

## Topic number 6: Event management platform

---

**Description:** Create an application that allows users to create, manage, and RSVP for events, with features like ticketing and response.

Where we can work with each other: Github, <https://github.com/QUT-code/ProjectSoftware.g>

Team: 2 students. (Sokha Ordorm, Chheng Rayuth)

Deadline: 4 weeks.

### 1. Project Overview

- + **Project Goal:** Develop an application console-based Java application for event management.
- + **Key Features:**
  - **Event Creation:** Users can create events with details like event name, date, time, location, and description.
  - **Event Management:** Users can view, update, or delete events.
  - **RSVP System:** Users can RSVP for events (e.g., "Attending," "Not Attending," "Maybe").
  - **Ticketing:** Users can generate tickets for events (optional: include ticket numbers or QR codes).
  - **Responses:** Users can view RSVP responses for their events.

### 2. Estimation Techniques

- + **Top-Down Estimation (Initial Planning):**
  - **Break the project into high-level phases:**
    - **Requirement Analysis (1 week)**
    - **Design and Architecture (1 week)**
    - **Implementation (Coding) (1.5 weeks)**
    - **Testing and Debugging (0.5 week)**

#### + **Bottom-Up Estimation (Detailed Planning):**

Break each phase into smaller tasks and estimate time, effort, and resources for each task.

#### **Example tasks:**

- **Requirement Analysis:**

- Brainstorm and finalize features (1 day).
- Define user roles (e.g., event organizer, attendee) (1 day).
- Document functional and non-functional requirements (2 days)

- **Design and Architecture:**

- Design the class diagram (e.g., Event, User, Ticket, RSVP) (2 days).
- Design the database schema (if using a database) (1 day).
- Plan the console-based user interface flow (1 day).

- **Implementation:**

- Implement the Event class with CRUD operations (3 days).
- Implement the RSVP system (2 days).
- Implement the Ticket generation feature (2 days).
- Implement the console-based user interface (3 days).

- **Testing and Debugging:**

- Write unit tests for all classes and methods (2 days).
- Debug and fix issues (2 days).

### **3. Progress Updates**

Weekly Progress Report:

Week 1:

Tasks done: Brainstormed features, defined user roles, documented requirements.

Tasks for next week: Design class diagram and database schema.

Problems: None.

Week 2:

Tasks done: Designed class diagram, database schema, and console interface flow.

Tasks for next week: Implement Event class and RSVP system.

Problems: None.

Week 3:

Tasks done: Implemented Event class and RSVP system, started Ticket generation.

Tasks for next week: Complete Ticket generation, implement console interface.

Problems: Minor issues with Ticket generation logic.

Week 4:

Tasks done: Completed Ticket generation, implemented console interface, wrote unit tests.

Tasks for next week: Final review and submission.

Problems: None.