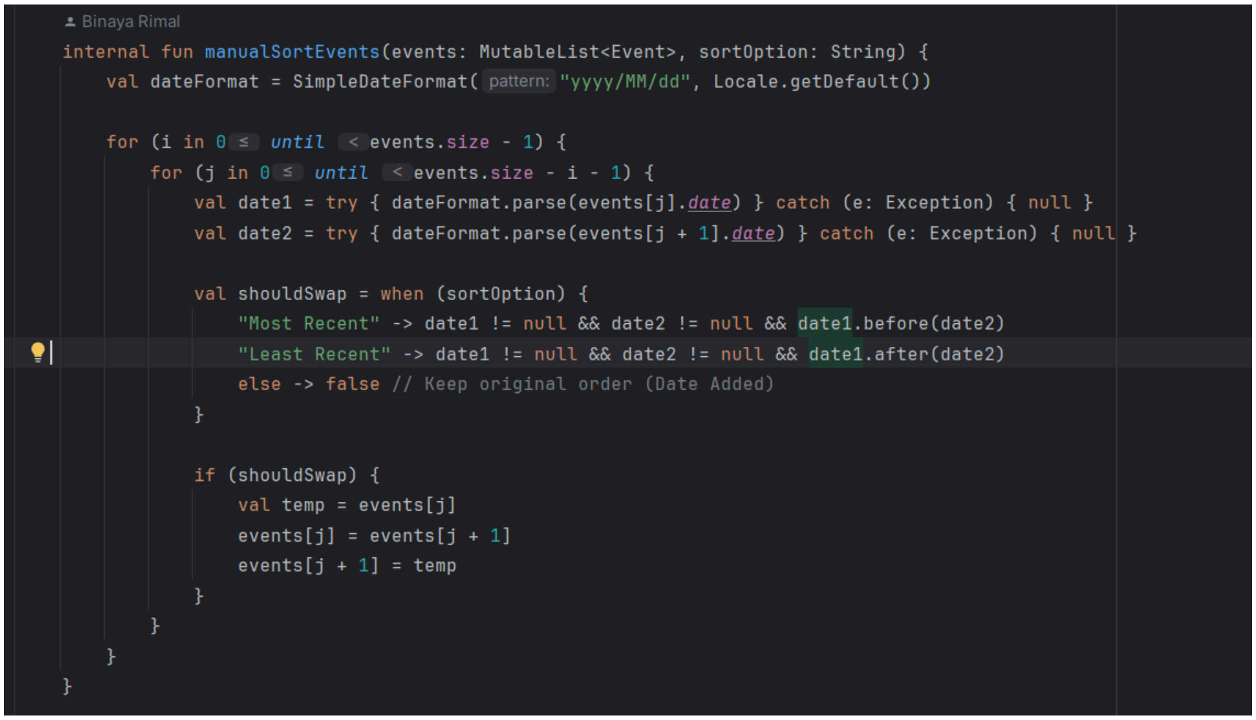
Binaya Rimal

02/02/2025

Algorithm and Data Structure Enhancement

For this milestone, our artifact remains the Event Tracker app, and we are adding a sorting feature that allow users to sort events based on most recent, least recent and date added. I have also been finalizing the conversion of the code from Java to Kotlin. Previously, the app displayed events solely in the order they were added. For the enhancement, we want to create a spanner feature on the app that allows users to select how they want the events to be sorted. Below is our sorting function:

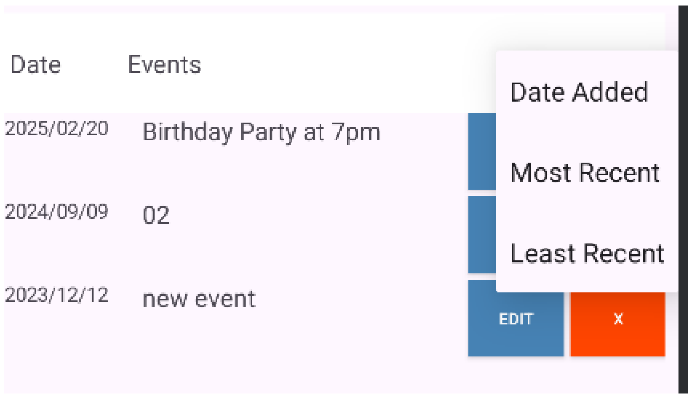
Sorting Function:



```
Binaya Rimal
internal fun manualSortEvents(events: MutableList<Event>, sortOption: String) {
    val dateFormat = SimpleDateFormat( pattern: "yyyy/MM/dd", Locale.getDefault())

    for (i in 0 ≤ until < events.size - 1) {
        for (j in 0 ≤ until < events.size - i - 1) {
            val date1 = try { dateFormat.parse(events[j].date) } catch (e: Exception) { null }
            val date2 = try { dateFormat.parse(events[j + 1].date) } catch (e: Exception) { null }

            val shouldSwap = when (sortOption) {
                "Most Recent" -> date1 != null && date2 != null && date1.before(date2)
                "Least Recent" -> date1 != null && date2 != null && date1.after(date2)
                else -> false // Keep original order (Date Added)
            }

            if (shouldSwap) {
                val temp = events[j]
                events[j] = events[j + 1]
                events[j + 1] = temp
            }
        }
    }
}
```

This Kotlin function, manualSortEvents, sorts a list of events based on a given sorting option (sortOption). It uses the **Bubble Sort** algorithm, iterating through the list and comparing adjacent event dates. The dates are parsed using SimpleDateFormat, and if parsing fails, they default to null. Depending on the sortOption, the function swaps elements when necessary: for "Most Recent," it moves newer dates forward, and for "Least Recent," it moves older dates forward. If the sort option is unrecognized, the list remains unchanged.

Below is the mobile image of our events sorted from most recent to least recent with drop down option on display:



Date	Events
2025/02/20	Birthday Party at 7pm
2024/09/09	02
2023/12/12	new event

I selected this artifact and enhancement because it aligns well with other projects I have been working on and fills a gap in my portfolio. Currently, I do not have any mobile app projects showcased, and this project demonstrates my ability to develop a well-structured, professional application that integrates frontend, backend, and database management while utilizing Kotlin, a programming language that is less commonly featured in portfolios.

The enhancement was designed with the user experience in mind. I believe that giving users control over how their events are sorted adds significant value, improving usability and making the app more flexible to accommodate different preferences. My goal is to create an intuitive and functional experience that highlights both technical efficiency and thoughtful design.

Enhancing and modifying the Event Tracker app was a valuable learning experience that strengthened my understanding of Kotlin, algorithm implementation, and mobile development. Converting the code from Java to Kotlin helped me appreciate Kotlin's efficiency and modern syntax, while implementing a sorting algorithm reinforced my problem-solving skills.

One challenge I faced while completing this project was deciding which data structure and algorithm to use. Many potential enhancements already have existing libraries that could handle sorting without requiring manual implementation. Nonetheless, I chose to implement a sorting algorithm manually to showcase my knowledge and proficiency in algorithm development. This decision allowed me to demonstrate my understanding of core computer science concepts while ensuring greater control over the sorting functionality within the app.

Below is a video demo that showcases what was done during this enhancement:

https://www.youtube.com/watch?v=Fg2B92pmaSA

Instructions to download the app:

Pre work:

1. Download Android Studio

2. Clone the Repository

Open the Project in Android Studio

1. Open **Android Studio** on your laptop.
2. Click **"Open"** or **"Open an Existing Project"**.
3. Navigate to the project folder you extracted/downloaded.
4. Click **OK** and wait for Android Studio to load the project.
5. **Wait for Gradle to sync** – If prompted, click **"Sync Now"**.
6. If you see a **missing SDK error**, go to **File > Project Structure > SDK Location** and set the correct Android SDK path.
7. If necessary, update dependencies in build.gradle and click **"Sync Now"** again.
8. Connect a **physical Android device** via USB **or** use an **Android Emulator** (AVD).
9. Click **Run**
10. Select the target device/emulator and wait for the app to launch.