Daniel Beeman, Holly Hardin, Linshu Huang, Yiming Ni, Maxwell Logan
Professor Hornof
CIS 422
1 February 2019

Initial Project Plan / SRS / SDS

Project link: http://ix.cs.uoregon.edu/~dbeeman/422/CIS422_G5Calendar/displayevents.php

1 Project Plan

1.1 Management Plan

1.1.1 Team Organization

The team will be organized by assigning team members to focus on certain artifacts. In particular, the artifacts and its contributors will be:

- Designing the software: Daniel, Holly, and Linshu
- Writing the code: Daniel, Yiming, Linshu, Holly
- Creating the documentation: Holly, Max, Daniel, Yiming
- Providing reviews: Yiming, Daniel, Max
- Testing results: Daniel, Yiming, Max, Holly, Linshu

1.1.2 Team Meetings

- Team meetings began on Monday's and Tuesday's, but were moved to Saturday's and Tuesday's to allow space between each meeting for individual members to work on their own.
- If a team member cannot attend the meeting, he or she shall communicate the absence to the team in-person or through Discord. He or she will be updated via Discord of the meeting's contents.

1.1.3 Communication

- All members are expected to contribute to discussions at team meetings. Also, members are encouraged to communicate through the chat application Discord, concerning the project, especially when they cannot attend a meeting in-person. Communication with the professor takes place through email and in-person discussions.

1.1.4 Decision-making Process

- Members should articulate their ideas about how the project should proceed at team meetings and through Discord. Thus, different options concerning the project will be considered and discussed. The majority vote of the best option will conclude decisions. For example, in coding meetings, a design question will arise, and the party members at the meeting will communicate, with the majority decision being implemented.

1.2 Schedules

1.2.1 Project Breakdown Schedule

L#	Description	Team Member Assigned	Due Date	Completed Date
L-1	Initial Project Plan Documentation	Holly	01/18/2019	01/18/2019
	IPP Review	Max, Daniel, Yiming	01/23/2019	01/23/2019
L0	Display Calendar	Holly, Daniel	01/22/2019	01/22/2019
	Add Event	Holly, Yiming	01/25/2019	01/23/2019
	Display Event	Holly [Linshu, Daniel]	01/27/2019	01/24/2019
	Performance Review	Holly	01/27/2019	01/26/2019
L1	Save data to database	Holly, Daniel [Linshu, Yiming]	01/27/2019	1/31/2019
	Add/Edit/Delete Event	Linshu, Holly, Daniel, Yiming	01/27/2019	1/31/2019
	Display Event	Holly [Linshu, Daniel, Yiming]	01/27/2019	1/26/2019
	Work for all days in 2018-2020	Holly [Daniel, Yiming]	01/27/2019	01/25/2019
L1.5	Update Tech. Doc	Max, Daniel, Yiming	02/01/2019	01/31/2019
	Unit Testing	Max	02/02/2019	01/31/2019
	Performance Review	Max	02/02/2019	01/31/2019

	Update User Doc	Max	02/03/2019	01/31/2019
L2	Add Categories	Linshu	02/01/2019	02/01/2019
	Add Priority	Linshu	02/01/2019	02/01/2019
L2.5	Update User Doc	Max	02/03/2019	01/31/2019
	Update Tech. Doc	Max	02/03/2019	01/31/2019
	Performance Review	Max	02/03/2019	
	Unit Testing	Max	02/03/2019	
	Put Together Presentation	Max, Team	02/05/2019	02/02/2019
Project Due: 10:00 PM February 5th, 2019				

1.3 Monitoring and Reporting

1.3.1 Tracking Progress

- Team members are to monitor and report their progress through the established document “CIS 422 Individual Work Documentation.” This Google document has been shared with all users via Gmail, and the following information shall be recorded:
    - Name
    - Date
    - Total time spent
    - What was worked on
    - Notes (optional)

1.4 Build Plan

1.4.1 Sequence of Steps

The steps for building the system will be accomplished in the order of:

1. Creating an HTML page that displays visual components of a monthly calendar.
2. Updating the HTML page to be interactive through popups (adding an event, editing an event, and deleting an event).
3. Modifying the HTML page to synchronize data with SQL with respect to the addition, modification, or deletion of an event.
4. Additional features including categories for events and priorities for events.

