Calendar+

Authors: Chad Kinnard, Bo Yang Li, Tyler Marrapese, Warren Overstreet

Revised Requirements

1. GUI

1.1. Implemented using HTML, CSS & JavaScript

1.2. Login Page

1.2.1. Field for username / email [REQUIRED FIELD]

1.2.2. Field for password [REQUIRED FIELD]

1.2.3. Create account button

1.2.3.1. When pressed, redirects to the account creation page

1.2.4. Login button

1.2.4.1. When pressed, logs the user in and redirects to the main webpage.

1.2.4.2. If either username/email or password are missing, a browser alert popup indicating required fields needing to be filled appears and the user will be able to try again.

1.2.4.3. If login credentials are incorrect, a browser alert popup indicating incorrect credentials appears and the user will be able to try again.

1.3. Account Creation Page

1.3.1. Field for username [REQUIRED FIELD]

1.3.2. Field for email [REQUIRED FIELD]

1.3.3. Field for password [REQUIRED FIELD]

1.3.3.1. 8 character minimum

1.3.4. Register button

1.3.4.1. When pressed, it creates an account for the user in the database, automatically logs them in, and redirects to the main webpage.

1.3.4.2. If any required field is missing, a browser alert popup indicating missing fields appears and the user will be able to try again.

1.3.4.3. If the password is less than 8 characters long, a browser alert popup indicating insufficient password length appears and the user will be able to try again.

1.4. Main Component

1.4.1. Monthly Calendar

1.4.1.1. 7x6 grid which will be populated with dates

1.4.1.2. Clicking a date will bring up the event creation side panel, with the date field already filled out

1.4.1.3. Events will show up within the box of the date they are assigned to

1.4.1.3.1. Clicking an event will bring up the side panel to edit/delete the event

1.4.1.3.2. Events will be colored if a color is chosen for them

1.4.1.3.3. If there are too many events for a grid signifying a day, the user will be able to scroll down within the grid to display the other events.

1.4.2. Weekly Calendar

1.4.2.1. 7x1 horizontal grid populated with dates

1.4.2.1.1. Holidays will be placed at the top

1.4.2.1.2. Each grid has a timeline for the day, displaying all events for the dates.

1.4.2.1.2.1. Only event title and times are displayed

1.4.2.1.2.2. Clicking on an event block will display full information about the event

1.4.2.1.2.3. Clicking an empty place on the timeline will create an event at the timeline position clicked

1.4.2.1.2.3.1. As with the monthly calendar, the side panel will be brought up to edit/delete the event

1.4.2.1.2.4. If applicable, a red line signifying the current time will be placed under one of the dates on the appropriate location on the timeline.

1.4.3. Daily Calendar

1.4.3.1. Weekday and date displayed at the top

1.4.3.2. The date display is followed by a timeline for that date

1.4.3.2.1. The day timeline works similarly to the weekly timeline

1.4.3.2.2. If applicable, a red line signifying the current time will be placed under the timeline (Only if date displayed is current date)

1.4.4. Yearly Calendar

1.4.4.1. 4x3 grid of months

1.4.4.1.1. Each month contains a 7x6 grid which will be populated with dates

1.4.4.1.1.1. Clicking a date will bring up a popup that says if there are any events on that date

1.4.5. Schedule View

1.4.5.1. Vertical list of all events contained within a given range (day, week, month)

1.4.5.1.1. Date, day, time period (from, to), title of event

1.5. Collapsible Side Panel

1.5.1. Button to create event

1.5.1.1. Will switch the side panel to one that prompts for details of event

1.5.1.1.1. Fields for title, description, date, time (from, to).

1.5.1.1.1.1. Title defaults to “No Title”

1.5.1.1.1.2. Date defaults to current date

1.5.1.1.1.2.1. When the date field is clicked, a miniature monthly calendar will pop up, allowing the user to select a date on the miniature calendar. Therefore, it is not possible to select any invalid date such as Feb. 30.

1.5.1.1.1.3. Time duration defaults to an hour

1.5.1.1.1.3.1. Changing the beginning time to beyond the ending time will cause the ending time to be shifted to one hour after new beginning time

1.5.1.1.1.4. Description optional, text only

1.5.1.1.1.4.1. No character limit

1.5.1.1.2. Save button to confirm the event

1.5.1.1.3. Delete button to delete the event

1.5.2. Simplified monthly-view calendar (won't have events displayed, just a blank calendar)

1.5.2.1. Clicking on a date will bring the focus of the main calendar component to that specific date, depending on which view you're in.

