

Open-Deck

**A Creative
Opportunity
For
Future Artists
And
Local Talent**

N11753579
Batbuyan Batjargal

1. Executive Summary

Open Decks is a web application designed to connect local DJs, independent musicians, and fans. It serves as a centralized platform for discovering, hosting, and booking small-scale music events like DJ nights, indie gigs, and community jam sessions. Users can create events, book tickets, and engage through comments, fostering a vibrant local music community.

2. Description of Site

Open Decks is a music focused event management platform designed for musicians and DJs to create, manage, and discover collaborative gigs. Unlike traditional event platforms that focus on mainstream concerts or ticketed events, Open Decks emphasizes smaller, community driven performances such as open-mic nights, jam sessions, and intimate gigs. The platform allows musicians to list events, match with other performers based on genre or instrument, and promote their shows to a dedicated audience of music enthusiasts. Additionally, Open Decks can create local open decks for up and coming artists to go perform on by signing up, allowing artists to sign up for specific performances.

Event hosts can also search for performers with specific sounds or styles, ensuring they can curate lineups that perfectly match the vibe of their events.

3. Description of Technology in Context

3.1. Web Application

A web app offers broad accessibility, as it can be accessed from any device with a browser and an internet connection, making it ideal for both musicians and event hosts. Updates are seamless and do not require user intervention (Zhao et al., 2022). However, web apps often lack offline functionality and deep integration with device features, such as push notifications or location-based services, which could limit user engagement.

3.2. Native Mobile Application (iOS/Android)

A native mobile app would allow **deeper engagement** with features like push notifications for event reminders, GPS integration for local gigs, and offline sign-up capabilities. This platform also enables smoother performance and a more polished user experience (Sahin & Akbulut, 2020). However, mobile apps require more resources to develop and maintain, especially if building separate apps for iOS and Android, and updates need to pass through app store approval processes.

3.3. Native Desktop Application (Windows/macOS)

A desktop app could cater to event hosts or venue managers who need to manage multiple gigs, schedules, and performer lineups from a workstation. It allows for **robust management features** and better integration with professional tools like spreadsheets or email clients. On the

downside, desktop apps have limited reach among casual users and lack portability, which makes them less appealing for musicians constantly on the move.

3.4. Open Deck Platform Comparison

For Open Decks the platform should be based on Web Application and its companion Mobile App. The web platform can serve as the main hub for event discovery and management, while the mobile app can act as a simplified process Desktop applications, while useful for hosts with complex needs, would serve a more niche audience and could be better suitable for Hosts and More Popular Creators on the Platform.

Other platforms like progressive web apps are a great consideration, blending the accessibility of web apps with some native features like offline caching and notifications (Google Developers, 2023). This could reduce development costs while still delivering a mobile-friendly experience.

4. User Story

4.1. View Event Details: As a user, I want to view the details of a specific event so that I can decide whether to attend or sign up to perform

4.1.1. The event detail page includes event description, location map, available performance slots, and performer list.

4.1.2. Ticket availability and price are clearly shown.

4.1.3. Comments and ratings for the event are visible to all users.

4.2. Register a Creator: As a musician or host, I want to register an account with my name, email, password, and phone number so that I can create events, sign up for performance slots, or book tickets.

4.2.1. Registration form collects name, email, phone number, and password securely in hash protected into a Database

4.2.2. Validation ensures no duplicate emails.

4.2.3. Successful registration triggers a confirmation.

4.3. View Upcoming Events: As a user, I want to view a list of upcoming Open Decks events and their locations so that I can find events to attend or perform at

4.3.1. A page displays a list of upcoming events with details such as date, time, location, and event type.

4.3.2. Events are sorted by date by default.

4.3.3. Users can filter events by city, genre, or date.

4.3.4.

4.4. Signup For Performance slots: As a musician, I want to sign up for an available slot at an event so that I can perform in front of a live audience.

