Calender App

Jeremy Gordon
Kyle Byassee

Abstract

Our group project is a calender application that runs in the background. Some of the functions are notifying the user of events, importing and exporting user profiles and customizing the user experience with pictures. The target audience are users that want more freedom from their calendar application.

1. Introduction

Our group project is a calender application it will be built in a C# WPF application that runs in the background. Some of the functions are notifying the user of events, view historic information and customizing the user experience with pictures. We are targeting individuals who want the ability to customize, export, import and set specific day/month/year/time notifications. We want to give the user a calendar application that better then what is offered currently.

1.1. Background

We decided on this project because we do not have much experience in c#. We also felt we could improve on the traditional calender application providing additional freedom and features. Providing features like viewing historic information and customizing backgrounds with user pictures. A calender application sounded like a project that could be fun, challenging and a worthwhile semester long project.

1.2. Impacts

The project is geared towards a general audience. Our application is appealing to that audience because of the ability to view historic information, customize the look of the applications background, set custom notifications. It may have a small but positive effect on the end user helping manage and organize life events.

1.3. Challenges

The layout of the GUI seems daunting since some months have more days than others. We plan to take inspiration from the numerous calender application that are already in the wild, Improving on the GUI layout making it more user friendly and intuitive.

Adjusting the GUI for events like leap years. - We plan on using already built time/date functions to account for this.

2. Scope

The app will display the current date and time when first opened. Let the user change the date and set a notification on that date at a specific time. The user should be able to set the background of the app to a designated picture format. The app should have important holidays and dates displayed.

2.0.1. Stretch-Goals. Export a user profile to be imported later into the same or different calender application. Adding weather to be displayed in the GUI of the application.

2.1. Requirements

The functional and non-functional requirements were gathered based upon what the program would need to be considered complete.

2.1.1. Functional.

- User needs to be able to specify a specific day/month/year to add a notification.
- User can change background of the calendar with a picture.
- User can change and navigate through the month, year and day.
- User can cancel or modify an existing notification.

Use Case ID	Use Case Name	Primary Actor	Complexity	Priority
1	Set a notification	User	Med	1
2	remove or change a set notification	User	Med	1
3	Change calender background	User	Low	1
4	View historic information	User	Low	1

TABLE 1. SAMPLE USE CASE TABLE

2.1.2. Non-Functional.

- User saved notifications should be encoded to protect user data.

2.2. Use Cases

1.

Use Case Number: 1-2
Use Case Name: Set a notification
Description: The user has a specific date and time they with for a notification to appear on. The user navigates to the desired date. Selects the date and then specifies a time where the program saves that data and creates a notification.

You will then go on to (minimally) discuss a basic flow for the process:

1) User navigates to desired date.
2) User selects the date.
3) User specifies a time on that date.

Termination Outcome: The user has a notification set on that time and day.

Alternative: Notification has already been created

1) User navigates to the same date
2) User selects the date
3) User selects the same time
4) User chooses to cancel or modify existing notification

Termination Outcome: The notification has been modified or canceled

Use Case Number: 3
Use Case Name: Change calender background
Description: The user desires to change the background of the calender

1) User clicks on a button to upload an image.
2) User selects the image.
3) User confirms the selected image.