1.5 Rationale for Build Plan

1.5.1 Reasoning

- The rationale for the build plan logically flows because each step builds a major component onto a previous step and improves usability.
    - The first step must be accomplished first because it will give the user a visual display. In doing so, the programmer will be able to identify where in the code the popups are to be implemented. This is also the easiest component to setup, making it an appropriate starting place.
    - The second step must be completed because it will extract data from the user according to the event. By doing this, the data will be available to be sent to SQL. This step is in preparation for displaying events on the calendar, generated by the user.
    - The third step gives total functionality to the system by providing a calendar that is up-to-date for the user.
    - Lastly, with a working calendar, additional features are included to provide a better utility to the user. This is the last step because these features are not mandatory in a robust and usable calendar, but improve the experience for the user.

1.5.2 Risks and Risk-Reduction Strategies

- As with any project, risks can arise. According to Writer (2010), a few of the software development risks that can occur are:
    - Estimating and scheduling
    - Sudden growth in requirements
    - Productivity issues
- The risk of estimating and scheduling can be avoided by considering milestones and considering the *realistic* amount of time required for completion. In doing so, deadlines can be considered and formed based off of the milestone. The need for additional features can also put added pressure on the group. We avoid this by communicating what additional features everyone is comfortable with and making sure we are all capable to include such features.
- Lastly, productivity issues can arise. If a task seems easy, procrastination may occur, which may make it difficult to meet the deadline for a milestone. This risk will be circumvented because we have created many milestones, each of which should be moderately easy to accomplish on time. Each team member is responsible to monitor and contribute, ensuring our project's success.

2 Software Requirements Specification (SRS)

2.1 Problem Statement

2.1.1 The Problem

- Someone may find it difficult to keep track of all the things he or she has to do throughout a day. For some, forgetfulness may cause events or small tasks to not be fulfilled. Unmet commitments and disorganized scheduling are a problem.

2.1.2 The Solution

- The solution is to develop a calendar application that allows a user to add an event, edit an event, or delete an event on a local computer that will help them keep track of their schedule. This way, a user will be able to see all events they have scheduled for a given month in one simple view. If a user wants to attend an event or include something on their schedule, they can easily check our calendar application to verify their availability status for such an event.

2.2 Description of User

2.2.1 The Users

- The user can be anyone – students, managers, workers, etc. who needs to organize their schedule in the University of Oregon's Computer and Information Science department. The last part of the description is as such since the program runs on the ix.cs.uoregon.edu server.

2.2.2 User Expectations

- The user can expect an interface that displays a calendar that allows the user to interact with widgets (buttons and text fields).
- Additionally, the user can expect that event information that is added, edited, or deleted will be saved and updated if any changes are made.
- Extra features that the user will be able to take advantage of include categories for events, as well as priorities for such events.

2.2.3 Previous Knowledge

- The users should have basic computer skills such as interacting with and navigating a website. The user should also have the desire to plan for events and tasks.

2.2.4 Current Technology Usage Patterns

- Current technology usage patterns are similar – web applications that sync event information for an individual's account.

2.3 Use Cases

2.3.1 Scenarios

- A few scenarios for using a calendar (Rao, 2017) could be:
    1. A person who wants to set time aside for goals, such as setting time aside for meditation or pursuing a hobby.
    2. A person who wants to keep track of tasks and reminders, such as scheduling for paying bills or sending a follow-up email to someone.
    3. A person who wants to schedule meetings or appointments easily by being able to visualize time slots in a day.
    4. A person who wants to view a day either far in the future or past to determine if that day is a Monday, Tuesday, etc.

