WASHINGTON STATE UNIVERSITY
VANCOUVER

CS 320 Course Project Final Report

for

Calendar App

Prepared by

Group Name: Team Pi

Caylin Seitz	<011674816>	caylin.seitz@wsu.edu
Ramsay Fencl	<011606348>	ramsay.fencl@wsu.edu

Date: December 16, 2020

1 Introduction

1.1 Project Overview

The calendar application allows for the user to manage an itinerary of events, including adding events that contain details and time information. Events can be deleted but not altered due to time constraints, as well as only being able to be deleted one at a time. The application stores all the user's events within the browser's local storage, meaning that the calendar can be closed and opened at any time while maintaining the saved information. This means that the only way to fully reset the calendar would be to clear the local storage of the page from the browser's settings, this would be a feature in the app if not again due to time constraints.

Users who struggle with time management would greatly benefit from this application. This application is meant to help users organize their time by days in a month. The user can put in their workload and other obligations for the week to have a list of events to follow. In the application they can add events with detailed information such as a description, event color, and time, the user can also view the list of events on a given day by simply selecting the day, events can be deleted one at a time for a given day by clicking on the delete button for the given event.

The calendar presents the current day with bolded text, days with events on them with the most recent event color, and selected days with a highlight. The calendar information is stored within the browser's local files.

1.2 Definitions, Acronyms and Abbreviations

TT: Time Taken
GUI: Graphical User Interface

1.3 References and Acknowledgments

https://developer.mozilla.org/en-US/docs/Web/javascript, resource for JavaScript functionality.

https://www.w3schools.com/, resource for html and CSS usage.

