

# Event Management System

## 1. Introduction

This system is designed to manage events, organizers, and venues for a company, university, or community. It allows users to perform CRUD operations (Create, Read, Update, Delete) on these entities, ensuring data integrity and ease of use. All actions include validation to ensure information consistency.

## 2. Functional Requirements

### 2.1 Function 1: Create Event (2 Marks)

The system requires input for creating a new event. The information to be provided includes:

- **Event ID (int):** A unique identifier for the event.
- **Event Name (String):** The title of the event.
- **Organizer ID (int):** A reference to the organizer's ID.
- **Venue ID (int):** A reference to the venue's ID.
- **Start Date (String):** The event's start date (YYYY-MM-DD).
- **End Date (String):** The event's end date (YYYY-MM-DD).
- **Expected Attendees (int):** Estimated number of participants.

#### Validation Rules:

1. Event ID must be unique.
2. Event Name must be at least 3 characters long.
3. Organizer ID must exist in the list of organizers.
4. Venue ID must exist in the list of venues.
5. Start Date and End Date must be valid dates (YYYY-MM-DD) and End Date must not be before Start Date.
6. Expected Attendees must be positive (greater than zero).

**If any validation fails, display a relevant error message and prompt for re-entry.**

### 2.2 Function 2: Display All Events (2 Marks)

- Show a list of all events, sorted alphabetically by event name.
- If names match, sort by Event ID ascending.

### 2.3 Function 3: Update Event (2 Marks)

- User enters Event ID.
- If not found: display "Event not found."
- Otherwise, user can update any field (leave blank to keep current value).

- Display “Update successful” or “Update failed.”

#### 2.4 Function 4: Delete Event (1 Mark)

- User enters Event ID.
- If not found: display “Event not found.”
- Otherwise, confirm and delete, then display “Delete successful.”

#### 2.5 Function 5: Find Events by Name (2 Marks)

- Search for events containing a given string in their name (case-insensitive).
- Display all matches, sorted alphabetically.
- If none: “No events found.”

#### 2.6 Bonus (1 Mark)

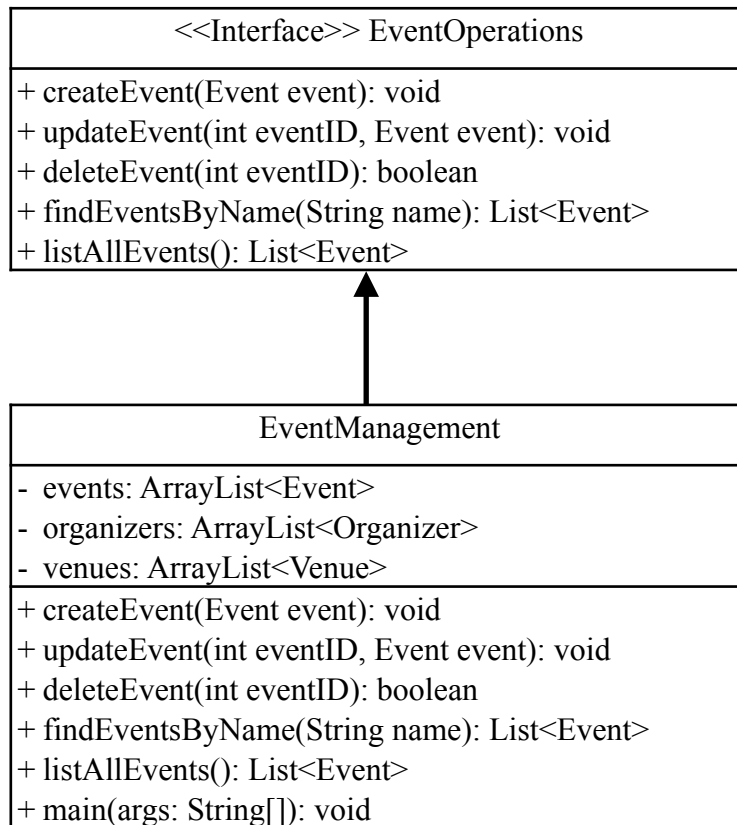
- The system can save events data to a file for future use (persistence).

