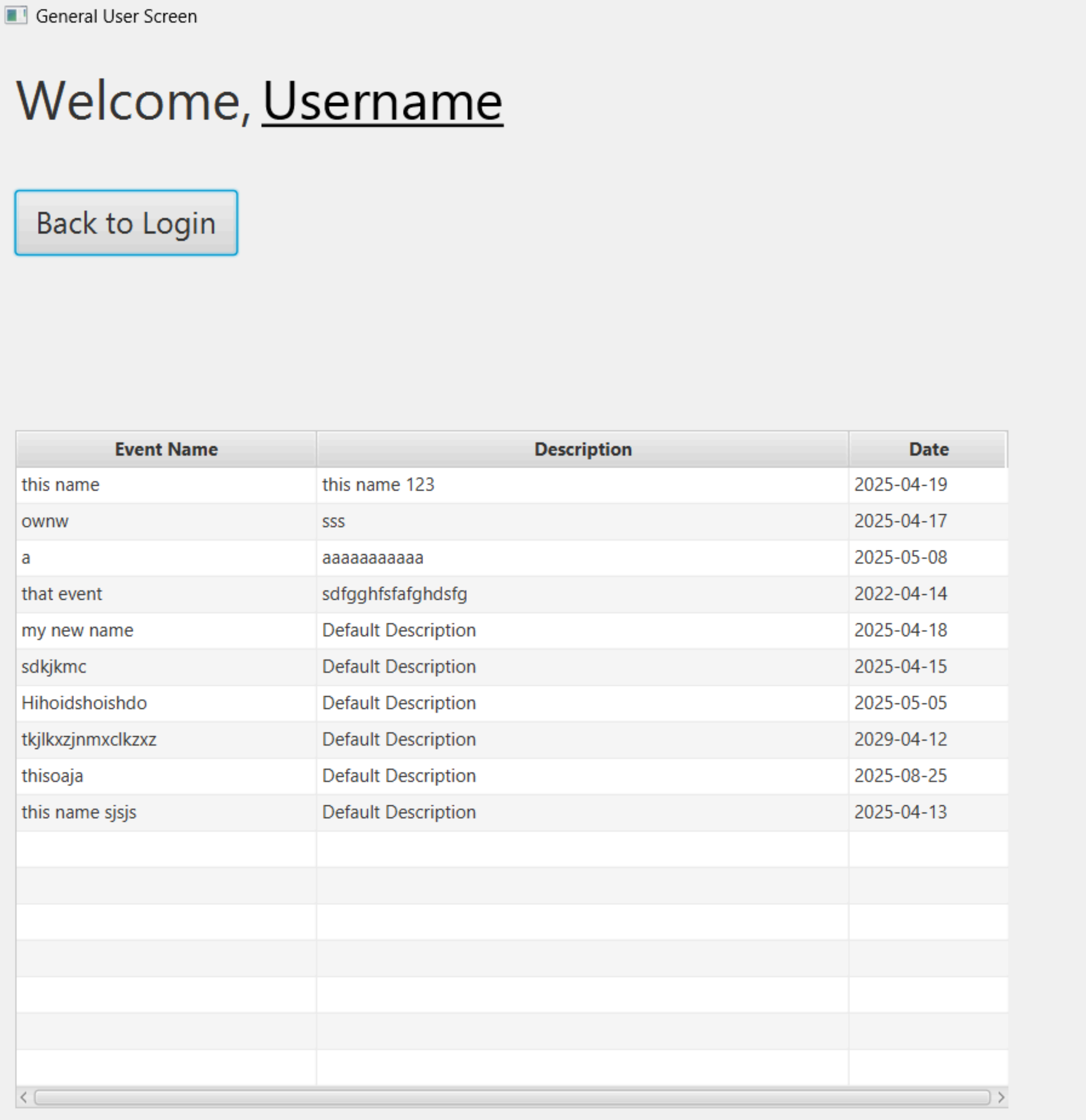
Non-Functional

- Passwords must be hashed during database insertion and authentication.
- The interface should be user-friendly and accessible
- The system should allow for relatively simple updates/debugging