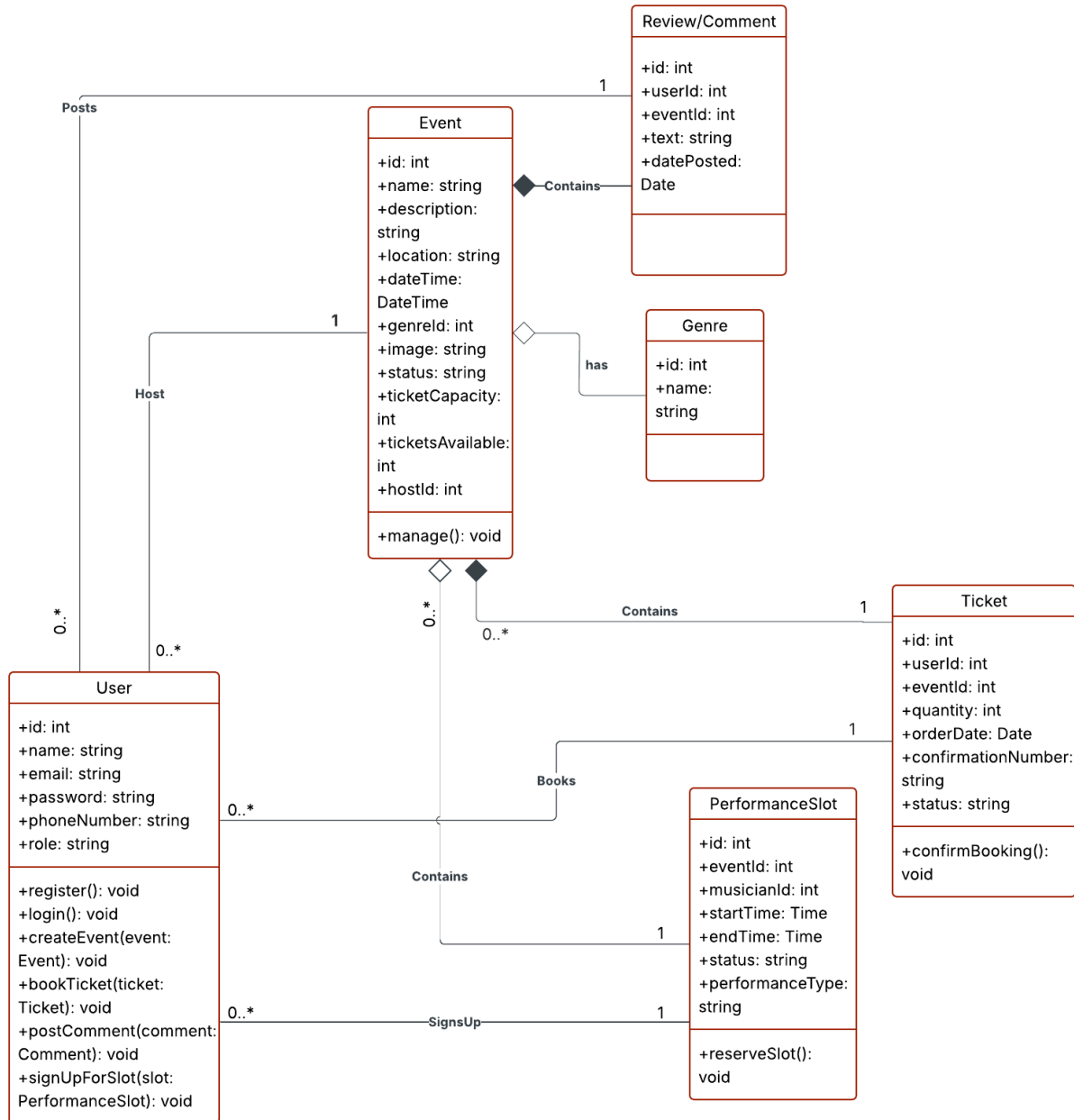
4.4.1. The event page displays available slots with times.

4.4.2. A slot can not be double booked.

4.4.3. Confirmation is shown after a successful signup

- 4.5. Buying Ticket:** As a user, I want to book tickets for an event so that I can attend and support my favorite artists.
 - 4.5.1. Logged-in users can select the number of tickets to book.
 - 4.5.2. The system prevents booking more tickets than are available.
 - 4.5.3. Confirmation with a booking ID is displayed after purchase.
 - 4.5.4. Event status automatically updates to “Sold Out” when tickets reach zero
- 4.6. Comment on Events:** As a logged-in user, I want to post comments on an event so that I can share feedback or interact with other attendees.
 - 4.6.1. Comment form is visible only to logged-in users.
 - 4.6.2. Comments display the user’s name and date posted.
 - 4.6.3. Comments are visible to all users including guest visitors.
- 4.7. Update Events:** As an event host, I want to edit details of my events so that attendees and musicians have the latest information.
 - 4.7.1. Hosts can only edit events that they created
 - 4.7.2. Fields such as description, image, time, and venue are editable.
 - 4.7.3. Event status updates automatically based on time or tickets.
- 4.8. Creating Events:** As an event host i want to be able to create events on Open deck for customers and creators to sign up for the performance slot and sell tickets.
 - 4.8.1. Form includes fields for event name, location, date, time, genre, and performance slots.
 - 4.8.2. Hosts can upload an event poster.
 - 4.8.3. Event status is automatically set to “Open” or “Closed” Deck