2.4 Requirements

2.4.1 Functional Requirements

- Absolutely Required:
    - Display a month's worth of events
    - Switch between various months and display each month's worth of events from 2018-2020
    - Add a new event
    - Edit an existing event
    - Delete an event
    - Transfer data from input text fields to the database when an event is added
    - Transfer data from input text fields to the database when an event is changed
    - Delete data from the database when an event is deleted
- Not Absolutely Required (Additional Features):
    - Permit users to define event categories such as "work" or "home"
    - Permit users to define priorities for events such as "low", "medium", or "high"

2.4.2 Non-functional Requirements

- A header that shows the name of the month being viewed
- A table that represents each day of the month being viewed
- A display of events that have been added (or edited), appearing on the day they are happening

2.4.3 Testing of Requirements

- **Tested Functional Requirements:**
    - Program correctly displays a month's worth of events.
    - All events from the years 2018-2020 can be displayed in a monthly format.
    - A user can add an event to the calendar and have it display on the specified day.
    - Data is able to transfer to and from the database for adding an event.
    - A user can edit an event on the calendar and have it display on the specified day.
    - Deletion occurs correctly.
    - Events correctly have a category attached to them
    - Events correctly have a priority attached to them

- **Tested Non-functional Requirements:**
    - A header displays the month being viewed
    - A table properly displays all days for a month, as well as days included in the table for dates just beyond and just before the current month. For example, March has 31 days and ends on a Sunday, which is the first entry in the last column for the month of March. Thus, April 1st through 6th are displayed on the table in a faded text style, since they are in the same row as March 31st, and since March 31st is the last day of March.
    - The calendar correctly displays any added events.

2.4.4 Known Bugs

- It is possible to access months from before January 2018, but they do not display correctly.
- It is possible to add an event that does not have a name, causing it not to appear on the calendar, but it appears in the database and is deletable.
- Going from some month in the future back to the DisplayCalendar.php file will cause user's calendar to either jump ahead to the month in the future or get stuck between a few months.
    - Quitting and restarting the browser will undo this bug.

3 Software Design Specification (SDS)

3.1 Description of Product

3.1.1 Externally Visible Behavior

Description:

Our product is going to be a calendar in which users can add, edit and delete events for their schedule. Users can use the internet browser to open and view the calendar. The calendar has "previous month" and "next month" buttons to let users go through all events in each month, from 2018 to 2020. The user can also view the previous month by pressing the next month, or previous month buttons. Additionally, when users click the events on the calendar, there will be convenient pop-ups to prompt users to edit or delete events. The page will also display events *if* there are any events saved, with different colors according to the classification level. An event that has been edited will reflect the changes or additions made in these widgets accordingly.

- The webpage will have the following buttons that details all possible actions on an event:
    - "Next Month"
    - "Previous Month"
    - "Add Event"
    - "Edit Event"
    - "Delete Event"

- If the user clicks the "Add event" or "Edit event" buttons, a list of options will appear which allows the user to:

    - Input the event's title
    - Add some descriptions for an event's detail
    - Set the starting date, starting time, ending date, ending date, and ending time
    - Set the priority of the event.
- When all details are set, user can click the "submit" button to finish adding and editing events.

3.2 Design Description

3.2.1 Major Components

The major components of the system are:

- The calendar display
- The database that stores event information
- The modules for Add, Edit, and Delete

The interactions of these components is outlined as such:

- The calendar display contains all of the information relevant to the user during a single month, and requests the most updated information from the database for that month
- The database stores all of the information relevant to the user
- The modules Add, Edit, and Delete create, modify, and destroy data in the database, then updates the calendar display

3.3 Subsystems

3.3.1 Static Model

- Figure 1.1 shows a static model of the major components and how they interact. The Graphical User Interface (GUI) first draws data from the database, then displays information to the user.
- The user can perform actions, which in turn updates the database. This cycle continues until the user closes the program.