2 Design

2.1 System Modeling

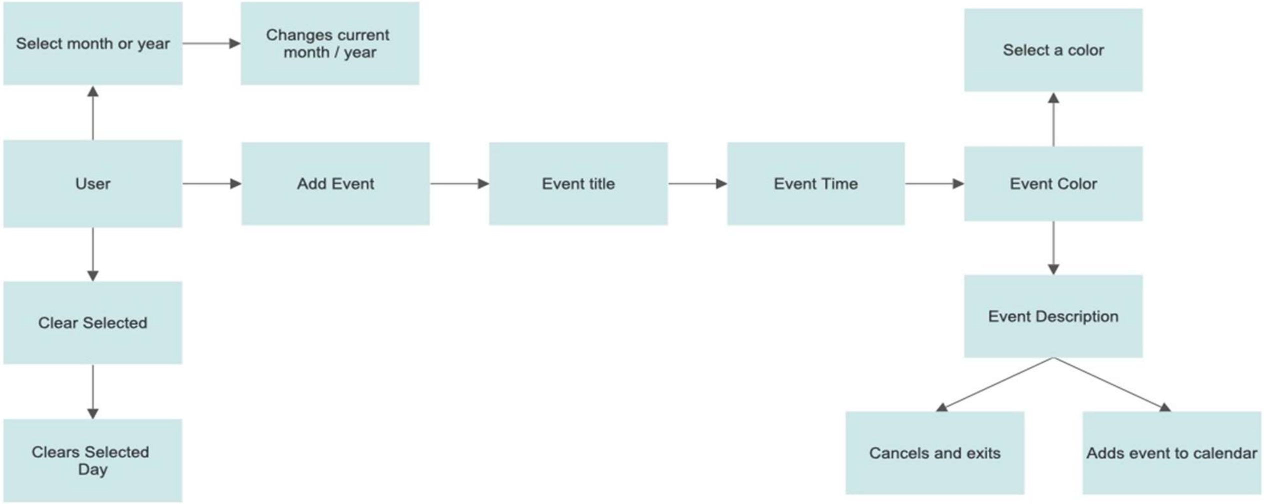


Figure 1:Sequence diagram

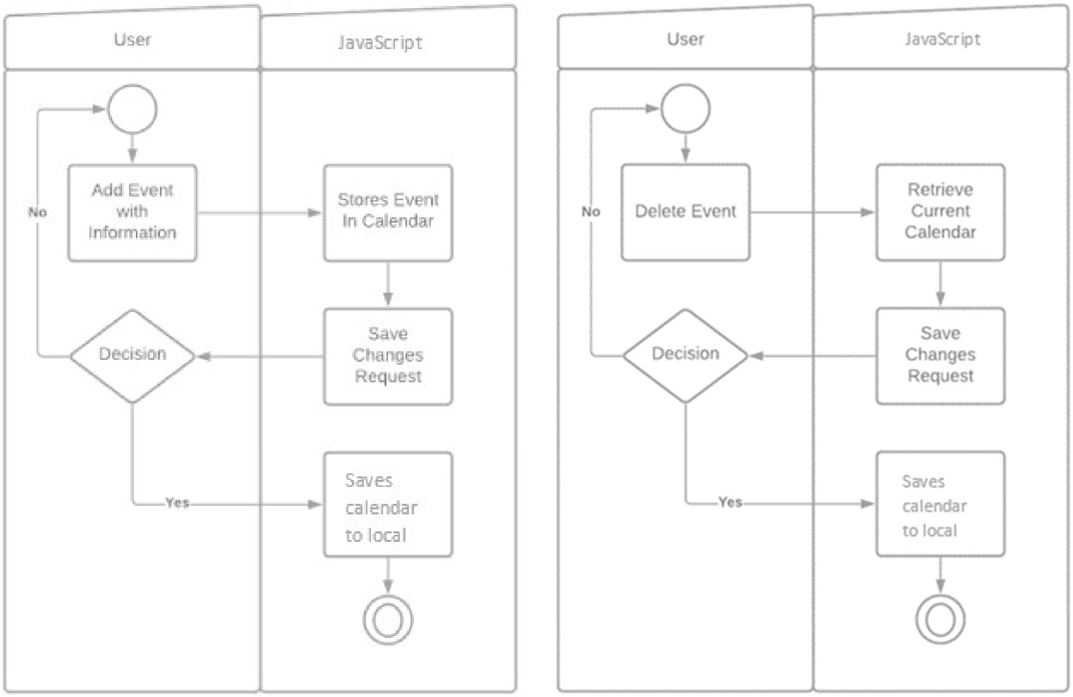


Figure 2: Activity Diagrams

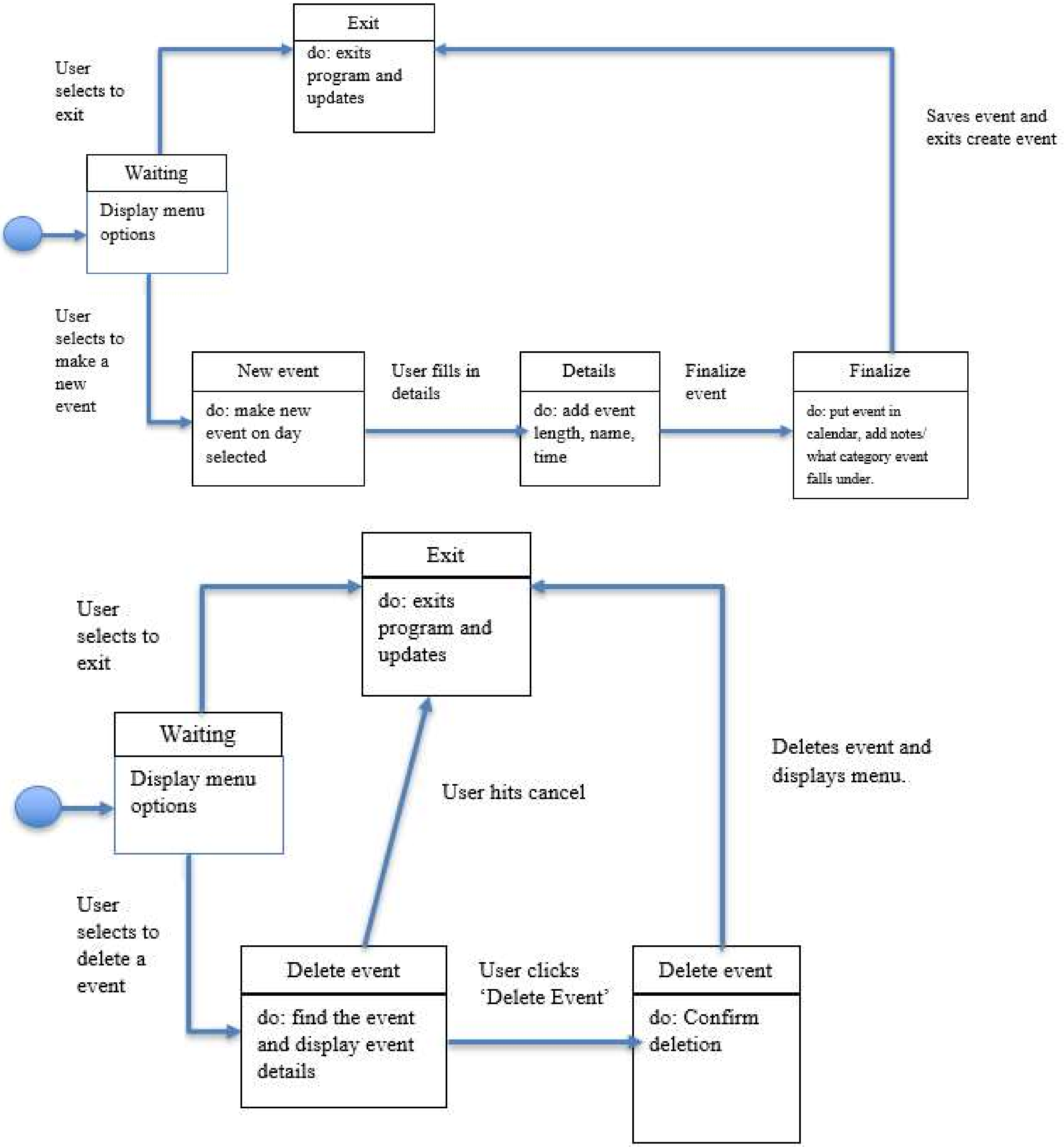
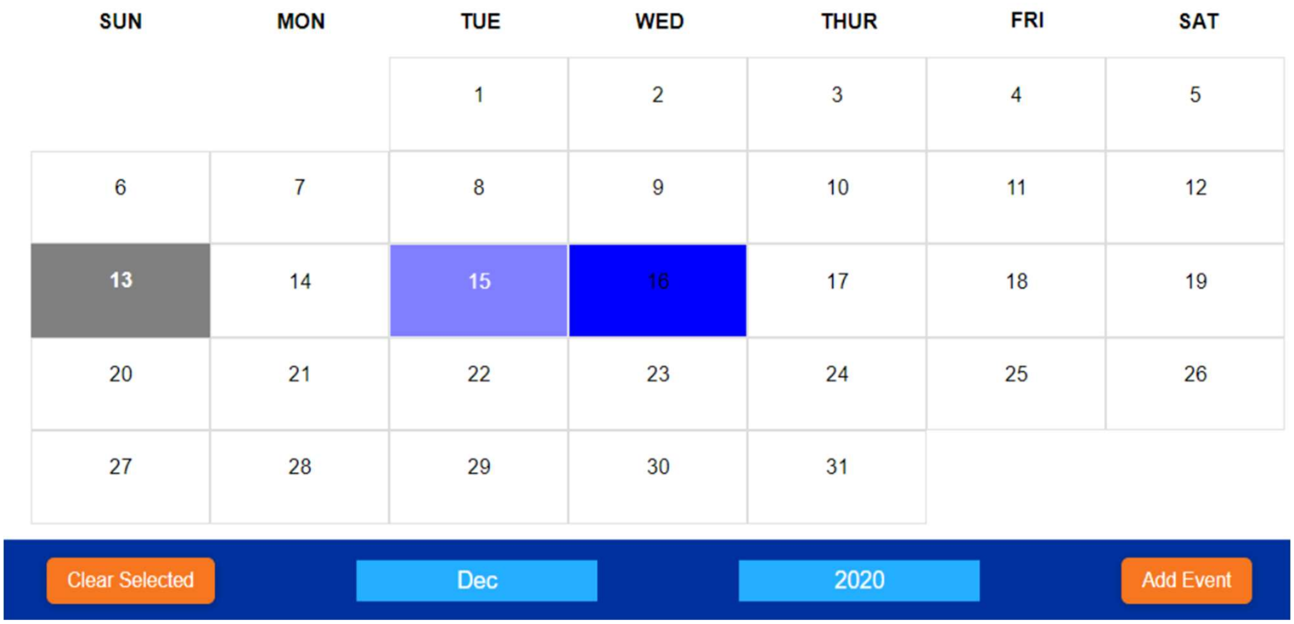

Exit
do: exits program and updates
User selects to exit
Saves event and exits create event
Waiting
Display menu options
User selects to make a new event
New event
do: make new event on day selected
User fills in details
Details
do: add event length, name, time
Finalize event
Finalize
do: put event in calendar, add notes/ what category event falls under.
Exit
do: exits program and updates
User selects to exit
Waiting
Display menu options
Deletes event and displays menu.
User hits cancel
User selects to delete a event
Delete event
do: find the event and display event details
User clicks 'Delete Event'
Delete event
do: Confirm deletion

Figure 3: Behavioral Diagrams

2.2 Interface Design

The interface was developed without any prior planning, and developed as needed to accommodate functionality on the javascript end.



