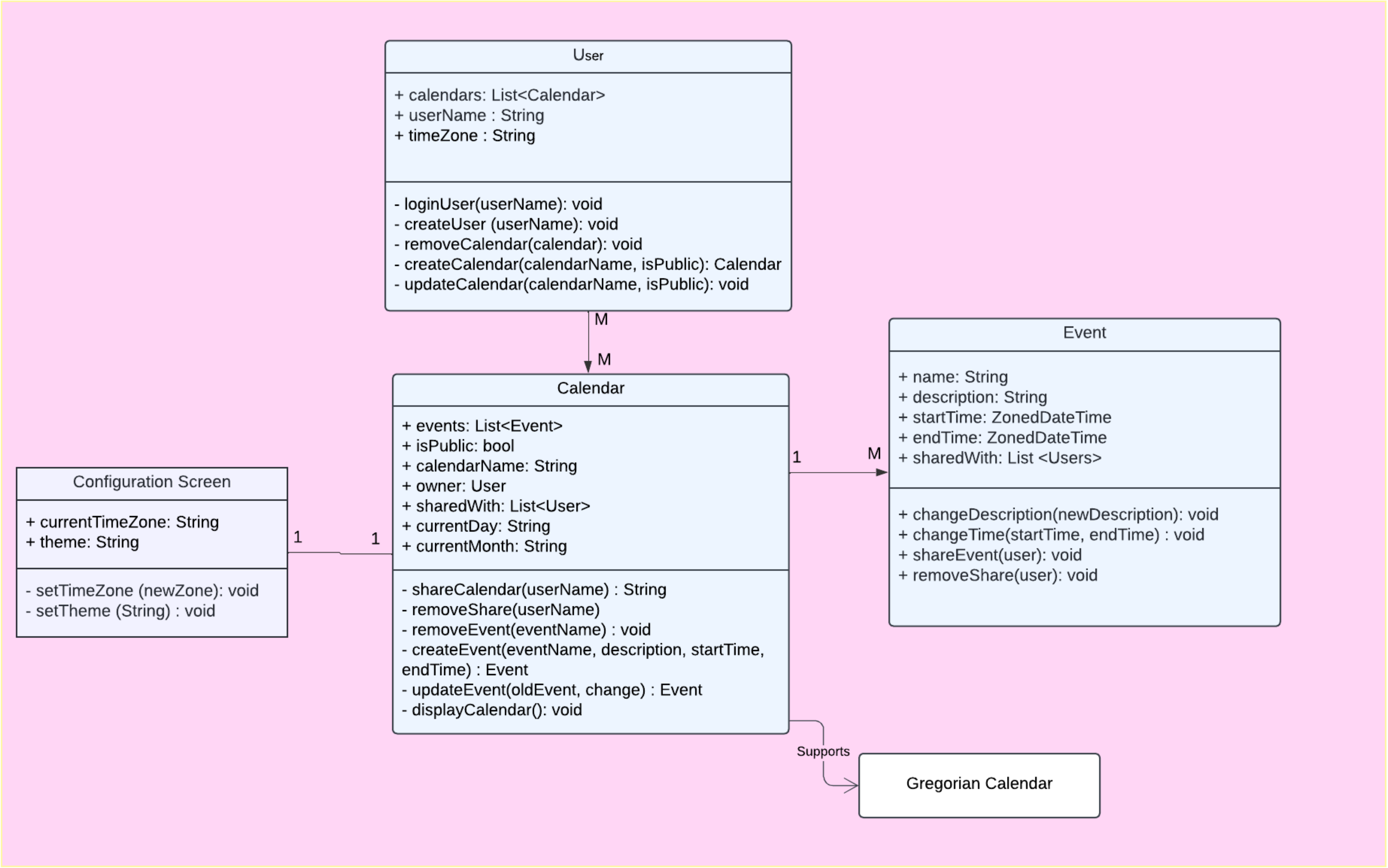
Calendars UML Diagram



User

+ calendars: List<Calendar>
+ userName : String
+ timeZone : String

- loginUser(userName): void
- createUser (userName): void
- removeCalendar(calendar): void
- createCalendar(calendarName, isPublic): Calendar
- updateCalendar(calendarName, isPublic): void