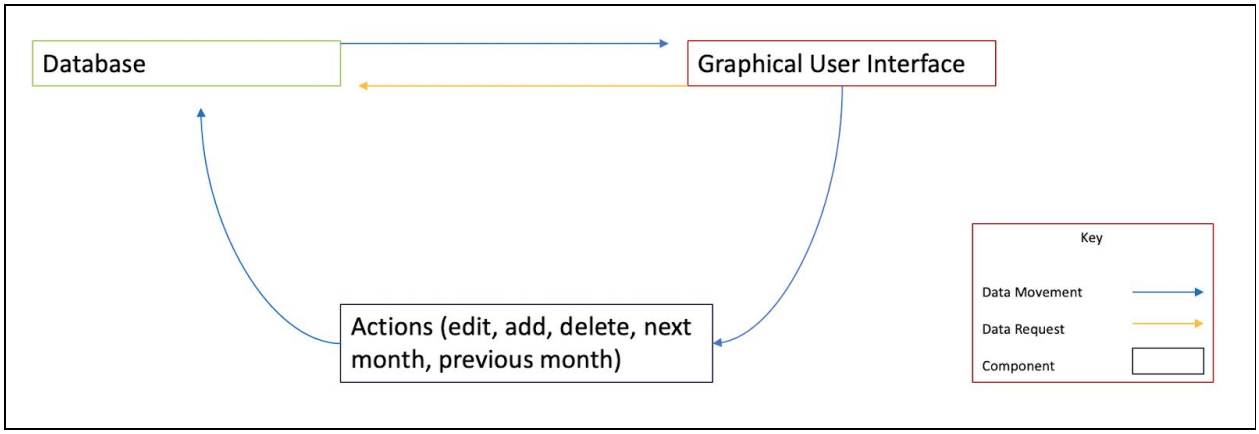

Figure 1.1: Static Model

3.3.2 Dynamic Models

- Figure 1.2 Displays information for a user when adding an event on the website.
- Figure 1.3 Displays the flow of steps taken when the user wants to edit an event.
- Figure 1.4 Displays a graph depicting the flow of steps taken when a user wants to delete an event.