1.5.3. [Optional] Create/Toggle visibility of multiple calendars

1.5.3.1. Button to create additional calendar

1.5.3.1.1. Field for label

1.5.3.2. Button to toggle visibility

1.6. Top Nav Bar

1.6.1. Button to hide and unhide the side panel

1.6.2. Depending on the view, two buttons to cycle forward and backward in time

1.6.2.1. Forward and backward 1 day in daily view, 1 week in weekly view, etc.

1.6.3. Buttons to switch between daily view, weekly view, etc

1.6.4. Button to view schedule (will display a list of all events, date and time of event, name of event, etc.)

1.6.5. [OPTIONAL] Button to view settings

1.6.5.1. E.g. Multiple time zones, default event duration, calendar import, etc.

1.6.6. Profile icon

1.6.6.1. Opens a small window under it

1.6.6.1.1. Sign out button

1.6.6.1.1.1. When pressed, logs the user out and redirects to the login screen

2. Account Login & Creation

2.1. When logging in, the user's data will be checked against the database of accounts as described in **3.2.**

2.1.1. Users may create an account if they do not have one

2.1.1.1. Username must be unique and 8 characters long

2.1.1.2. Password must be at least 8 characters long

2.2. When logged in, a user may view their personal events

2.2.1. Events will appear on the calendar as described in **1.3.1.3.**

3. Data Management
    3.1. Event Data
        3.1.1. All Event data will be stored within the database with the given attributes
            3.1.1.1. Date, Time(start, end), ReminderDate(optional), ReminderTime(required only if ReminderDate has a value), Title, Username, Location(optional), Description(optional), EventColor, event_id(unique integer assigned to each event)
            3.1.1.2. Users will default to a basic calendar with a default set of U.S. Federal holidays
                3.1.1.2.1. A template will be created and copied to each user's list of events upon account creation
        3.1.2. Upon Deletion request, Event Data will be removed from the database
            3.1.2.1. Deletion will occur at user request or once the event falls out of the calendar's scope of time
    3.2. Account Data
        3.2.1. User Login Data will be stored within the database with the given attributes
            3.2.1.1. Username (unique varchar chosen on account creation), Password, Email
            3.2.1.2. A user's Events will be tied to their account through their UserID
    3.3. Server Information
        3.3.1. Hosted via Linode
            3.3.1.1. $5 monthly plan
                3.3.1.1.1. Nanode 1GB
            3.3.1.2. Ubuntu Linux Distro
            3.3.1.3. Hosted from Atlanta, GA

4. Reminder System
    4.1. If an entry is found on an event's ReminderDate field an alert will occur at the chosen time found in ReminderTime
        4.1.1. Alerts consist of the following
            4.1.1.1. a small notification may appear with the title of the event in question visible on the notification
                4.1.1.1.1. User may simply click the button on the notification to cause it to close
        4.1.2. [Optional] Chrome alerts
            4.1.2.1. Adds events to Windows notification tray

Design Description

The Block Diagram Perspective

The block diagram (*Figure 1*) of the Calendar+ web application illustrates a three layer system composed of the user computer/browser, the hosting server, and the database. The user computer is the presentation layer of the Calendar+ web app; it represents the visible parts of the web application to the user and is responsible for network communications with the backend as well as providing system notifications to the user so that reminders can be seen by the user. If the user enacts an action that does not require any response from the backend, such as switching to the yearly calendar view, the browser layer will complete the action itself without having to communicate with any other layer.

Whenever the user desires a task involving the backend such as account login to be completed, the user computer sends a request to the server hosting the Calendar+ application. The server is responsible for validating the user-defined task. In the event that the user request contains some sort of error, the server will send the request back to the user computer which is then responsible for showing the error on the user's screen. For example, if the user tries to login with incorrect credentials, the server will deny the login attempt and communicate to the user computer to display an error to the user. In any case, the server is ultimately responsible for enforcing the restrictions of the Calendar+ web application. It serves as an intermediary between the user computer/browser layer and the database, reading data from the database and transferring it to the browser layer and writing data to the database as requested by the front-end as necessary.

The database is by far the simplest part of the system. It is responsible for holding all application data across all users. As all verification and validation is completed by the server layer which is the only layer the database communicates with, the database itself does not need to worry about potential errors in its data.

Overall, the benefits of this closed three-layer system are clear. It offers simplicity, clear separation of concerns, and a degree of modularity that makes it easy to change out individual layers or to add additional layers if extra functionality is required. However, the drawbacks are also clear and enormous. Because this system is a single unit, making a change requires redeploying the entire system. Because data access requires a chain of communication that goes from the browser layer all the way down to the database and then all the way back up to the browser layer, this layered architecture has no scalability. Its modularity is limited to the individual layers. And lastly, if one layer fails, the entire system also fails. There is no fault tolerance.

The User Interface Perspective

There are a total of seven pages and layouts, including: Login, Registration, Month view, Week view, Day view, Year view, and Schedule view. Almost all elements and pages use the CSS Flex property, to maintain some level of responsiveness regardless of what resolution the website is being viewed at. Whenever someone visits the site and is not logged in, they will be greeted by the Login page. The contents of the login page (*Figure 2.1*) are centered both vertically and horizontally. From top to bottom, the page consists of: the Calendar+ logo, an input field for a username / email, an input field for a password, and side-by-side, a register button and a sign in button. In the event that a user’s login information is incorrect, clicking the “sign in” button will notify them that some of the information was incorrect. If the information is all correct, they will be signed into their account and redirected to the Month view page. If the user does not have an account, they can click the “register” button, which will redirect them to the Registration page.

The Registration page (*Figure 2.2*), like the Login page, is centered both vertically and horizontally. From top to bottom, the page consists of: the Calendar+ logo, an input field for a username, an input field for an email, an input field for a password, an input field to confirm the password, and a register button. In the event that any of the fields do not contain “legal” information, or the passwords do not match, clicking the “register” button causes the user to be notified of the issue and an account is not created. If the information is all allowed and correct, clicking the “register” button creates an account and redirects the user to the login page, where they can plug in the information they just provided to successfully log in to their new account.

The actual calendar pages all consist of three main elements: the top bar, the sidebar – which has multiple possible layouts and functions – and the main “calendar” component.

The top bar, as the name suggests, is at the top and spans the entire width of the screen. From left to right, the top bar consists of: the Calendar+ logo, a left / previous button, the currently viewed date(s), a right / next button, a dropdown to select the view, and a sign out button. The “currently viewed date” changes depending on the current view; for instance, it can read: “February 2022”, “February 19, 2022”, “2022”, etc. The dropdown will have five options, one for each of the different views. Clicking any of these options changes the main “calendar” component to the corresponding layout. Upon clicking the left / previous button, the currently displayed month, week, etc, is changed to the previous month, week, etc. Upon clicking the right / next button, the currently displayed month, week, etc, is changed to the next month, week, etc. Clicking the “sign out” button signs the user out of their account and redirects them to the Login page.