Calendar

+ events: List<Event>
+ isPublic: bool
+ calendarName: String
+ owner: User
+ sharedWith: List<User>
+ currentDay: String
+ currentMonth: String

- shareCalendar(userName) : String
- removeShare(userName)
- removeEvent(eventName) : void
- createEvent(eventName, description, startTime, endTime) : Event
- updateEvent(oldEvent, change) : Event
- displayCalendar(): void

Event

+ name: String
+ description: String
+ startTime: ZonedDateTime
+ endTime: ZonedDateTime
+ sharedWith: List <Users>

+ changeDescription(newDescription): void
+ changeTime(startTime, endTime) : void
+ shareEvent(user): void
+ removeShare(user): void

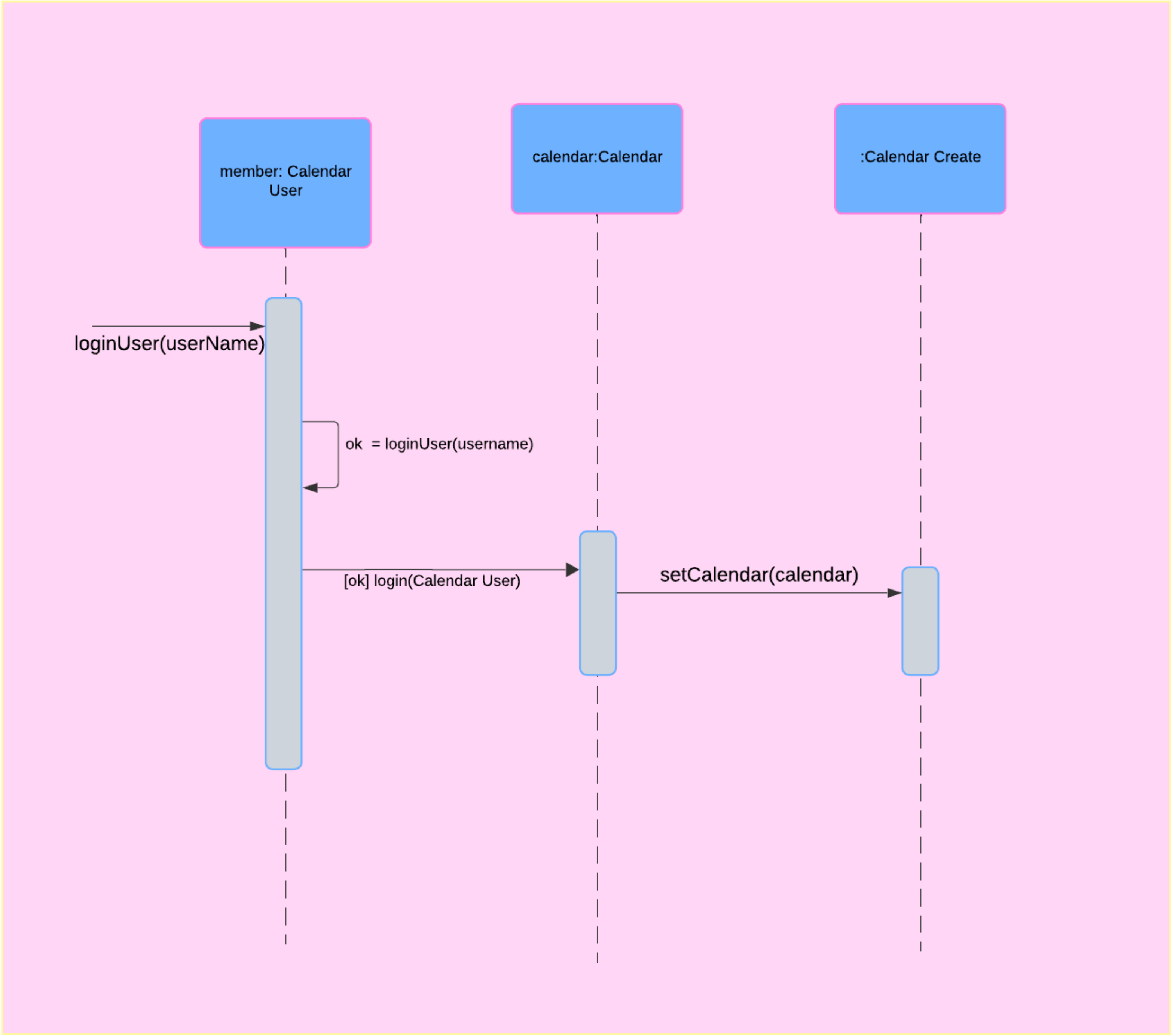
Configuration Screen

+ currentTimeZone: String
+ theme: String

- setTimeZone (newZone): void
- setTheme (String) : void

Gregorian Calendar

Calendars Sequence Diagram: Creating a Calendar



UML Class Diagram

- User:
    - Variables:
        - A list of Calendars to allow for the support of multiple calendars
        - A username per user to login
        - Each user will have a time-zone associated with them. By default, the calendars use UTC+8 (coordinated universal time), but if the string ‘PST’ was passed in, the time would be edited by -8 hours.
    - Functions:
        - loginUser takes in a username and logs the user in
        - createUser takes in a username and creates a new User with that name, also logging in the user
        - removeCalendar takes a calendar and deletes its data
        - createCalendar takes a CalendarName and it’s public status, creating a basic calendar with no events created.
        - UpdateCalendar is used to update the CalendarName or its public status
- Calendar:
    - Variables:
        - A list of events to allow for multiple vents
        - isPublic is a boolean (false = private, true = public). Public calendars can be seen by all users, private calendars are only seen by those with access to it. Private calendars can still be shared with other users.
        - calendarName is the calendar’s identifier
        - Owner is always the user that created the calendar
        - sharedWith is a list of Users that have access to the calendar
        - currentDay and Month are strings that are used when displaying the calendar. They use the device date
    - Functions:
        - shareCalendar takes a userName and gives those users access to the calendar
        - removeShare takes a userName and revokes their access to the calendar
        - removeEvent takes an eventName and deletes the event
        - createEvent initializes an event with its data variables (name, startTime, endTime, description) and returns an event to the calendar
        - updateEvent takes an existing event and updates a specific variable. It can only change one variable at a time currently