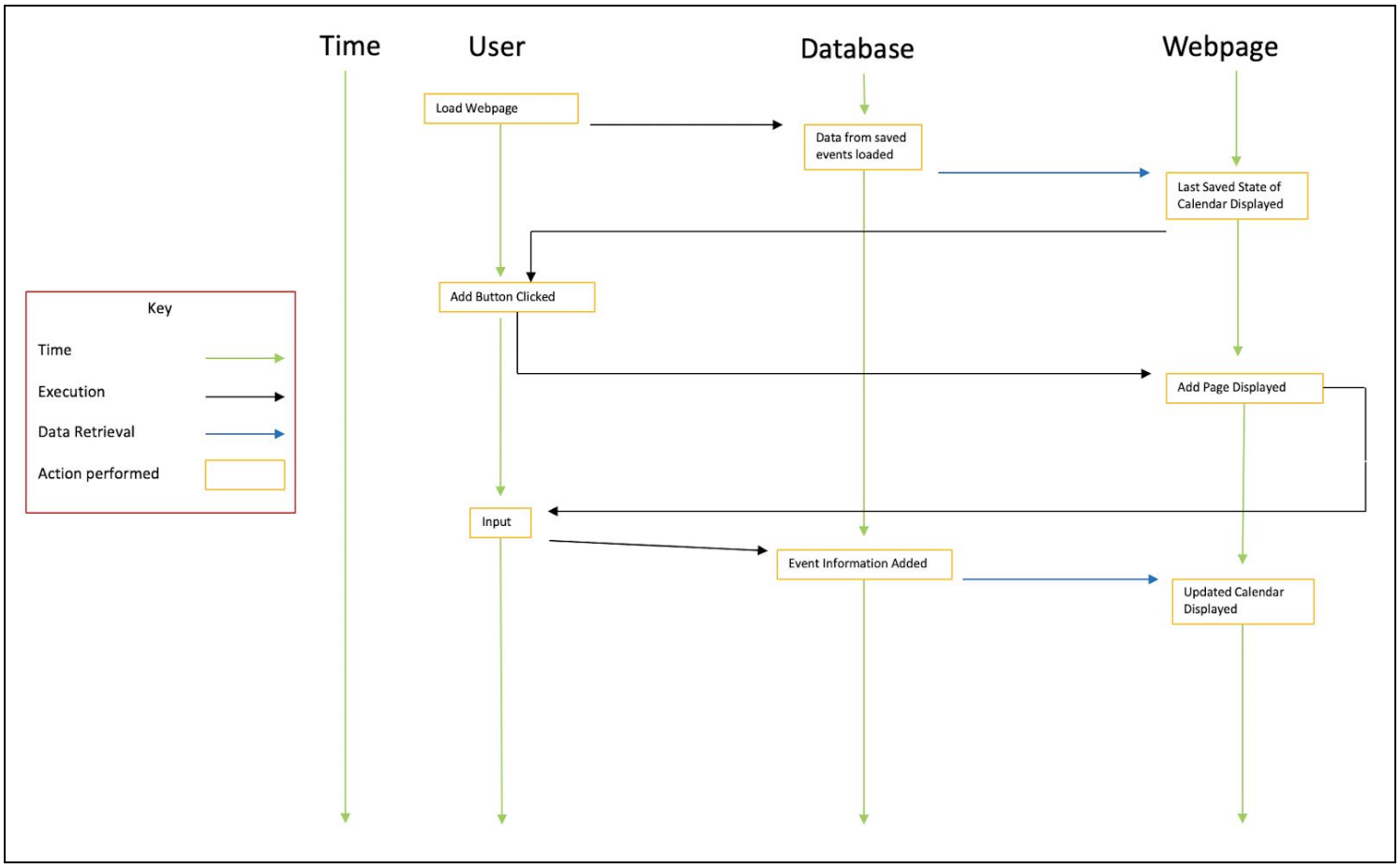

Figure 1.2 An event is added to the calendar

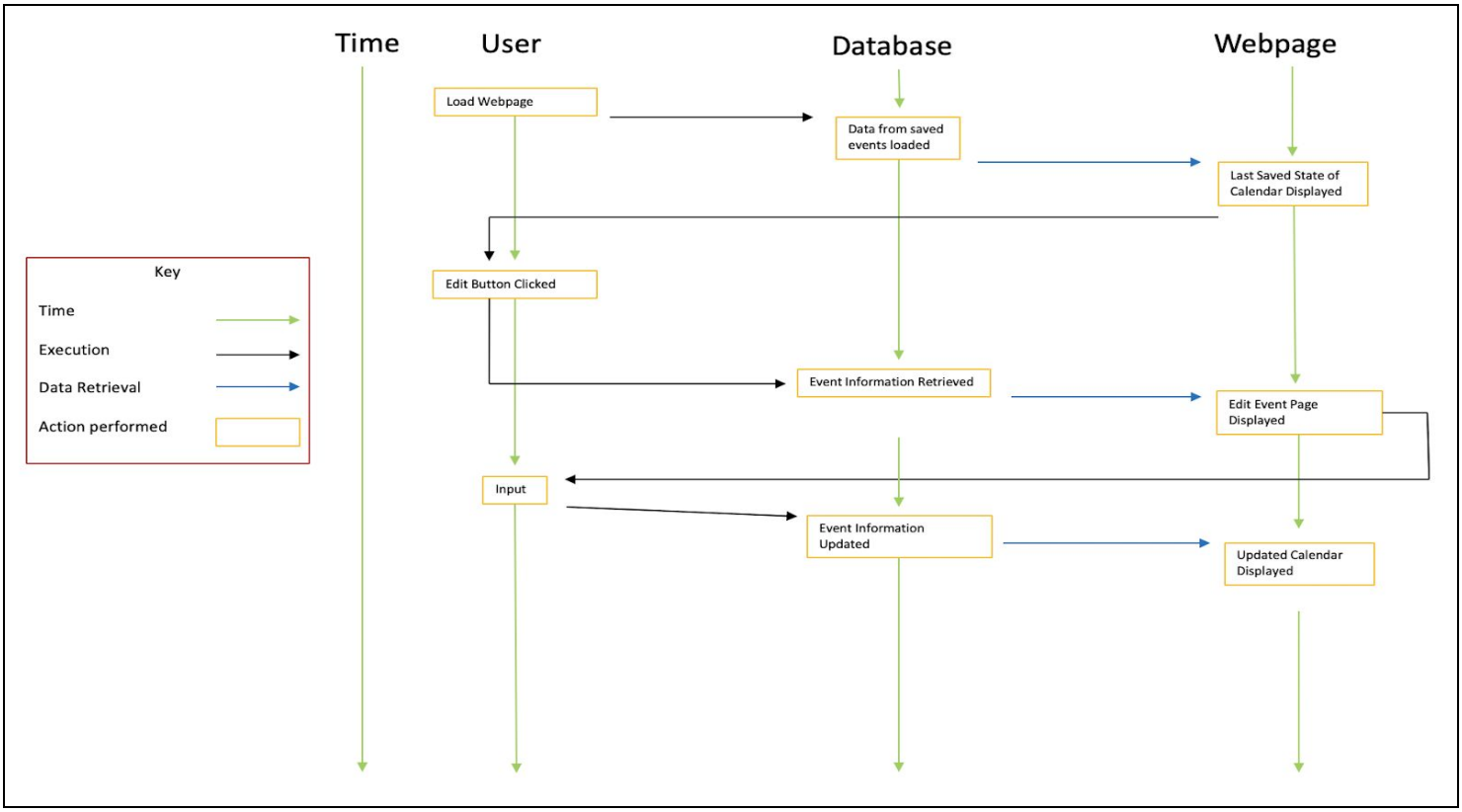

Figure 1.3 An event is edited on the calendar

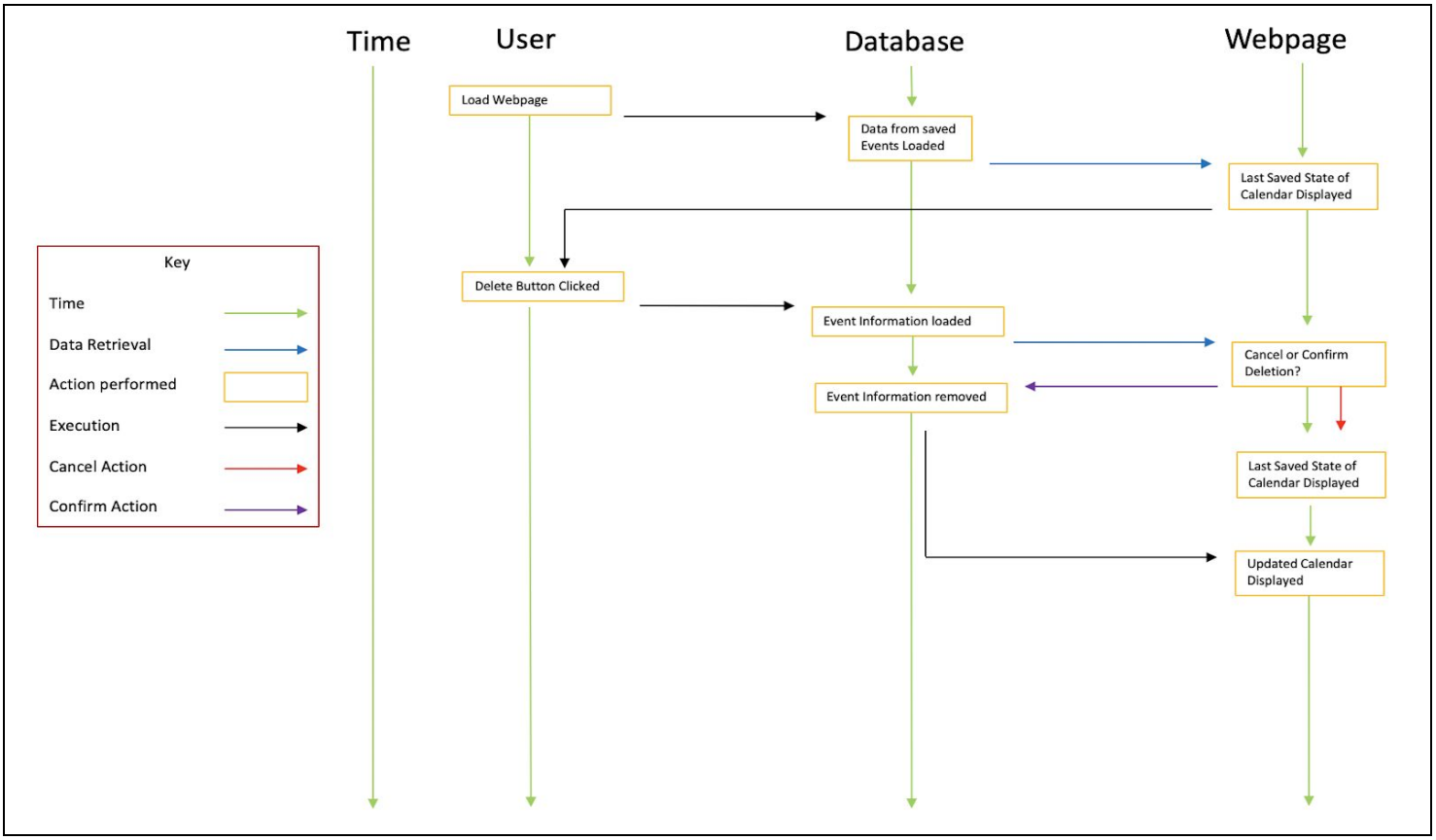

Figure 1.4 An event is deleted from the calendar

3.4 Design Rationale

3.4.1 Reasoning

- To some, Apple's calendar can be considered as easy-to-understand and easy-to-use. Thus, we chose this particular design to reflect the layout of Apple's calendar. Although we will not implement all of the capabilities of Apple's calendar, the functionality of adding, editing, and deleting an event should be similar.
- Our calendar is based on an iterative approach to implementing a calendar. We are starting with a very simple calendar, and expanding components and modules from there.
- Our group members have strengths in HTML, PHP, and MySQL, and therefore have decided to compose our program with those resources.
- Members implementing the database will be able to query results based on whatever input is put into the calendar by the user. These queries will be straightforward for the most part, since all that will be contained in the database is calendar event information. The complexity will be in sending data from the database to the front-end GUI.
