Manage Events Package

Relationships Between Classes

- **CreateEventGUI**, **EditEventDetailsGUI**, **ViewEventDetailsGUI**, **EventTabManageGUI**, and **OrganizedEventsGUI** are GUI classes responsible for displaying and interacting with event-related content.
- **EventController** handles the creation, editing, deletion, liking of events, and registration for reminders. It is the main logic layer behind all GUI-driven event actions.
- **EventController** interacts with **ProxyDBController** to persist and retrieve event data (create, update, delete, like, and reminders).
- **EventEntity** is the core event data model used throughout controllers and GUI components.
- **AccountController** invokes methods on **EventController** when users create or manage events tied to their account or venue.

EventEntity

Attributes

- `id: integer` — Unique identifier for the event.
- `ownerId: integer` — ID of the user or venue that created the event.
- `name: string` — Title of the event.
- `description: string` — Details about the event.
- `dateTime: DateTime` — Date and time when the event starts.
- `location: string` — Address or venue where the event is held.
- `images: string[]` — List of URLs or paths for event images.
- `tags: string[]` — Categories or themes assigned to the event.
- `likes: integer` — Number of likes the event has received.

Methods

- `EventEntity(...)` — Constructs a new event with all relevant attributes.

EventController

Methods

- `EventController()` — Initializes the event controller instance.
- `likeEvent(accountId: int, eventId: int)` — Adds a like from a user to a specific event.
- `registerEventReminder(accountId: int, eventId: int)` — Registers the user to receive event reminders.
- `createEvent(name, ownerId, description, date, location, startingTime, images)` — Creates a new event entry.
- `editEvent(id, ownerId, name, description, date, location, startingTime, images)` — Modifies an existing event.
- `deleteEvent(eventId: int)` — Removes the event.
- `getEvent(eventId: int)` — Returns the `EventEntity` with the specified ID.

ViewEventDetailsGUI

Attributes

- `btnMorePics: Button` — Triggers image gallery view.
- `eventTitle: String` — Stores the title of the viewed event.
- `btnPicThumbnails: Button[]` — Thumbnail buttons for image browsing.
- `eventDescription: String` — Event text content.
- `eventLocation: String` — Location of the event.
- `eventDateTime: DateTime` — Event start time.
- `eventCategories: String[]` — Event tags/categories.
- `btnHome, btnSearch, btnProfile: Button` — Navigation buttons.

Methods

- `ViewEventDetailsGUI()` — Displays details of a specific event.
- `btnHomePress()` — Navigate to home page.
- `btnSearchPress()` — Navigate to search screen.
- `btnProfilePress()` — Navigate to user profile.
- `btnPicThumbnailsPress()` — Expand image thumbnails.
- `btnMorePicsPress()` — Loads additional event images.

CreateEventGUI

Attributes

- `txtTitle: TextField` — Input for event title.

- `txtDescription: TextArea` — Input for description.
- `lblVenueLocation: Label` — Label displaying selected venue location.
- `btnUploadPhotos: Button` — Upload event images.
- `btnSelectLocation: Button` — Location picker trigger.
- `isSelectLocationEnabled: bool` — Toggle for location input.
- `btnSelectDateTime: Button` — Opens date/time picker.
- `btnSelectCategories: Button` — Opens tag/category selector.
- `btnHome, btnSearch, btnProfile: Button` — Navigation buttons.

Methods

- `CreateEventGUI()` — GUI for new event creation.
- `btnUploadPhotosPress()` — Trigger for uploading images.
- `btnSelectLocationPress()` — Open location input.
- `btnSelectDateTimePress()` — Open date/time selector.
- `btnSelectCategoriesPress()` — Open categories selector.
- `btnHomePress()` — Go to home.
- `btnSearchPress()` — Go to search.
- `btnProfilePress()` — Go to profile.

EditEventDetailsGUI

Attributes

- `eventPics: Image[]` — Images linked to the event.
- `fldTitle: Field` — Field to edit event name.
- `fldDescription: Field` — Field to edit description.
- `btnLocation: Button` — Edit location button.
- `btnEditPhotos: Button` — Edit photos button.
- `btnDateTime: Button` — Edit date/time.
- `btnSeeCategories: Button` — See/edit categories.
- `btnHome, btnSearch, btnProfile: Button` — Navigation buttons.

Methods

- `EditEventDetailsGUI()` — GUI for editing an event.
- `btnHomePress()` — Navigate home.
- `btnSearchPress()` — Navigate search.

- `btnProfilePress()` — Navigate profile.
- `editTitlePressed()` — Edit event title.
- `editDescriptionPressed()` — Edit event description.
- `btnLocationPress()` — Edit event location.
- `btnEditPhotosPress()` — Trigger photo editing.
- `btnDateTimePress()` — Modify start date/time.
- `btnSeeCategoriesPress()` — Modify event tags.

OrganizedEventsGUI

Attributes

- `btnHome, btnSearch, btnProfile: Button` — Navigation buttons.
- `eventTabGroups: EventTabManageGUI[]` — List of events displayed in tabs.

Methods

- `OrganizedEventsGUI()` — Displays all events organized by the user.
- `btnHomePress()` — Go to home.
- `btnSearchPress()` — Go to search.
- `btnProfilePress()` — Go to profile.

EventTabManageGUI

Attributes

- `eventImg: Image` — Preview image of event.
- `eventName: String` — Event name display.
- `eventStatus: EventStatus` — Status of event (live, not-live).
- `eventLikeCounter: int` — Like count.
- `userHasLiked: bool` — Flag indicating if current user has liked.
- `btnLikeEvent, btnDeleteEvent, btnEditEvent: Button` — Action buttons.

Methods

- `EventTabManageGUI()` — Tab representing one event in list.
- `btnEditEventPress(event: EventEntity)` — Triggers edit for selected event.
- `btnDeleteEventPress(event: EventEntity)` — Deletes selected event.

- `btnLikeEventPress(event: EventEntity)` — Likes the event.
- `showEmptyListMessage()` — Displays message if no events exist.