        - displayCalendar will generate a command-line representation of the preferred display. Possible inputs are day, week, month, year
- Event:
    - Variables:
        - Name: The event name
        - Description: A written description of the event

        - startTime & endTime: by importing java.time.*, we’ll use the ZoneDateTime to properly take in military time (ex. 13:40)
        - sharedWith is a list of users that have access to this event. Even though calendars already have a sharing function, I thought that events should be able to be shared individually as well.
    - Functions:
        - changeDescription is used to change the description completely
        - shareEvent works the same as calendar’s implementation
        - removeShare works the same as callender's implementation
        - changeTime(startTime, endTime) allows for editing of the start and end time
- Configuration Screen
    - Variables:
        - currentTimeZone is the time zone that is used by that specific calendar. This exist in-case other users from other time zones have access to the calendar and need to change it for their personal preferences
        - Theme is a string that will change the color of the calendar.
    - Functions:
        - setTimeZone allows for a change to the calendar’s time zone
        - setTheme allows for changes to the calendar’s theme

Relationships:

- A user can have many calendars, and a calendar can have many users
- A calendar can have many events, but an event only belongs to one calendar
- Each calendar has a configuration screen associated with it
- Calendars support the gregorian calendar format

Future Implementation Considerations

This Calendars design allows for the easy implementation of future changes 1, 7, and 10
A GUI implementation has operations that can be directly mapped to actions (logging users, creating/deleting events, updating events). The only function that might become redundant is displayCalendar(), since the GUI would display the calendar completely differently.
For 7, a notification class can easily be created with relations to both Calendars and Events. It can have the message with user-specific details and notify recipients when a shared event is updated. It’ll be easy to link the calendars or events as well in the notification. As for expanding settings/configurations, the configurationsScreen class already provides a basis for users to edit their calendars. Future additions to the screen could be language preferences or event reminder times.

With 5, calendars would have a png/jpg file associated with them that the user can upload.
I didn’t realize it but I had implemented 9 on accident, so that was definitely easy to implement!