

Storage Manager (syuk)

- ☐ FileManager.java
- ☐ Create `loadEvents()` which return `List<Event>`
- ☐ Create `saveEvents()`: It must loop through the list and write strings to the CSV.

The Object Architect (izzat)

- ☐ Event.java
- ☐ Storage manager cant do his job if file doesnt define what an Event is
- ☐ Define the class properties: `id`, `title`, `description`, `start`, `end`.
- ☐ Create the **Constructor** (so storage manager can create new events)
- ☐ Create **Getters/Setters** (so Member 3 can display the title).

The UI Lead (marchie)

- ☐ SchedulerApp.java and CalendarView.java
- ☐ **Main Menu:** Inside `SchedulerApp.java`, write the `while` loop that asks: "1. View, 2. Add, 3. Exit".
- ☐ **Visuals:** Create a method that prints the calendar grid
- ☐ **Input:** Write the code to ask the user for "Event Title" and "Date : 22/10/2023" when they select "Add Event"
- ☐ Event Update & Delete
- ☐ GUI

The logic controller (hasanul)

- ☐ SearchUtils.java
- ☐ **Conflict Detection:** Write a method `isTimeSlotFree()` that checks if a new event overlaps with an existing one
- ☐ **Search:** Write the logic to filter the event list by keyword or date.
- ☐ **Additional Data:** Handle the reading of `data/additional.csv` to add additional data

The Complex Features Engineer (kimi)

- ☐ Recurrence.java,analytics.java and RecurrenceLogic.java
- ☐ Handle recurrent event
- ☐ Handle additional features like :
- ☐ Backup data
- ☐ **Statistics:** Write the code to calculate insights, like finding the "Busiest Day of the Week".
- ☐ **Reminders:** Write the startup check that says: *"Alert: You have an event in 30 minutes!"*.
- ☐ **Cross-PC Testing:** You must pull the code to your own computer and verify that Member 1's relative paths (`data/event.csv`) work without errors.