The sidebar is locked to the left side of the screen, and has three different views / layouts. The main view (*Figure 2.3*) consists of a Create Event button and a miniaturized version of the month calendar. Clicking the “Create Event” button changes the layout of the sidebar to the “event creation” layout (*Figure 2.4*). This layout has an input field for the title of the event, fields for time and date selection, an “All Day” checkbox, an optional input field for the event location, an optional input field for the event description, optional fields for reminder time and date, a back button, and a save button. Clicking the “All Day” checkbox disables the input fields for time selection and automatically sets the start time to 12:00 AM and the end time to 11:59 PM. Clicking the “Back” button will cause the state of the sidebar to be reverted to the main sidebar layout. In the event that some of the required fields do not contain any information, clicking the “Save” button will notify the user of the missing fields and will not create the event. The “event information” sidebar (*Figure 2.5*) is only accessible by clicking on an event tile. It consists of five non-editable fields which are automatically filled out with information from the database: Event Title, Date, Time, Location (if available), and Description (if available); as well as three buttons: back, edit, and delete. Clicking the “Back” button will cause the state of the sidebar to be reverted to the main sidebar layout. Clicking the “Edit” button will open the “event creation” sidebar layout with pertinent information already filled out. If the “Delete” button is clicked, the user will be prompted to confirm the decision, as the event will be permanently deleted from the database and will not be recoverable.

The Month view of the main “calendar” component (*Figure 2.6*) contains a 7x5 or a 7x6 grid of cells (depending on the month). Each column of the grid is topped by the respective day of the week (Sun - Sat). Each cell in the grid contains a date (the numbered day of the month), and if an event is in the database for that specific date, an event tile containing the title of that event. If there are multiple events on the same day, they will be listed one on top of the other inside of that day’s cell (*Figure 2.7*). Clicking on an event tile will open the “event information” sidebar.

The Week view (*Figure 2.8*) contains a 7x24 grid of cells. Each column of the grid is topped by the respective day of the week (Sun - Sat), as well as the numbered day of the month. If there is an event that would fall on a day on the calendar, there is an event tile that spans from the cell containing the starting time, to the cell containing the ending time. The event tile contains the title of the event & the starting and ending times of the event. If there are events that would overlap times on the same day, they are narrowed down to each occupy (100% * n) of the width of that column, where n is equal to the number of tiles that would overlap.

The Day view (*Figure 2.9*) contains a 1x24 grid of cells. Above the column is the weekday and day of the month which is currently being displayed. If there is an event that would fall somewhere on the viewed date, there is an event tile that spans from the cell containing the starting time, to the cell containing the ending time. The event tile contains the title of the event & the starting & ending times of the event. If there are events that would overlap times, they are narrowed down to each occupy (100% * n) of the width of the container, where n is equal to the number of tiles that would overlap.

The Year view (*Figure 2.10*) contains a 4x3 grid of cells. Each cell contains a miniaturized version of the month calendar (one cell for every month of the year).

The Schedule view (*Figure 2.11*) contains a vertical list containing only dates which have at least one event on them. Each panel in the list contains a date, time, and event title.

The Database Perspective

For the database portion of Calendar+ we have used MySQL. If you refer to figure 3.1 in the appendix you will see that our database will, at current, feature two tables: Events, and Users. The Users table will feature three attributes: user_name, Password, and Email. The user_name attribute will be the primary key meaning that it will be a unique value and shall be used to access data from the Events table. The user_name attribute will be of the varchar data type and will at maximum be 16 characters long and must have an entry within the table. The Password attribute will be of the varchar data type and will at maximum be 20 characters long and also must have an entry within the table. The Email attribute will also be of the varchar data type and will at maximum be 40 characters long and must have an entry within the table. All of the entries of the Users table will be input during account creation.

The second table named Events will feature nine attributes: event_id, user_name, event_title, eventDate, startTime, endTime, eventLocation, eventDescription, reminderDate, eventColor. The event_id attribute will be of the integer data type and shall auto increment, event_id will be the primary key of the Events table. The user_name attribute will be a foreign key relating the Events table to the Users table and must have an entry within both tables. The user_name attribute will relate to all events created by that user_name, thus allowing us to query the database on events created solely by that user’s account to populate the calendar with their events. The eventDate attribute will be of the Date datatype and will store the date of each event in the YYYY-MM-DD format, eventDate is required for all entries of the Event table. The startTime attribute will be of the Time datatype and will store the beginning time of each event in the hh:mm:ss format, this data type is required. The endTime attribute will be of the Time datatype and will store the end time of each event in the hh:mm:ss format and must be greater than the startTime of this same event. The eventLocation attribute will be of the varchar data type and will at maximum be 60 characters, this attribute can be null. The reminderDate attribute will be of the date datatype and is optional and will store the reminder date of each event in the YYYY-MM-DD format if the user wants a reminder. The reminderTime attribute will be of the time datatype in the YYYY-MM-DD format and will be required if the user enters a reminderDate but will not be required if the user does not enter a reminderDate.The reminderDate attribute will be checked and if this attribute is not null then the reminder system will alert the user at the chosen time and date. The eventColor attribute will be of the varchar attribute and will be 6 characters long. The eventColor attribute will be stored with hexadecimal data for the color the user wishes the event to appear on the calendar. All of the entries of the Events table will be input during event creation.

The Functional Perspective

We decided to break down the JavaScript of Calendar+ into multiple .js files. There is a file for the login, day view, week view, month view, year view, schedule view, common functions, sidebar, and events. The login file is used for the original login screen. It contains functions such as getUsername(), getPassword(), registerUser(), acceptUser(), receiveLoginError(). The functions getUsername() and getPassword() receive an existing user's credentials. The function acceptUser() allows the user access to Calendar+ if it matches an existing user in the database. The function receiveLoginError() denies the user access if the credentials do not match an existing user in the database. The registerUser() function redirects the user to the account registration page.

The register file is used for the account registration page. It contains the functions createUsername(), createPassword(), getEmail(), acceptCreation, and receiveCreationError(). This is functionally the same as the login page. The main difference is the restrictions in making the account. For example, if a user enters a password that is not at least eight characters long, then the receiveCreationError() function is called.

