Event Finder CLI Requirements

Background

Event Finder is an interactive command-line application that enables members of a campus community to discover, create, and manage events stored in a PostgreSQL database. This document describes the data model, functional and non-functional requirements, and detailed use-case scenarios.

Scope

- Target users are students, faculty, and staff at a college or university.
- Command Line Interface (CLI) based application.
- PostgreSQL database that stores all application data.

Glossary

User: A registered person who can log in to the system.

Organizer: A user who creates events (any user may do so).

Attendee: A user who has joined an event.

Event: A scheduled gathering with optional capacity and deadline.

Venue: A physical or virtual location where events take place.

Category: A label that groups similar events.

Entities and Attributes

USER_ACCOUNT

- user_id – integer, primary key, auto-generated
- username – string, required, unique
- email – string, required, unique, valid e-mail format
- first_name – string, required

- last_name – string, required
- password_hash – string, required, stored using bcrypt
- phone – string, optional
- year_in_school – string, optional
- major – string, optional
- attended_event_ids – integer array, list of joined event IDs
- is_active – boolean, default true
- created_at – timestamp
- updated_at – timestamp

CATEGORY

- category_id – integer, primary key
- name – string, required, unique
- description – text, optional
- created_at – timestamp

VENUE

- venue_id – integer, primary key
- name – string, required
- address – text, optional
- capacity – integer, optional
- description – text, optional
- created_at – timestamp

EVENT

- event_id – integer, primary key
- title – string, required
- description – text, optional
- event_datetime – timestamp, required
- end_datetime – timestamp, must be later than event_datetime
- venue_id – foreign key to VENUE, required
- category_id – foreign key to CATEGORY, required
- organizer_id – foreign key to USER_ACCOUNT, required

- max_attendees – integer, optional, greater than zero
- current_attendees – integer, default 0
- registration_deadline – timestamp, must be on or before event_datetime
- is_public – boolean, default true
- is_active – boolean, default true
- created_at – timestamp
- updated_at – timestamp

Relationships

- One CATEGORY may have many EVENT records.
- One VENUE may host many EVENT records.
- One USER_ACCOUNT may organize many EVENT records.
- Attendance is a many-to-many relationship tracked by the attended_event_ids array in USER_ACCOUNT.

Application Requirements

User Registration

- Collect first name, last name, username, e-mail, and password.
- Username and e-mail address must each be unique.
- Store password as a bcrypt hash.

User Login

- Authenticate with e-mail and password.
- On success, greet the user and open the user menu.

List Upcoming Events

- Display all future, active events sorted by date and time.
- Each row shows ID, title, date/time, category, venue, and attendee count.

Create Event

- Any logged-in user can create an event.
- User may create a new category or venue during the workflow.
- All data validations and business rules apply.
- registration_deadline cannot be later than event_datetime.

Join Event

- User selects an event ID.
- System enforces capacity, deadlines, and prevents duplicates.
- current_attendees must never exceed max_attendees (when max_attendees is set).
- A user cannot join the same event more than once.

Leave Event

- User selects an event ID they have joined.
- System decrements attendee count in a safe transaction.

Logout and Exit

- User can log out (return to main menu) or exit the application.

Useability

- Runs on Node 18 or later and PostgreSQL 12 or later across Windows, macOS, and Linux.
- CLI prompts include validation messages and clear labels.
- Listing events must complete in under one second