4. Wireframes / UI Mockups

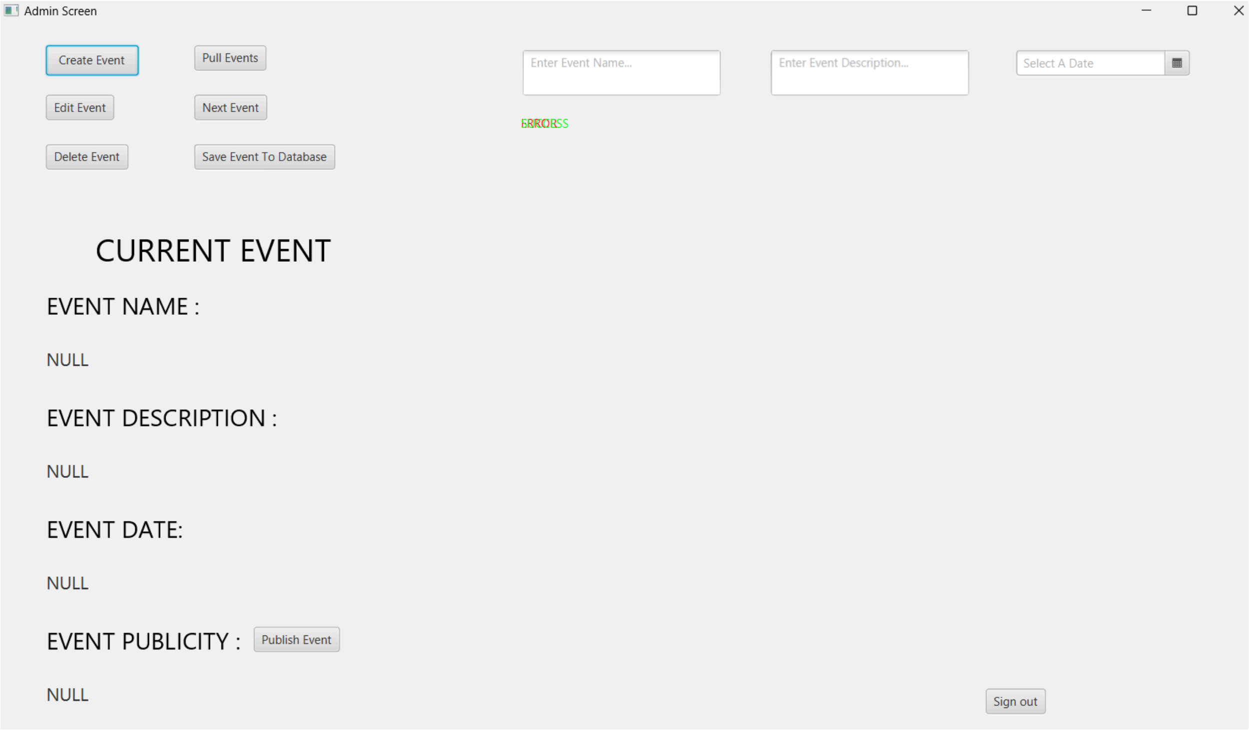

Login Screen
Username:
Password:
Login
Log In As Guest
Make a new account
Temp goto Admin Screen


Create Account Screen
New Username:
New Email:
New Password:
Submit New Account
Back to Log In



Welcome, Username

Event Name	Description	Date
this name	this name 123	2025-04-19
ownw	sss	2025-04-17
a	aaaaaaaaaaa	2025-05-08
that event	sdfgghfsfafghdsfg	2022-04-14
my new name	Default Description	2025-04-18
sdkjkmc	Default Description	2025-04-15
Hihoidshoishdo	Default Description	2025-05-05
tkjlkxzjnmxclkzxz	Default Description	2029-04-12
thisoaja	Default Description	2025-08-25
this name sjsjs	Default Description	2025-04-13



CURRENT EVENT

EVENT NAME :

NULL

EVENT DESCRIPTION :

NULL

EVENT DATE:

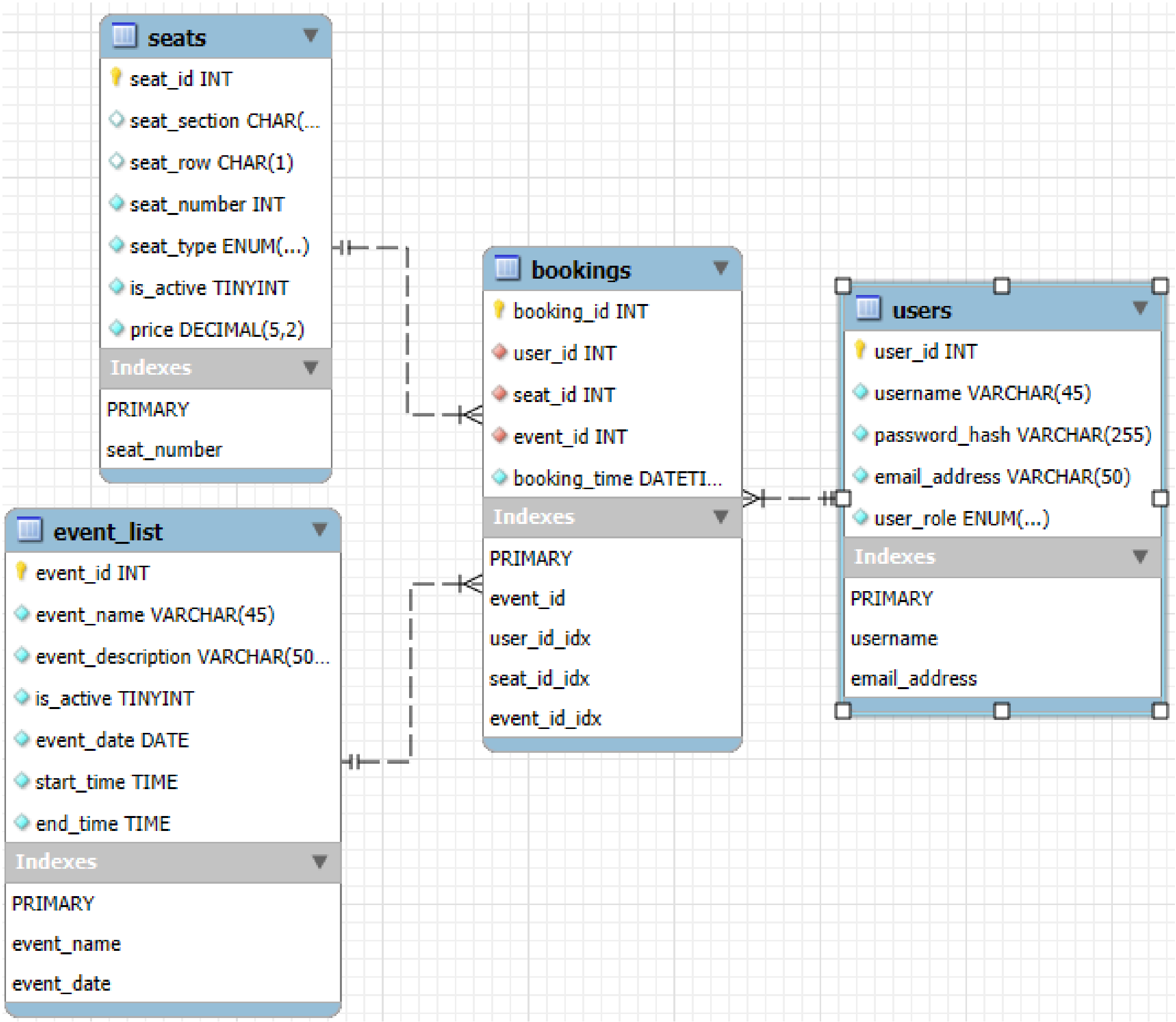
NULL

EVENT PUBLICITY :

NULL

4. System Architecture Diagram

Create a simple diagram showing how your frontend, backend, and (optional) web API connect. You can use tools like Canva, Lucidchart, or Draw.io.



Simple Diagram Booking System.pdf

https://drive.google.com/file/d/1NCQc0XYgfsZMhQfXDQr0Uux3HwBWAM5j/view?usp=sharing (Simple Diagram Booking System)

5. Development Progress

Through Sprint 1 we have accomplished most of our original goals. We have a working version of everything we set out to make. Primarily what is left is just debugging and making sure there are no errors, and polishing. Within our product backlog we wanted to work on getting the entire admin screen working, a login screen which will allow users to create an account if they do not have one or use a guest account, and finally a very basic screen for when a user signs in. We have accomplished all of these tasks, as well as creating our Database and having the database integration.

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

6. Team Collaboration

What has each team member worked on during Sprint 1? How are you communicating (e.g., Discord, email, etc)? Are you using any kind of additional coordination tools?

Kiara:

- **Login Page**
- **Create Account Screen**
- **JDBC class**
- **User class**
- **Working on seat and main seating screens**

Nate:

- **Event Class**
- **Admin Screen and Functionality**

Beau:

- **Database creation script**
- **Main application**
- **SceneManager class for centralized scene control**
- **Class for password hashing/authentication**
- **Working on general user screen controller and fxml**

We are using discord to communicate and plan out Scrum meetings.

7. Challenges Encountered

What technical or team challenges have you faced? How did you address or plan to resolve them?

One challenge we faced was within our initial design we wanted to have a scrollable list of created events, and when clicking on a specific event it would bring up a new screen that would allow us to interact with that specific event. Making an interface that works in this way proved to be very complicated for us as we are all learning how to use JavaFX. So in the meantime we have two different implementations of showing all the events. In the Admin screen we have the attributes of one selected event shown and you can cycle through them all as they are stored within an observable list. Whereas in the User screen they are shown inside of a TableView node. Depending on how well the TableView works we may switch the Admin screen to function this way as well.

We also set up a time to meet to discuss how to work with Github Desktop which didn’t take long to figure out at first. Making the configurations and setting up the SDK and libraries we needed did take some time to work with but we eventually got it down in the end. We addressed any issues by meeting face-to-face and looking over what needs to be done or changed.