The different schedule views are very similar in structure and semantics. Due to this and the top bar remaining basically the same throughout, we decided to make a common file that hosts many of these functions. For example, this includes the function changeView(view) which allows switching between the different calendar views via a drop down menu by taking whichever view as an argument. The other functions in this file are logOut() which directs the user back to the login screen, nextView(view) which allows users to quickly switch between days, weeks, months, etc., and displayEvent(view) which will make registered events appear on each calendar view.

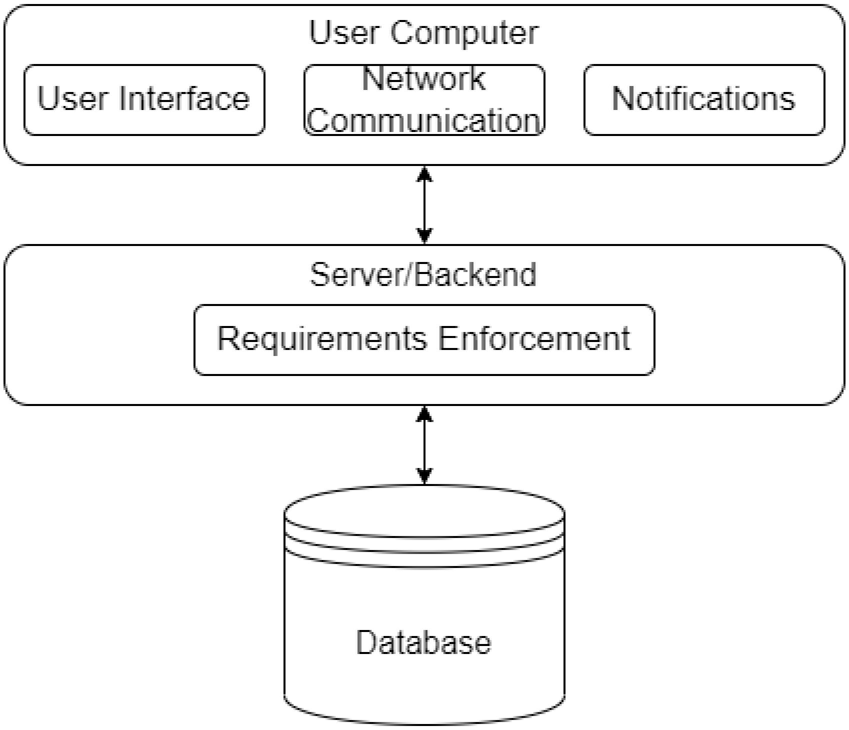
The day, week, month, year, and schedule files include the functions showDay(), showWeek(), showMonth(), and showYear(), and showSchedule() respectively. This is where the structure of each page will be handled.

The sidebar file focuses on the side bar which remains the same no matter which view is currently displayed. It contains the miniCalendar() function which automatically shows the month, no matter which view is displayed on the main page, that the user is on. It also contains the createEvent() function which redirects to the event creation sidebar.

The events file is where the user creates their personal events. This includes the functions saveEvent(), deleteEvent(), reqError(), and acceptEvent(). The function saveEvent() saves the information that the user entered such as the title, time, date, and description. This is only accepted by acceptEvent() if the required fields (title and time/date) are filled out. Otherwise, the reqError() function will set off a pop-up alert informing the user to fill out the required fields. After the event is saved, the deleteEvent() function will allow the user to delete the saved event.