5. UML CLASS DIAGRAM



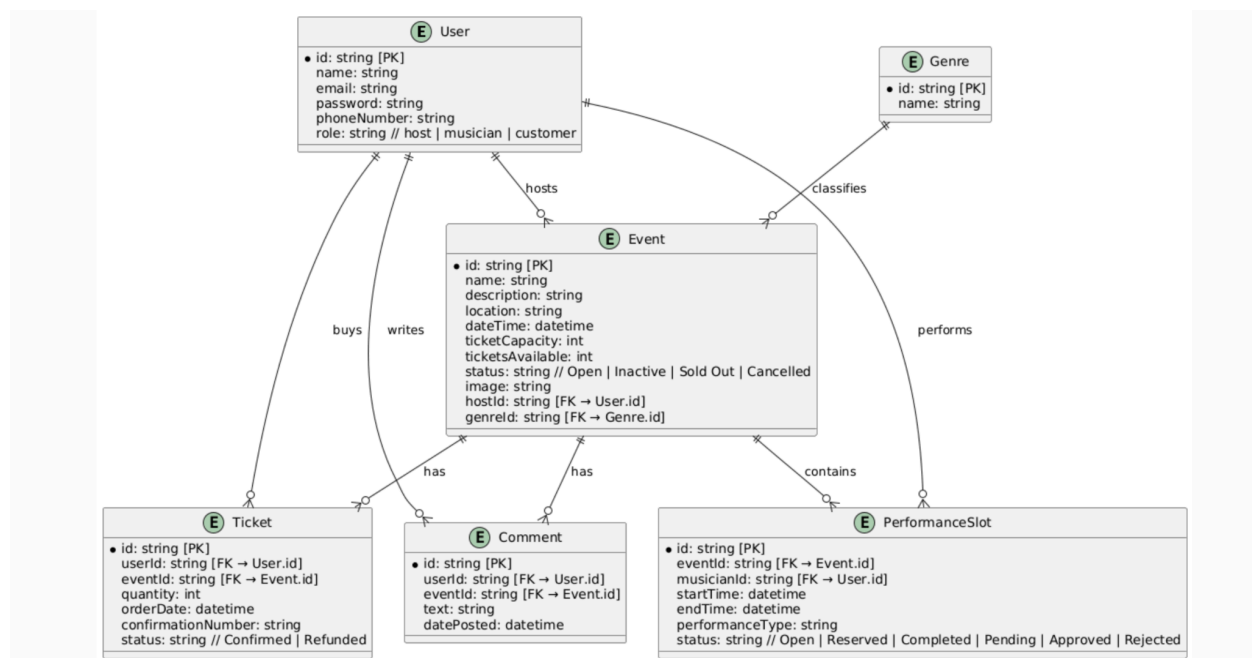
It identifies the main classes as User, Event, Ticket, PerformanceSlot, Comment, and Genre and then specifies their attributes and operations. For instance, the Event class groups together attributes related to event management, such as description, date, and ticketing capacity, while the User class centralizes user details and their role within the system.

The associations in the class diagram are justified because they reflect the logical connections between objects in the domain. A User class can be linked to Event as a host, to Ticket as a customer, and to PerformanceSlot as a musician. Each of these associations highlights the

different responsibilities users hold in the platform. One Event can have many Tickets, but each Ticket is linked to a singular Event and one Event can contain multiple PerformanceSlots, but each PerformanceSlot belongs to exactly one Event.

The inclusion of entities like Comment and Genre shows social community inclusivity will often help to maintain each artist in the app, further creating certain features like following, and posting their music in the future. Comments provide a way for users to engage with events while Genres allow events to be discovered more easily by filtering them.

6. Data Model



The UML data model for Open Decks illustrates key entities of the platform, including users, events, tickets, performance slots, comments, and genres. Each entity is defined with attributes that describe its core information. For example, the User entity stores identity and role information and while the Event manages details such as name, location, time, capacity, and status. This structure ensures that all the main functions of the platform, managing events, booking tickets, and assigning performance slots are up to date and visible

Relationships between entities are defined using UML associations with multiplicities. A user can host many events, but each event must have exactly one host. Customers can purchase

many tickets, and each ticket must be tied to one event and one customer. Musicians can perform in multiple performance slots, and events can have multiple slots to accommodate different performers. Comments also connect users to events, allowing interaction and feedback. Genres act as categories and link to multiple events for classification.

Events naturally involve many tickets and often multiple performers, while users can act as customers, musicians, or hosts depending on their role. Event having only one host and each ticket belonging to both a user and an event. Separating entities like Genre and Comment, makes the model avoid duplication and remain flexible for future expansion. This approach makes it easy for both stakeholders and developers to understand how users, events, and supporting elements interact within the system. The Model was made on PlantUML.

References:

Zhao, L., Chen, Y., & Wang, Z. (2022). *Web application development: Challenges and opportunities*. Journal of Software Engineering, 18(3), 45-59.

Sahin, Y., & Akbulut, A. (2020). *Mobile application usability and performance*. International Journal of Mobile Computing, 12(1), 15-28.

Google Developers. (2023). *Progressive Web Apps*. Retrieved from <https://web.dev/progressive-web-apps/>

Appendix 1: Use of AI Declaration in IAB207

Section A: I confirm that I did not use AI at all for my assignment.