### 3. UML Class Diagram

Organizer
- organizerID: int - organizerName: String
+ getOrganizerID(): int + setOrganizerID(id:int): void + getOrganizerName(): String + setOrganizerName(name:String): void + toString(): String

Venue
- venueID: int - venueName: String
+ getVenueID(): int + setVenueID(id:int): void + getVenueName(): String + setVenueName(name:String): void + toString(): String

Event
- eventID: int - eventName: String - organizerID: int - venueID: int - startDate: String - endDate: String - expectedAttendees: int
+ getEventID(): int + setEventID(id:int): void + getEventName(): String + setEventName(name:String): void + getOrganizerID(): int + setOrganizerID(id:int): void + getVenueID(): int + setVenueID(id:int): void + getStartDate(): String + setStartDate(date:String): void + getEndDate(): String + setEndDate(date:String): void + getExpectedAttendees(): int + setExpectedAttendees(n:int): void + toString(): String



## 4. Initial Data for List of Organizers and Venues

### 4.1 List of Organizers

```
1. List<Organizer> organizers = new ArrayList<>();
2. organizers.add(new Organizer(1, "EventPro Solutions"));
3. organizers.add(new Organizer(2, "Green Light Agency"));
4. organizers.add(new Organizer(3, "University Youth Union"));
5. organizers.add(new Organizer(4, "TechWorld Group"));
6. organizers.add(new Organizer(5, "Community Council"));
7. organizers.add(new Organizer(6, "City Entertainment Co."));
8. organizers.add(new Organizer(7, "Art & Culture Center"));
```

### 4.2 List of Venues

```
1. List<Venue> venues = new ArrayList<>();
2. venues.add(new Venue(1, "Grand Hall"));
3. venues.add(new Venue(2, "City Conference Center"));
4. venues.add(new Venue(3, "Open Air Park"));
5. venues.add(new Venue(4, "Exhibition Pavilion"));
6. venues.add(new Venue(5, "Downtown Auditorium"));
7. venues.add(new Venue(6, "Community House"));
8. venues.add(new Venue(7, "University Main Hall"));
```

## **5. User Interface Requirements**

- The system will be a console-based application, with a menu-driven interface. The user will be presented with a menu that allows them to select from the available functions (Create, Read, Update, Delete, and Search).
- Sample Main Menu:
  1. Create Event
  2. Display All Events
  3. Update Event
  4. Delete Event
  5. Find Events by Name
  6. Save to file (Optional)
  7. Exit

**-- THE END --**

# AI Tools & Code Quality Requirements

## 1. Introduction

In addition, students are required to use **AI tools (Snyk AI and ChatGPT)** for code analysis, debugging, and algorithm/pseudocode development.

## 2. AI Tools & Code Quality Requirements

### 2.1 AI Code Analysis & Debugging (Snyk AI) – (5 Marks)

- Run Snyk AI (via IDE plugin) on your project to check for security vulnerabilities, code quality issues, and bugs.
- Save and submit the generated Snyk AI analysis report (as a PDF or screenshot).
- Fix at least one critical or high-severity issue found by Snyk AI, and describe the fix in a short paragraph (3–5 sentences) in your final report.

#### • Sample Snyk AI Scan Evidence and Description

##### 1. Snyk AI Scan Evidence (Example Screenshot Text)

1. Snyk Scan Results – Event Management System Project
- 2.
3. [High Severity]
4. Issue: Insecure use of java.util.Random
5. File: src/util/EventCodeGenerator.java, line 23
6. Description: Use of java.util.Random for security-sensitive operations can lead to predictable values. Consider using java.security.SecureRandom.
7. Recommendation: Replace java.util.Random with java.security.SecureRandom
- 8.
9. [Medium Severity]
10. Issue: Potential NullPointerException
11. File: src/EventManager.java, line 88
12. Description: Possible dereference of null object 'foundEvent' before calling toString().
13. Recommendation: Add a null check before calling methods on potentially null objects.