SUN	MON	TUE	WED	THUR	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		

3 Implementation

3.1 Development Environment

This project was developed on Visual Studio Code and tested within the Microsoft Edge browser. The languages used were HTML, JavaScript, and CSS.

3.2 Task Distribution

Ramsay: Developed app and tested app, reviewed documentation and finished final report.

Caylin: Documented “progress” and wrote the initial draft of the final report.

Logan Tan (Honorary member): Helped update the SRS to have the project complete in time.

3.3 Challenges

All of the “smart” functionality had to be cut due to time and manpower constraints. There was also a team member who did not participate at all in the project and the whole project had to be done by two people, with additional help from a friend at the last minute.

4 Testing

4.1 Testing Plan

All testing was done within the browser by stepping through the code using JavaScript's console log function and debugger function call, as the basic front end was developed first, the use cases were tested and developed in accordance to what interactions the user could take on the front end of the application.

- Make sure days are selected and added to the SelDays array when clicked
- Make sure all days within SelDays have unique events added to Events array with the same id on event creation
- Make sure that events are loaded back onto display when swapping months
- Make sure that the Events array is saved to local on event creation and deletion
- Make sure that the Events array is loaded from local on page load.

4.2 Tests for Functional Requirements

When testing the buttons on the app all worked. When pressed all the listeners responded correctly. The month and year buttons also worked correctly, and you can select a different month and year. When the month and year changes the weeks adjust correctly. Testing for events adding correctly, asserted that a date matched the date to the created event date, as well as the event id's within the DOM's matched those within the Events array.

All functions were tested as they were added, to ensure that there were no complications with previously written code.

4.3 Tests for Non-functional Requirements

Testing for the following non-functional requirements was done as functionality was added:

- Same event can be added in multiple days
- When events have the same id (due to previous nonfunctional req), event deletion only occurs for the selected day.
- Event Id's are stored within the DOM so that when being displayed by the event list, there are no duplicates

4.4 Hardware and Software Requirements

Hardware: No requirements other than any computer that can run HTML5 browsers. While the application does load on mobile devices, it is not recommended since formatting was done for a minimum of a 600x500 pixel screen.

Software: Microsoft Edge

5 Analysis

Milestone 1:

Ramsay: 9 hours
Caylin: 4 hours

Milestone 2:

Ramsay: 8 hours
Caylin: 5 hours

Milestone 3:
Ramsay: 24 hours coding, 5 hours documentation, 29 hours total
Caylin: 4 hours documentation
Logan Tan (honorary member): 1 hour of documentation

Milestone 3 took the most time to complete. Caylin accidently corrupted the file she was using to create the calendar and start over. Luckily, Ramsay started a new version and completed the project. Some features were left out due to this time and manpower constraint and the database requirement was left out as well since the project requirements changed.

6 Conclusion

The team (hopefully) has learned about team management on group projects, as most of the work was left in the hands of a single individual for the entire project. It is also best to seek the help of higher ups such as professors earlier in the project to potentially find a solution, when meekly asking for participation on group members own volition fails.

Besides, group struggles, time management was also used improperly. This can be attributed to life distractions and other courses obligations for each team member. However, it is also important to not be discouraged due to circumstances such as a lack of other members participation. Ramsay learned this personally, as if he had done the project from the start without hoping for the full help and participation of the group, the full feature set could have been implemented in time. Though this is not held against said individuals in any way.

I (Ramsay) also learned the value of true friendship, as in your time of need they will be there to help you complete your documentation in time. (Shout out to Logan Tan).

Appendix A - Group Log

Meetings were sparse, as it was difficult for the group to decide on dates to meet when the team lead (Ramsay) would ping them within the discord server. This was often due to no one responding.

First meeting notes 10/21/2020

Specs-

- Users can set blackout times on their calendar
    - Set events that conflict with blackout times are given +1 priority
        - This is so assignments can be completed before a work shift/class/etc...
- Tracks set events
    - Events can be categorized by class/subject
    - Events can be created/deleted/marked as complete
    - Events will have priority levels
    - Events that conflict will be marked
- Creates an itinerary
    - Organizes events based on importance/priority/completion time
    - Schedules around blackout times
- Calculates completion times
    - Upon completing an event, prompts for time spent
    - Takes average times for event type/subject into account
        - Math assignments may be quicker than physics, etc...

Second meeting notes 11/30/2020

Specs –

Changed the scope of the project due to new time constraints.

Third meeting notes 12/07/2020

Specs –

Decided on Time of Presentation