Appendix

Figure 1 - Block Diagram of the Calendar+ System



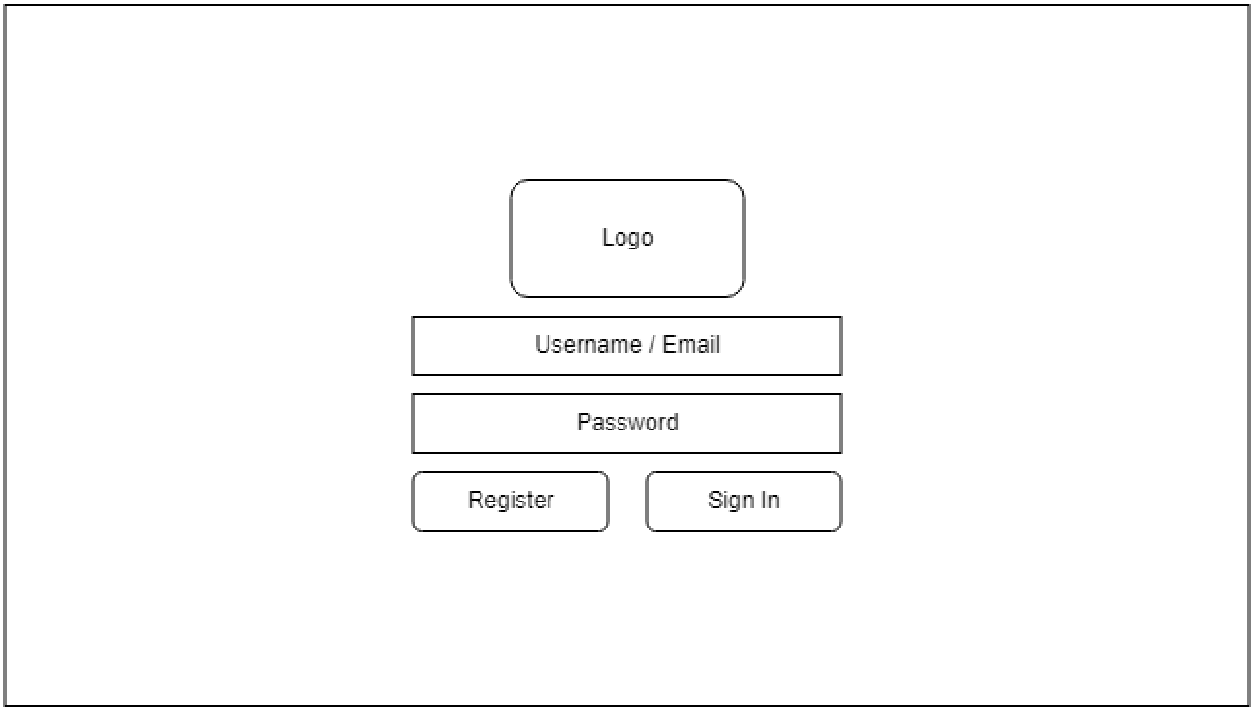


Figure 2.1 – Login Page

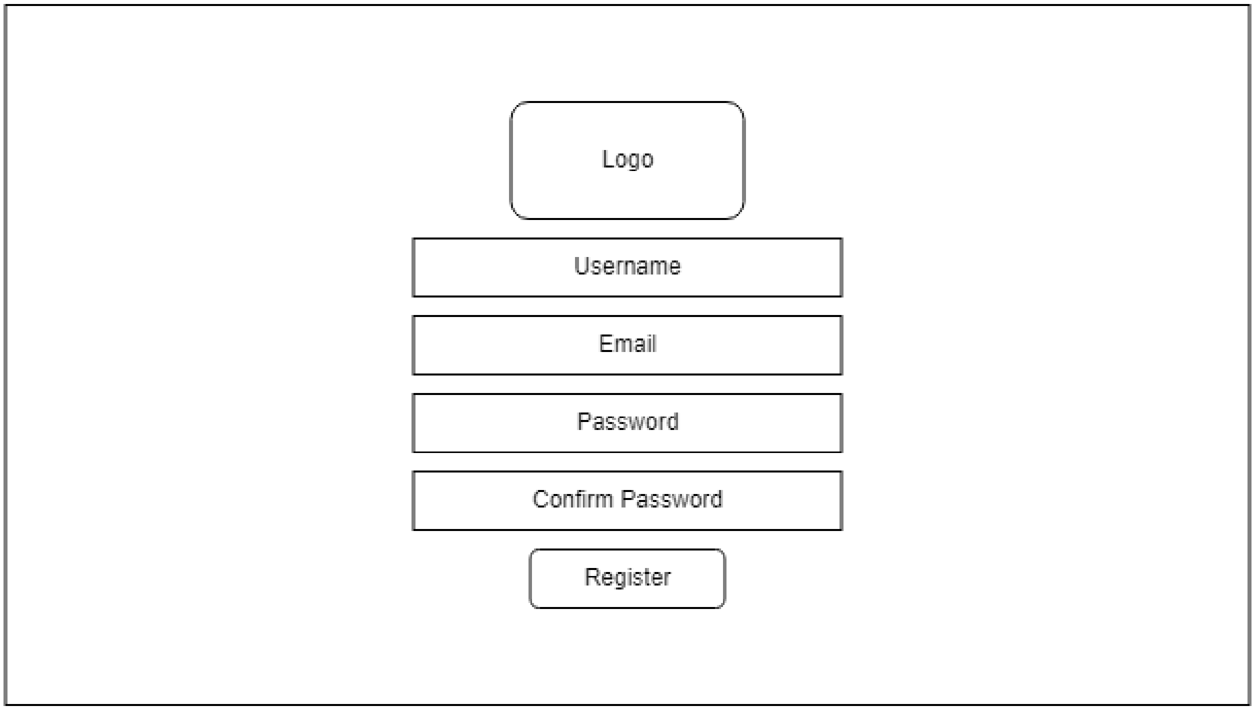
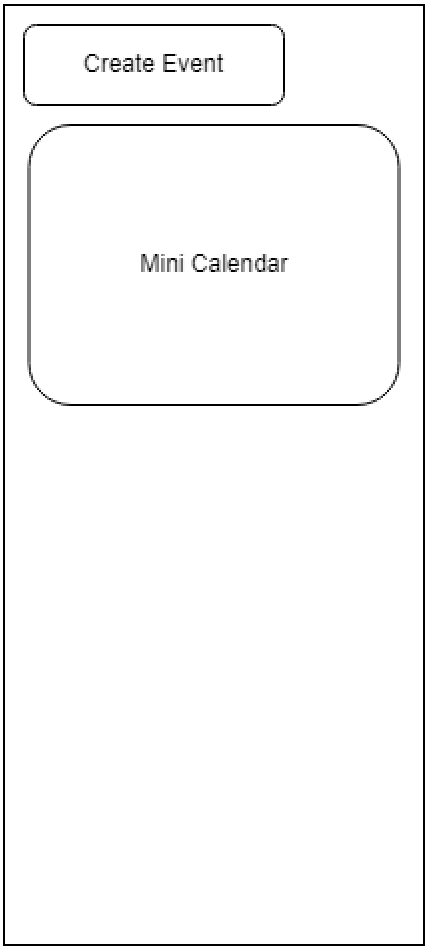


Figure 2.2 – Registration Page



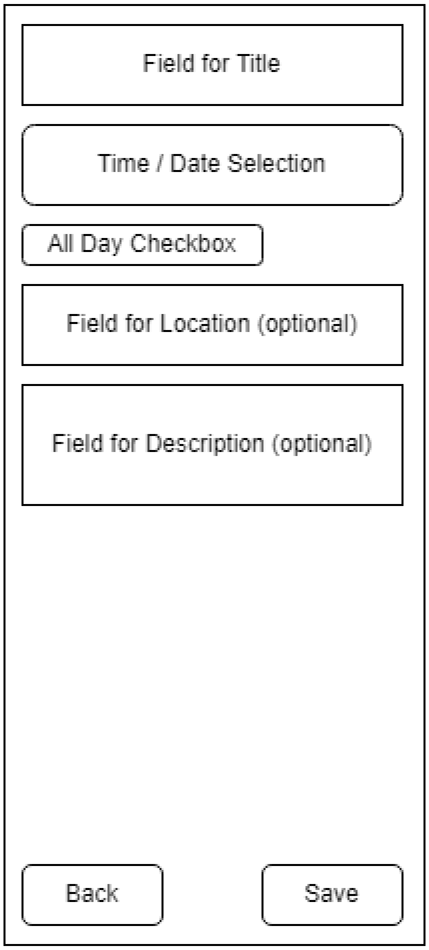

Field for Title
Time / Date Selection
All Day Checkbox
Field for Location (optional)
Field for Description (optional)
Back
Save

