

COSC 112: PROGRAM SPECIFICATION

Calendar App – Dasha Asienga, Priya Bakshi, and Lynca Kaminka

Describe your program at a high level

Our program is a calendar application that allows users to keep track of activities and duties that they need to do. Their information is stored in a file, such that every time the user launches the app, they are able to retrieve any items or information that they had added previously as well as add new ones. They are also able to delete entries. Because they are able to access previous entries, our calendar app also includes a “to-do-list” feature and a “reminder” feature for activities that are coming up. They are able to select how they view the calendar, that is, on a daily or monthly template. We allow different users to use the calendar as well as change their calendar display (theme).

What will your program look like in the end?

When they launch the app, the user will view the sign in/ sign up page. The sign up page allows users to create an account and the sign in page allows existing users to log into their account. Upon logging in, a user is taken to the choice page where they can choose to see the calendar month view, their running to-do list, or their reminders for the day. In the calendar, they can press on any day and view any events that they had inputted as well as add, edit, or delete events. They are able to move back or forward any month and day. The to-do list feature allows users to add activities or tasks that they need to complete (which can be deleted once completed). The reminder feature lists a numbered list of the upcoming events from the calendar for the current day.

How will the user input work?

User input is primarily through mouse and keyboard. The mouse feature allows them to click on anything in the app (e.g, click on a day or a functionality) , and the keyboard input allows them to type in where prompted (e.g, add something to the to-do list or add something to the calendar).

How will the program respond?

The output from the program is updated immediately.

What purpose does it serve?

It’s a productivity tool that is useful in day-to-day life. We wanted to have practice creating something practical.

Provide a list of the classes you will define.

- *Calendar* – this class holds the primary structure of the monthview and the dayview of the calendar.

- *CalendarFramework* – this class defines the main calendar view (month view). It allows a user to view what day it is, scroll through any month, and click onto a specific day of interest.
- *DayView* – this class defines the day-view on the calendar. It writes to file when a user adds an event and reads from file to populate any existing events.
- *ChoicePage* - this class allows the user to choose which functionality they want to make use of: the calendarApp, the ReminderApp, or the ToDoLists app
- *AppFrame* – this class contains the ToDoLists framework. It allows the user to enter an activity on their to-do list, remove it, or check it off.
- *ReminderFrame* – this class populates an ordered, numbered list of all the activities of the day.
- *Main* – this class launches the app by opening the Sign in or Sign up window when the app is run.
- *LoginPage* – this class creates the database for the app founder, which is supposed to contain all users' usernames and passwords, and allow users to sign in. In case, it's their first time using the app, they will have to sign up, and their information will still be saved.
- *IDsandPasswords* – is a class whose objects will be used as the arguments for the constructor of the LoginPage.
- *ThemeFrame* – allows users to choose a theme for their calendar.
- *Frame* – this interface is implemented by all the frames in the app: the month view, the day view, the to-do list app, and the reminders app.

Notes (7 of 8 of the required components completed):

1. We use Java Calendar and Time libraries to access time information.
 - a. We also used an external Java package to help implement the calendar app.
 - i. This package took us a long time to understand.
 - b. We implemented a lot of action-listeners.
2. We created a *Calendar* class that is extended by both *CalendarFramework* and *DayView*.
3. We created a *frame* interface that is implemented by all the frames in the app: the month view, the day view, the to-do list app, and the reminders app.
4. We use numerous built-in data structures in virtually every class, primarily linked lists and hashmaps.
5. All the data structures we used are generic.
6. We use file input and file output to save information from the calendar and load them onto the calendar. It's vital to the success of the program.
7. We use randomization to randomize a message when the user clicks on the sign in button.

Describe the semantics and use of the class. What does it represent? When your program is run, does one instance exist? or a few? or many?

For one user, there will only be one instance of every class created that will load all the information for that one user.