# Map/List/Search Package – Class Overview

## MapGUI

### Attributes

- `fldSearch: Field` – search input field
- `btnListView: Button` – switches to list view
- `btnCenterLocation: Button` – centers map on user
- `btnPin: Button` – interacts with map pin
- `btnHome: Button` – navigates home
- `btnSearch: Button` – triggers search
- `btnProfile: Button` – opens profile

### Methods

- `MapGUI()` – constructor; initializes map-related UI
- `btnHomePress(): void` – handles home button
- `btnSearchPress(): void` – handles search button
- `btnProfilePress(): void` – handles profile button
- `btnPinPress(eventId: Integer): void` – selects a pin/event on map
- `btnCenterLocationPress(): void` – recenters map on user location
- `btnListViewPress(): void` – switches from map to list view
- `fldSearchSubmitPressed(): void` – submits search from text field

## ListGUI

### Attributes

- `btnHome: Button` – navigates home
- `btnSearch: Button` – triggers search
- `btnProfile: Button` – opens profile
- `fldSearch: Field` – search input field
- `btnMapView: Button` – switches to map view
- `btnViewAllFilters: Button` – opens all filters

- `btnClearFilters: Button` – clears all filters
- `eventTabGroups: EventTabListGUI[]` – collection of event tabs

### Methods

- `ListGUI()` – constructor; initializes list-related UI

## EventTabListGUI

### Methods

- `EventTabListGUI(event: EventEntity)` – constructor for event tab
- `btnLikePress(): void` – triggers like logic
- `btnShareEventPress(): void` – shares the event
- `btnEventReminderPress(): void` – sets a reminder
- `openEventDetails(): void` – opens event details view

## FilterTabsPopUp

### Attributes

- `btnCategories: Button` – open category filters
- `btnClear: Button` – clears selected filters
- `selectedTags: String[]` – list of selected tags

### Methods

- `populateCategories(eventCategories: String[]): void` – fills UI with categories
- `selectButton: Button` – confirms selection
- `btnClearPress(): void` – clears selected categories
- `getSelectedTags(): String[]` – returns selected tags
- `popDialog(): void` – closes popup

## FilterEntity

### Attributes

- `type: String` – filter type (distance, time, event type etc.)

- `value: String` – filter value (e.g., "<1km", "in 2hrs", "rock")

## SearchController

### Attributes

- `filters: List<FilterEntity>` – current applied filters

### Methods

- `SearchController(filters: List<FilterEntity>)` – constructor with filter list
- `searchByKeyword(keyword: String): List<EventEntity>` – searches events by keyword
- `searchByCategory(category: String): List<EventEntity>` – filters events by category
- `searchByLocation(tags: List<LocationEntity>): List<EventEntity>` – filters events by location
- `searchByUsername(names: List<String>): List<AccountEntity>` – finds users by name
- `applyFilters(filters: List<FilterEntity>): List<EventEntity>` – applies all filters

## GoogleMapsController

### Methods

- `initializeMap(userLocation: String): void` – sets up map for given location
- `refreshNearbyEvents(): List<EventEntity>` – fetches nearby events
- `updateMapRegion(region: MapRegion): void` – updates visible map area
- `switchToListView(events: List<EventEntity>): void` – changes view to list
- `drawRouteToEvent(event: EventEntity): void` – shows route to event