Figure 2.3 – Main Sidebar | Figure 2.4 – Event Creation Sidebar

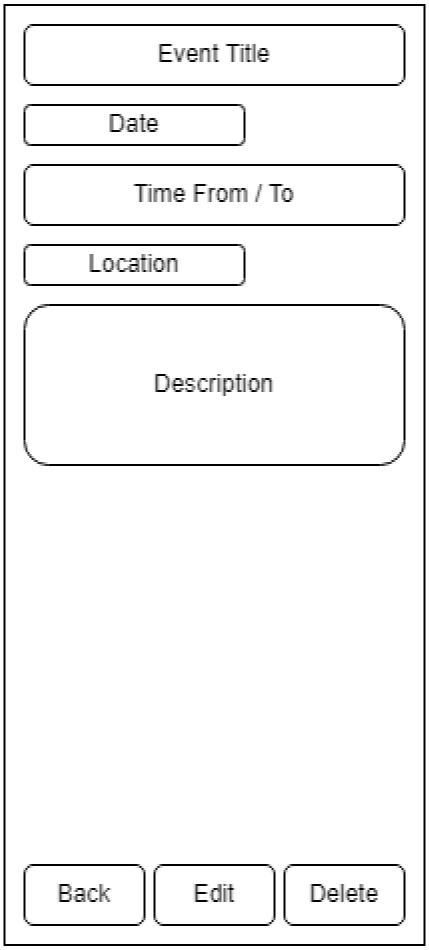


Figure 2.5 – Event Information Sidebar

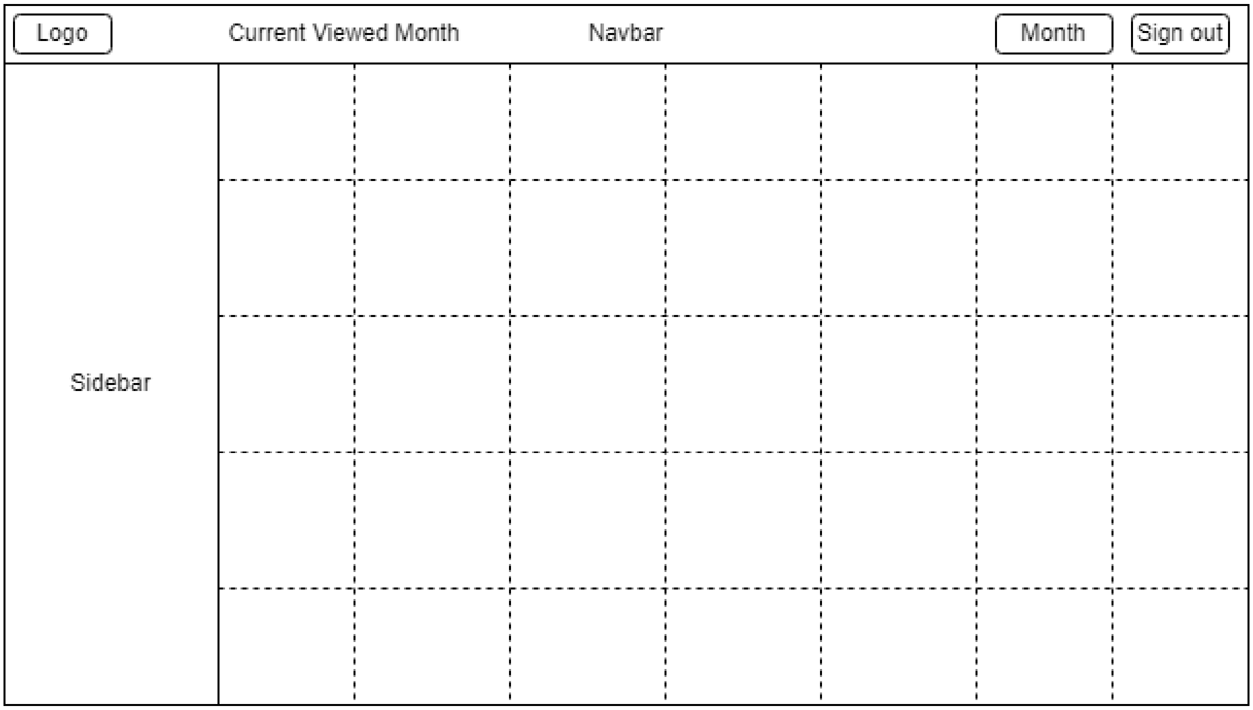