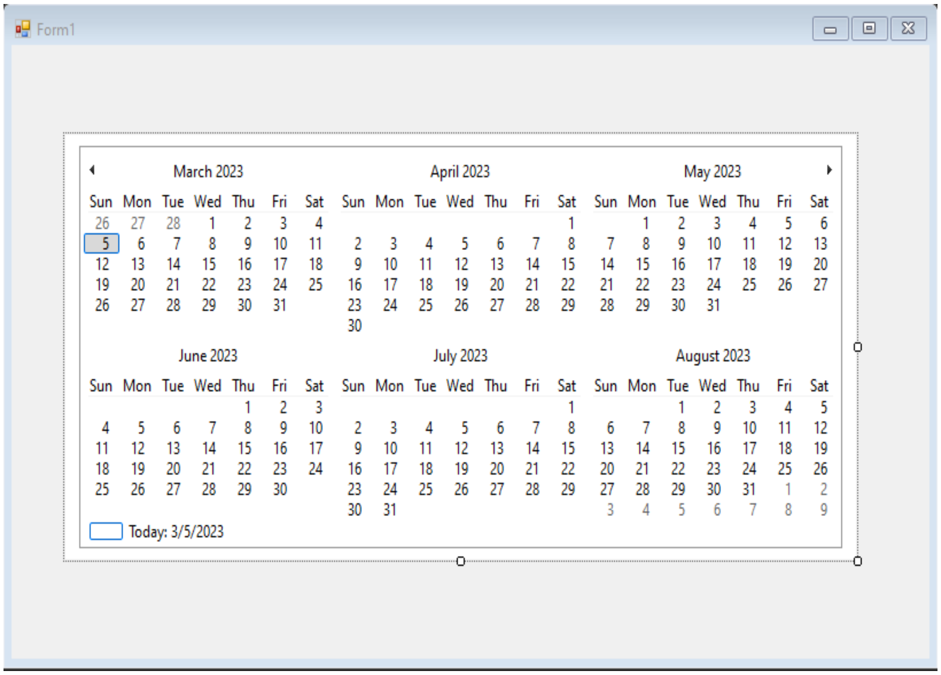
Termination Outcome: The user has change the background of the calender.

Use Case Number: 4
Use Case Name: View historic information
Description: A user desires to view specific historic information on a date.

1) User navigates to a specific date.
2) User selects the specific date.

Termination Outcome: The historic information is displayed to the user.

2.3. Interface Mockups



March 2023

Sun	Mon	Tue	Wed	Thu	Fri	Sat
26	27	28	1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	

April 2023

Sun	Mon	Tue	Wed	Thu	Fri	Sat
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30						

May 2023

Sun	Mon	Tue	Wed	Thu	Fri	Sat
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

June 2023

Sun	Mon	Tue	Wed	Thu	Fri	Sat
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	

July 2023

Sun	Mon	Tue	Wed	Thu	Fri	Sat
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31					

August 2023

Sun	Mon	Tue	Wed	Thu	Fri	Sat
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31	1	2
3	4	5	6	7	8	9

Mockup of the GUI of our calender application

3. Project Timeline

Go back to your notes and look up a typical project development life cycle for the Waterfall approach. How will you follow this life cycle over the remainder of this semester? This will usually involve a chart showing your proposed timeline, with specific milestones plotted out. Make sure you have deliverable dates from the course schedule listed, with a plan to meet them (NOTE: these are generally optimistic deadlines).

4. Project Structure

At first, this will be a little empty (it will need to be filled in by the time you turn in your final report). This is your chance to discuss all of your design decisions (consider this the README's big brother).

4.1. UML Outline

Show the full structure of your program. Make sure to keep on updating this section as your project evolves (you often start out with one plan, but end up modifying things as you move along). As a note, while Dia fails miserably at generating pdfs (probably my fault), I have had much success with png files. Make sure to wrap your images in a `figure` environment, and to reference with the `ref` command. For example, see Figure 1.

4.2. Design Patterns Used

Make sure to actually use at least 2 design patterns from this class. This is not normally part of such documentation, but largely just specific to this class – I want to see you use the patterns!

5. Results

This section will start out a little vague, but it should grow as your project evolves. With each deliverable you hand in, give me a final summary of where your project stands. By the end, this should be a reflective section discussing how many of your original goals you managed to attain/how many desired use cases you implemented/how many extra features you added.

Figure 1. Your figures should be in the *figure* environment, and have captions. Should also be of diagrams pertaining to your project, not random internet kittens

5.1. Future Work

Where are you going next with your project? For early deliverables, what are your next steps? (HINT: you will typically want to look back at your timeline and evaluate: did you meet your expected goals? Are you ahead of schedule? Did you decide to shift gears and implement a new feature?) By the end, what do you plan on doing with this project? Will you try to sell it? Set it on fire? Link to it on your resume and forget it exists?

References

[1] H. Kopka and P. W. Daly, *A Guide to LaTeX*, 3rd ed. Harlow, England: Addison-Wesley, 1999.