- We are aiming for a robust calendar that will have simple and clear functionality, with functionality over extravagance, while still being elegant and easy to use.

4 Technical Documentation

4.1 Software Location and Dependencies

4.1.1 Location

- All source code is on Github at https://github.com/logangator2/CIS422_G5Calendar. Please follow the guide in section 5.1, titled "Installation" to install the software.
- It is also hosted on ix.cs.uoregon.edu.

4.1.2 Software Dependencies / Required Versions

- MySQL Workbench
- The University of Oregon ix.cs.uoregon.edu server

4.2 File Descriptions

- Each file below first connects to the ix server and establishes a database connection before doing any unique task.

--- Month Pages ---

4.2.1 NextMonth.php

- Displays the next month of events. This file gets reloaded if the user clicks the "Next Month" button twice (or more) in a row.

4.2.2 PreviousMonth.php

- Displays the previous month of events. This file gets reloaded if the user clicks the "Previous Month" button twice (or more) in a row.

--- Add Event Pages ---

4.2.3 AddEvent.php

- Displays a series of text fields and drop down menus for adding an event to the calendar. Works with getaddinfo.php to display drop-down menus.

4.2.4 getaddinfo.php

- Holds structure of user input for adding events. For example, it holds the drop-down box categories for choosing a starting month January-December.

4.2.5 AddEventToDatabase.php

- Adds information from user input to the database.
- Displays the updated calendar.

4.2.6 addquery.php

- Temporarily stores event information.

--- Edit Event Pages ---

4.2.7 ChooseEventToEdit.php

- Displays a drop-down menu to select which event to edit.

4.2.8 EditEvent.php

- Gets information for editing an event, displays current event information as well as a series of dropdown menus for updating the event. Uses geteditinfo.php to display dropdown menus for updated event information.

4.2.9 geteditinfo.php

- Largely similar to getaddinfo.php, displays dropdown menus for updating an event.

4.2.10 deletequery.php

- Removes the old event from the database. The updated event is treated as a new event, so the old one must be removed. This file does that.

4.2.11 EditEventToDatabase.php

- Sends updated input from the user to the database.
- Displays the updated calendar.

--- Delete Event Pages ---

4.2.12 ChooseEventToDelete.php

- Displays a drop-down menu to select which event to delete.

4.2.13 DeleteEvent.php

- A confirm deletion page that lists all of the information on the selected event as well as the options to ‘Delete Event’ or ‘Cancel’. If the user selects to delete an event,

DeleteEventToDatabase.php is run, but if the user selects to cancel the deletion of an event, DisplayCalendar.php is run.