Figure 2.6 – Month View

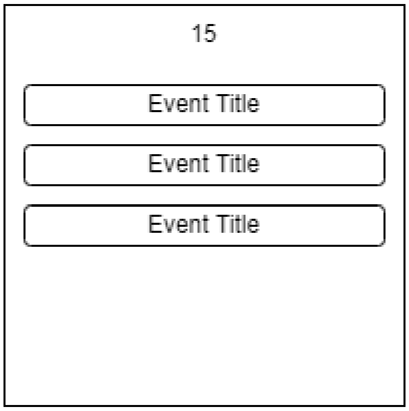


Figure 2.7 – Zoomed In Month View Cell

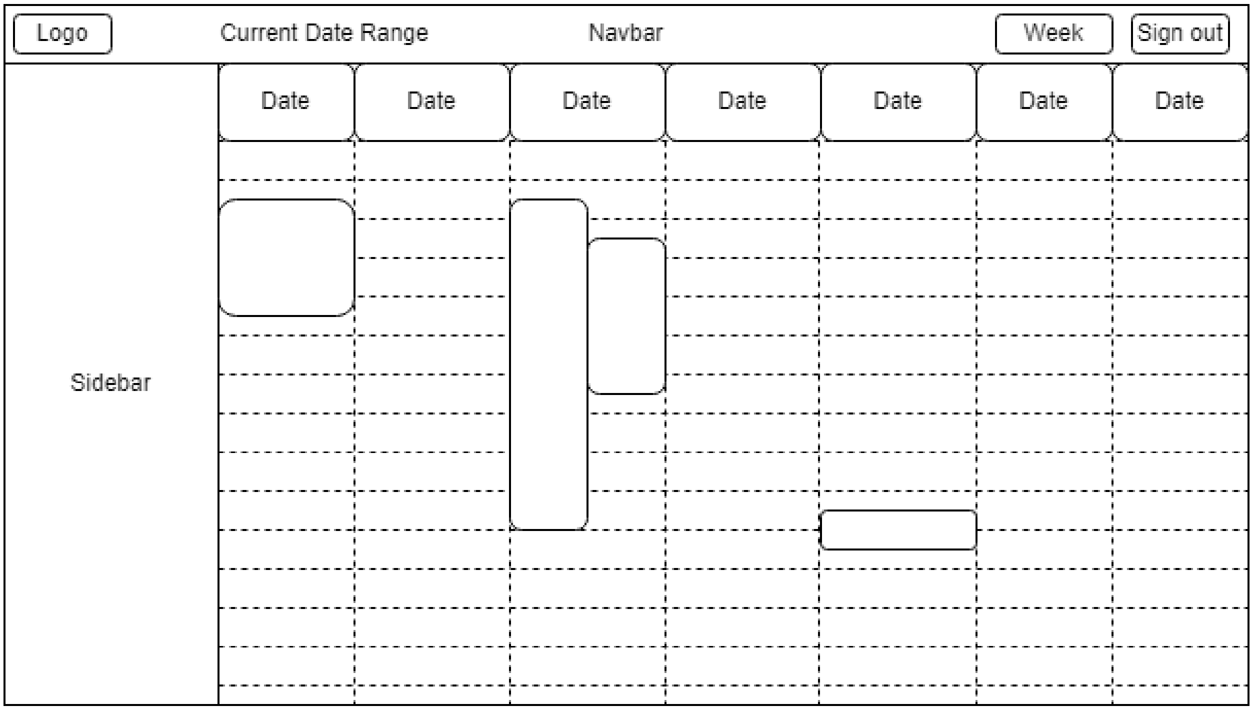


Figure 2.8 – Week View

