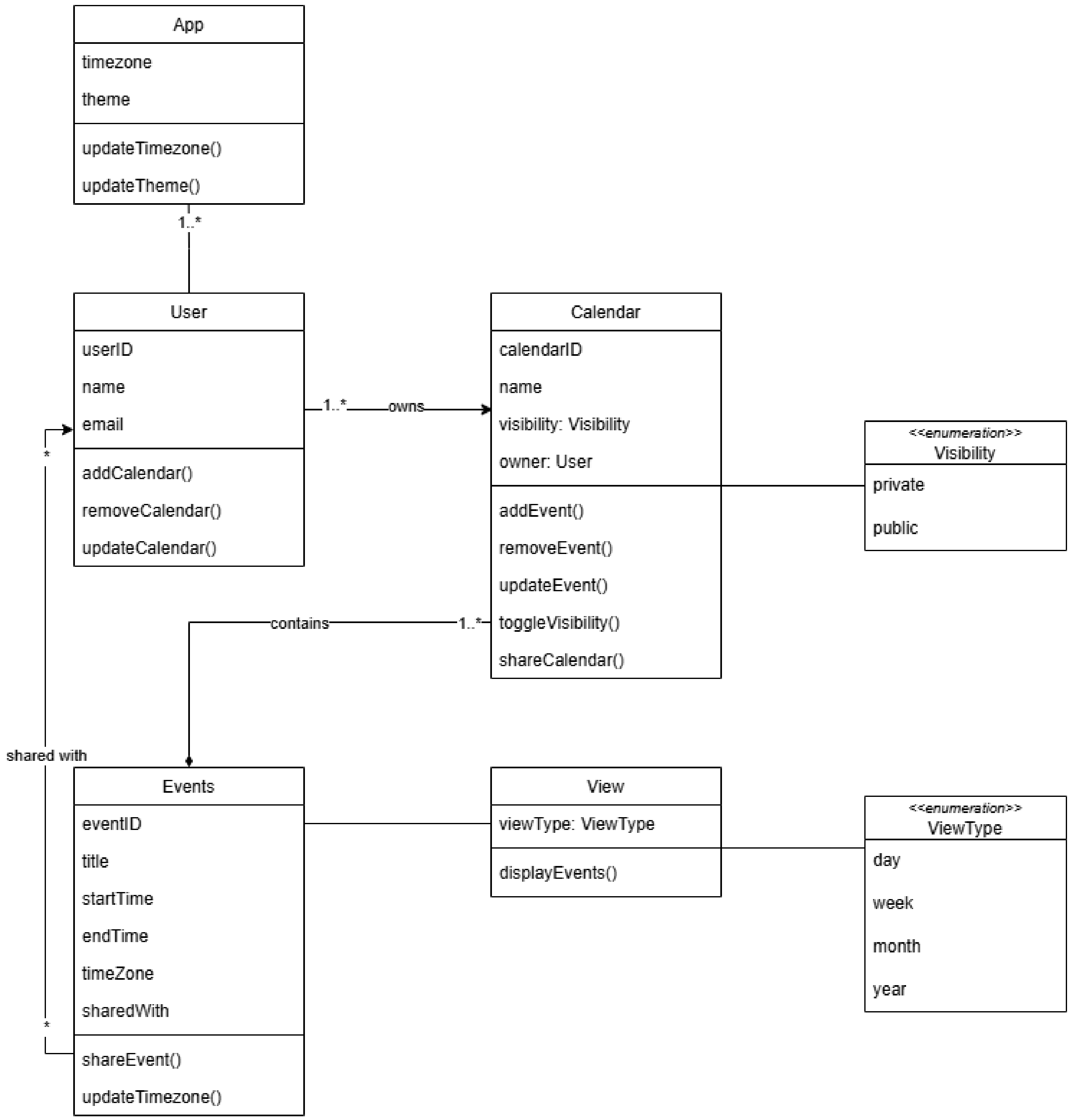
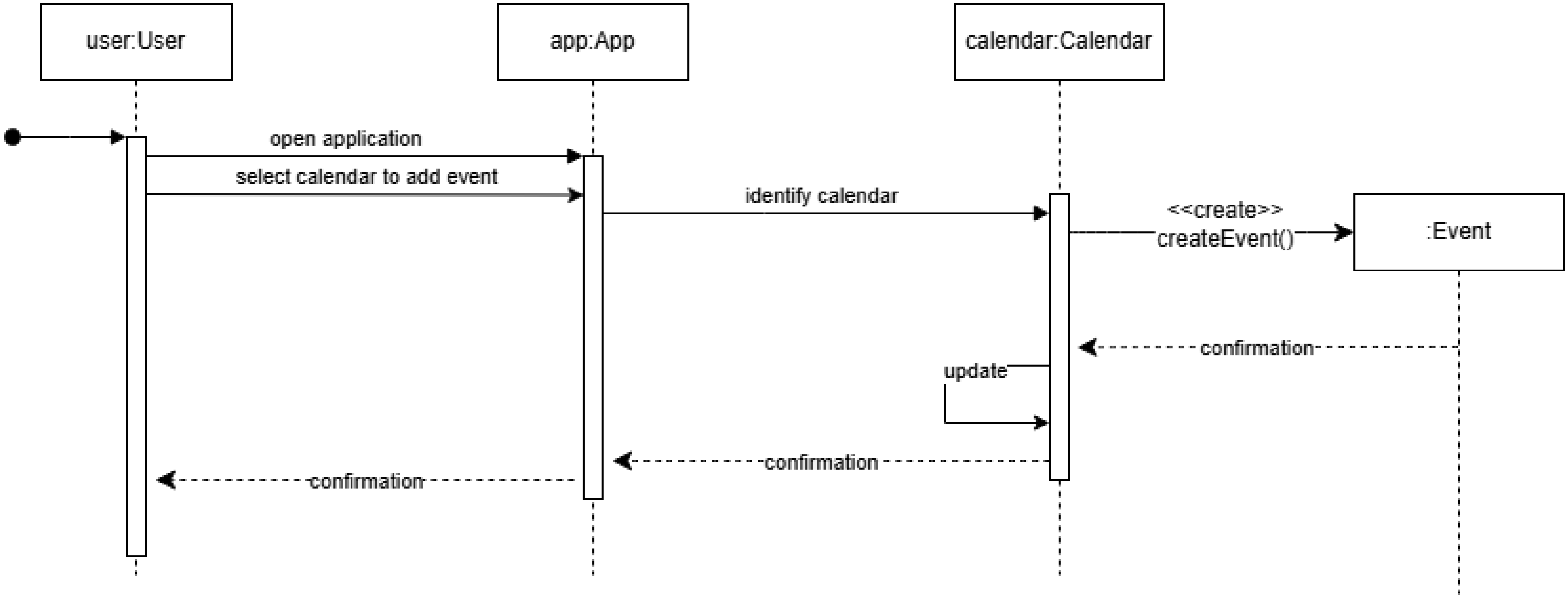
Class Diagram