##### 2. Description of the Fix (for Report)

1. During the development of the Event Management System, I ran a Snyk AI scan using the Snyk IDE plugin for Netbeans.  
Snyk identified a high-severity issue: insecure use of java.util.Random in EventCodeGenerator.java for event code generation.  
To fix this, I replaced Random rand = new Random(); with SecureRandom rand = new SecureRandom(); as recommended.
2. After applying the fix and rescanning, Snyk reported no further high-severity issues.
3. Additionally, Snyk found a possible NullPointerException in EventManagement.java on line 88. I resolved this by adding a null check before calling toString() on the foundEvent object.

## 2.2 Pseudo Code & Algorithm Analysis (ChatGPT) – (5 Marks)

- **Requirement:**

For at least one core function, students must use ChatGPT to:

- Generate pseudo code for the function, and include the prompt & response in the report.
- Analyze (may ask ChatGPT directly for suggestions).

- **Expected Deliverables:**

- The original ChatGPT prompt and response for pseudo code generation.
- Algorithm Analysis (can be from ChatGPT).

- **Sample: Pseudo Code & Algorithm Analysis (findEventsByName)**

- Prompt Sent to ChatGPT

1. Please write the pseudo code for a function named findEventsByName that searches for events containing a search string in their name (case-insensitive), from a list of Event objects. Each Event has eventID, eventName, and other fields. The function should return a list of matching events sorted alphabetically by eventName.

- ChatGPT Response (Pseudo Code)

1. Pseudo Code: findEventsByName
- 2.
3. FUNCTION findEventsByName(eventList, searchString):
4.     resultList ← empty list
5.     LOWER\_SEARCH ← lowercase(searchString)
6.     FOR each event IN eventList:
7.         IF lowercase(event.eventName) contains LOWER\_SEARCH:
8.             add event TO resultList
9.     SORT resultList BY eventName (alphabetically)
10.    RETURN resultList

- **Algorithm Analysis**

- **Case-insensitive Search:** The algorithm converts both the event's name and the search string to lowercase to ensure case-insensitive matching.
- **Sorting:** Sorting ensures the output is user-friendly and meets assignment requirements.
- **Edge Cases:**
  - If no events match, the function returns an empty list.
  - If search string is empty, all events will be returned (could be restricted by an extra check).

### 3. How to Submit

Along with your Java project (code and report), include:

- Snyk AI Scan Evidence and Description report (PDF file).
- A section with ChatGPT prompt, pseudo code and algorithm analysis (PDF file).

# How to Use Snyk AI Plugin in Visual Studio Code

## 1. Install the Snyk Extension

- Open **Visual Studio Code** -> **Go to the Extensions panel (Ctrl+Shift+X)**.
- Search for “**Snyk**” -> Click “**Install**” on the **Snyk Security** extension by Snyk.

## 2. Authenticate Snyk in VS Code

- After installation, you’ll see a Snyk icon in the Activity Bar on the left -> Click the Snyk icon. You’ll be prompted to authenticate:
  - Click “Sign in with Snyk.”
  - Complete the login via your browser (create a free Snyk account if needed).

## 3. Open Your Project

- In VS Code, open the folder containing your project (Java, etc.).

## 4. Run a Snyk Scan

- Click the **Snyk icon** to open the Snyk panel. -> Click “**Test**” (or “**Scan Project**”) to scan your open folder.
- Wait for the scan to complete (this may take a few seconds).

## 5. Review the Results

- Results are displayed directly in the Snyk panel.
  - **Security vulnerabilities** in dependencies (e.g., Maven, npm, etc.).
  - **Code quality issues & bugs** (if Snyk Code is enabled).
- Each issue displays:
  - **Severity** (Low/Medium/High/Critical)
  - **File & line number**
  - **Description and recommended fix**

## 6. Fix Issues and Re-scan

- Click on an issue to jump to the line of code -> Apply the suggested fix (e.g., update a library, improve code).
- Re-run the scan to check that the issue is resolved.

## 7. Export or Screenshot the Results (for Reports)

- You can **take a screenshot** of the Snyk panel showing found issues.
- Optionally, copy and paste the results into your assignment report.

## 8. References

- <https://docs.snyk.io/scm-ide-and-ci-cd-integrations/snyk-ide-plugins-and-extensions/visual-studio-code-extension>
- <https://snyk.io/>

-- THE END --