

## **UMPLE CODE** (see in project wiki)

- // ManageEventActivity handles both event creation and editing.
- // In edit mode, it uses a temporary reference to the existing Event object ('eventToEdit').
- // This is not a true 1–1 ownership it's just a transient link for editing, so we do NOT model it as a strong UML association.
- // The activity depends on OrganizerViewModel and CategoryViewModel to fetch and update event and category data,
- ${\it //}\ but\ the\ ViewModels\ do\ not\ store\ a\ reference\ to\ the\ activity,\ so\ the\ relationships\ are\ one-sided.}$
- // OrganizerDashboardActivity launches ManageEventActivity for editing or creating events.
- // This is not a persistent relationship, just a usage dependency.

// We only include meaningful fields like `organizerId`, `selectedCategoryId`, and `isEditMode` that represent //persistent state or logic. UI-specific components (EditTexts, Buttons, etc.) are intentionally excluded to keep //the diagram focused on structure and behavior, not layout.

class AppCompatActivity {}

```
// ======= MODELS =======

class Event {
    String eventld;
    String organizerld;
    String description;
    String categoryld;
    double fee;
    long eventStart;
    long eventEnd;

// Each Event refers to exactly one Category
    1 -- 1 Category;
}

class Category {
    String categoryld;
```

```
String name;
 String description;
                                                                          // ======= SERVICES ======
                                                                          class LoginService {
class RegistrationForm {
                                                                           // Auth and user parsing
 String firstName;
                                                                           FirebaseAuth mAuth;
 String lastName;
                                                                           DatabaseReference usersRef;
 String username;
 String email;
                                                                          // ====== ACTIVITIES ======
 String phone;
 String password;
 String confirmPassword;
                                                                          // Dashboard Activities
 boolean isOrganizer;
                                                                          class AdminDashboardActivity {
 String companyName;
                                                                           isA AppCompatActivity;
                                                                           // Activity holds one AdminViewModel
                                                                           1 -- 1 AdminViewModel;
// ====== ACCOUNTS ======u
class UserAccount {
                                                                          class OrganizerDashboardActivity {
 String userID;
                                                                           isA AppCompatActivity;
                                                                           1 -- 1 OrganizerViewModel;
 String firstName;
 String lastName;
 String username;
 String email;
                                                                          class ParticipantDashboardActivity {
 String phoneNumber;
                                                                           isA AppCompatActivity;
 String role;
                                                                            1 -- 1 ParticipantViewModel;
class AdminAccount {
                                                                          // -----
 isA UserAccount;
                                                                          class ManageCategoriesActivity {
                                                                           isA AppCompatActivity;
class OrganizerAccount {
                                                                           1 -- 1 CategoryViewModel;
 isA UserAccount;
 String companyName;
                                                                          class ManageEventActivity {
 // Each OrganizerAccount can own many Events
                                                                           isA AppCompatActivity;
                                                                           boolean isEditMode;
 1 -- * Event;
                                                                           Event eventToEdit;
                                                                           // eventToEdit is a temporary reference, not a persistent ownership
class ParticipantAccount {
                                                                           String organizerId;
 isA UserAccount;
                                                                            String selectedCategoryId;
                                                                           List<Category> categoryList;
                                                                            1 -- 1 OrganizerViewModel;
// ====== REPOSITORIES ======
                                                                            1 -- 1 CategoryViewModel;
class EventRepository {
// Repository manages many Events
 1 -- * Event;
                                                                          class CreateAccountActivity {
isA AppCompatActivity;
                                                                            1 -- 1 RegistrationForm;
class AdminViewModel {
 // ViewModel uses exactly one EventRepository (to manage Events)
 DatabaseReference userRef;
 1 -- * UserAccount;
                                                                          class LoginActivity {
                                                                           isA AppCompatActivity;
                                                                            1 -- 1 LoginService;
class OrganizerViewModel {
 EventRepository eventRepo;
                                                                          class MainActivity {
 1 -- 1 EventRepository;
                                                                           isA AppCompatActivity;
class ParticipantViewModel {
}
class CategoryViewModel {
 DatabaseReference dbRef;
 1 -- * Category;
```