Sequence Diagram



Explanation

The App class handles global settings like timezone and theme, with methods for updating them. The User class represents individual users, identifiable with attributes like userID, name, and email, and allows users to manage their calendars through methods such as addCalendar(), removeCalendar(), and updateCalendar(). The Calendar class represents a collection of events, identified by attributes like calendarID, name, and visibility - an enumeration of public or private - and provides methods to add, remove, update, and share events or toggle visibility. Events are represented by the Event class, which includes details such as eventID, title, startTime, endTime, timezone, and sharedWith. Finally, the View class manages how calendars and events are displayed, with support for day, week, month, or year views through its displayEvents() method.

Future Changes

The design is easily changeable because of the modularity of the core components.

Adding a GUI

In order to add a GUI, the View class handles the display of events and calendars. It abstracts the visualization logic from the core logic, making it easy to integrate a GUI.

The addition of other types of calendars

The Calendar class supports other types of calendars by extending the Calendar class or introducing a CalendarType enumeration.

Notifications are shown to a particular user when another user shares a calendar event

The Event class already has a sharedWith attribute, which can be modified to identify users who need to be notified. This could be done by introducing a Notification class to manage notifications, with methods like sendNotification().

The ability to support different languages

The design can support localization by using a resource file to store UI text in different languages. A Localization utility class could retrieve translated strings based on the app's current locale, which could be stored as an attribute in the App class.