4.2.14 DeleteEventToDatabase.php

- Deletes information from user input to the database.
- Displays the updated calendar.

---Miscellaneous---

4.2.15 DisplayCalendar.php

- The landing page for the website. Displays initial blank calendar information, or your previously entered events.

4.2.16 buttons.php

- Stores information for redirect depending on what buttons are pushed on what pages.

4.2.17 calendar.php

- Contains basic calendar structure used on calendar display pages.

4.2.18 getinfo.php

- Gets information via POST for AddEventToDatabase.php and EditEventToDatabase.php.
- The POST information is received from getaddinfo.php and geteditinfo.php.

4.3 File Relations

- Each calendar display page contains buttons that redirect the user to the other file that matches that files' title. For example, the DisplayEvents.php file has buttons for "Add Event", "Edit Event", "Delete Event", "Next Month", and "Previous Month" that all redirect the user to the corresponding file.
- Each xTOx.php file merely helps direct the webpage back into itself. For example, Adding an event on the AddEvent.php webpage redirects to the AddEventToDatabase.php file, which reflects the altered DisplayEvents page to include the newly added event.
- All of the lowercase .php files hold initialization info for each page: certain information that allowed for redirects, temporary data holding, calendar structure, etc.

5 User Documentation

5.1 Installation

1. In order to use this software, you must have access to an ix.cs.uoregon.edu account.
2. Open up the command terminal and use the command "ssh <insert your username here>@ix.cs.uoregon.edu" and enter your password.
3. Use the command "cd public_html/"
4. Enter "git clone https://github.com/logangator2/CIS422_G5Calendar.git"

5. Go to ix.cs.uoregon.edu/~<insert your username here>/CIS422_G5Calendar/final/DisplayCalendar.php to begin viewing your calendar.
6. Additionally, without an ix.cs.uoregon.edu account, using the following URL will work with the calendar we have created: http://ix.cs.uoregon.edu/~hhardin/CIS422_G5Calendar/final/DisplayCalendar.php

5.2 User Guide

5.2.1 Basic Display

- After installation and setup, in the basic display you will see a calendar, displaying the current month, and visually showing all of the days in that month.
- At the bottom of the display, below the calendar, you will see options for adding an event.

5.2.2 Viewing the Next or Previous Month

- On any of the calendar display screens, you can press the button to view the next month by pressing "Next Month" or to view the previous month by pressing "Previous Month".

5.2.3 Adding an Event

- In the options for adding an event, you will see:
    - A text field to enter an event name and another for an event description.
    - Areas to input the starting date, starting time, ending date, and ending time.
    - Drop-down menus for selecting the priority and category of an event
    - A 'Submit' button, which when clicked, will add your event to the calendar, if you have entered any data.

5.2.4 Editing an Event

- To edit an event, press "Edit event" at the end of any calendar display page.
- This will bring up a drop-down menu where you can edit any existing event.
- Pressing submit will bring you to a page largely similar to "Adding an Event" above, although it will instead display the current information of the event that you wish to change.

5.2.5 Deleting an Event

- To delete an event, press "Delete event" at the end of any calendar display page.
- This will bring up a drop-down menu where you can delete any existing event.
- After pressing "Submit", the event's information will be shown and you must select "Submit" once again to delete the event for good.
- WARNING: Once you delete an event, there is no way to get that event back unless you manually re-add it.

Works Cited

1. Writer, Staff. “Top 10 Software Development Risks.” *ITProPortal*, 14 June 2010, https://www.itproportal.com/2010/06/14/top-ten-software-development-risks/
2. Rao, Srinivas. “Why Calendars are More Effective Than To Do Lists.” *The Mission*, 23 December 2017, https://medium.com/the-mission/why-calendars-are-more-effective-than-to-do-lists-9bc6ce3bee50
3. “HTML Tables.” *Browser Statistics*, www.w3schools.com/html/html_tables.asp.
4. “So You Want A Calendar, Huh?” *Htmlgoodies.com*, www.htmlgoodies.com/tutorials/tables/article.php/3479801/So-You-Want-A-Calendar-Huh.htm.