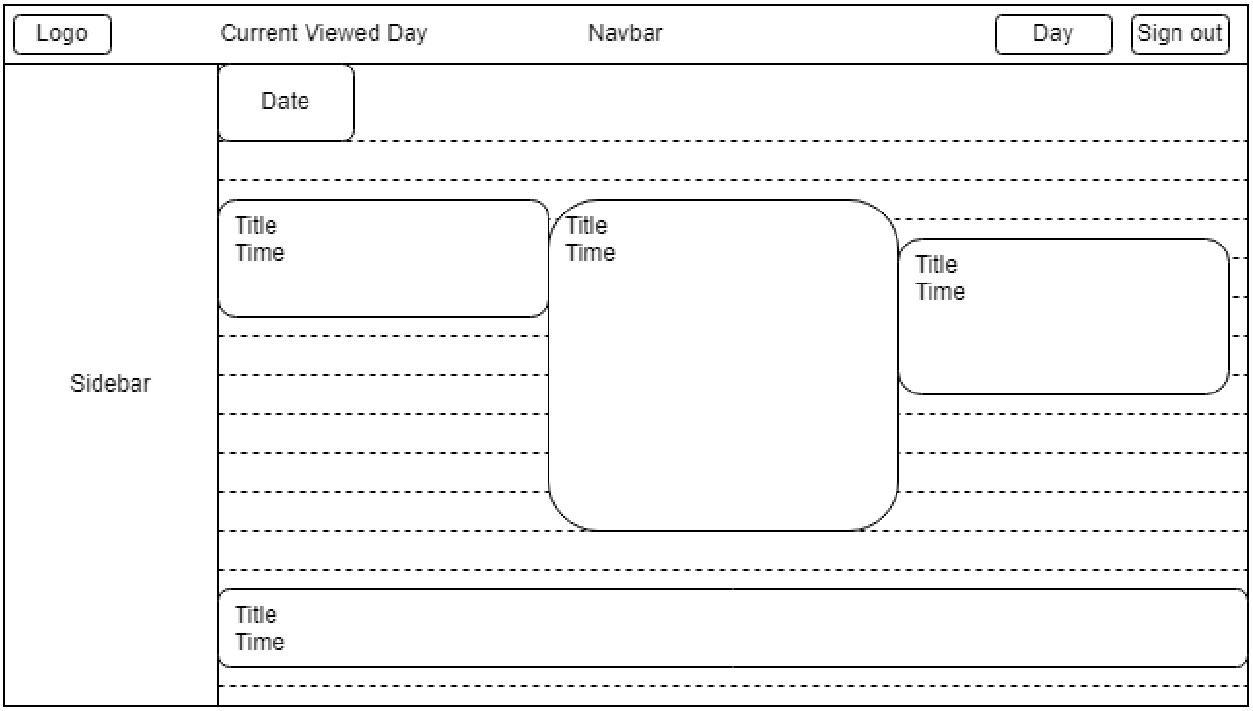


Figure 2.9 – Day View

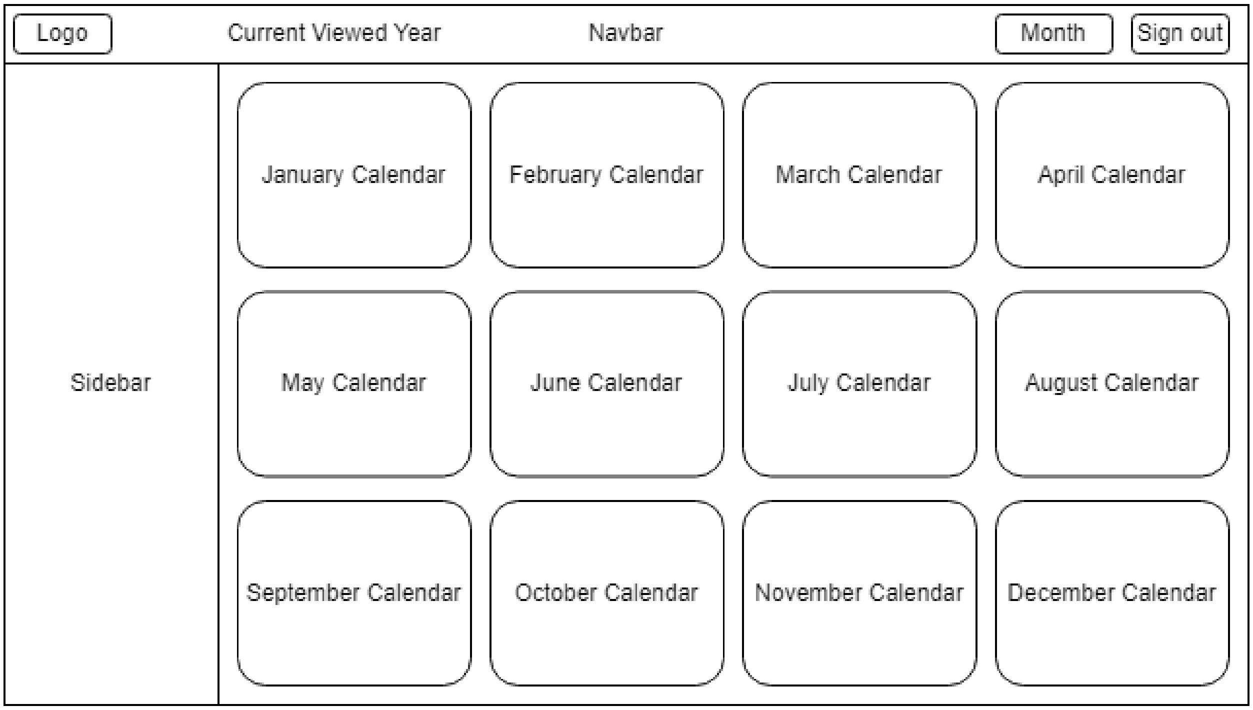


Figure 2.10 – Year View

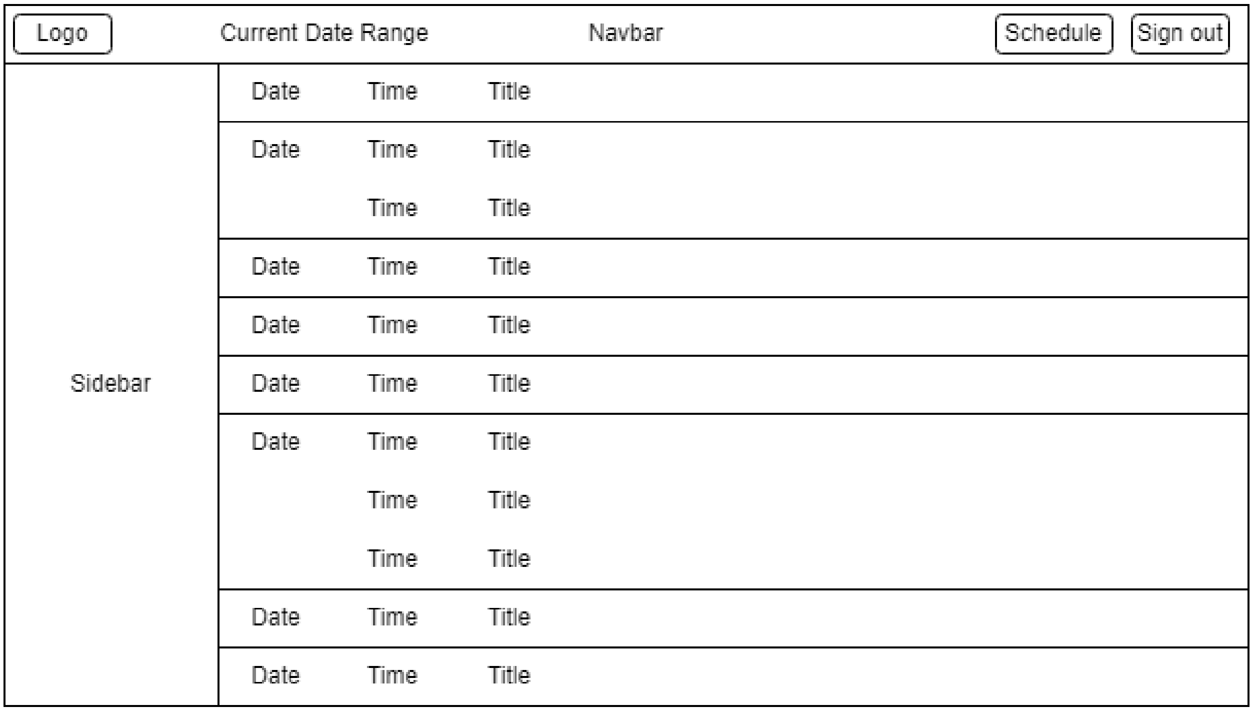
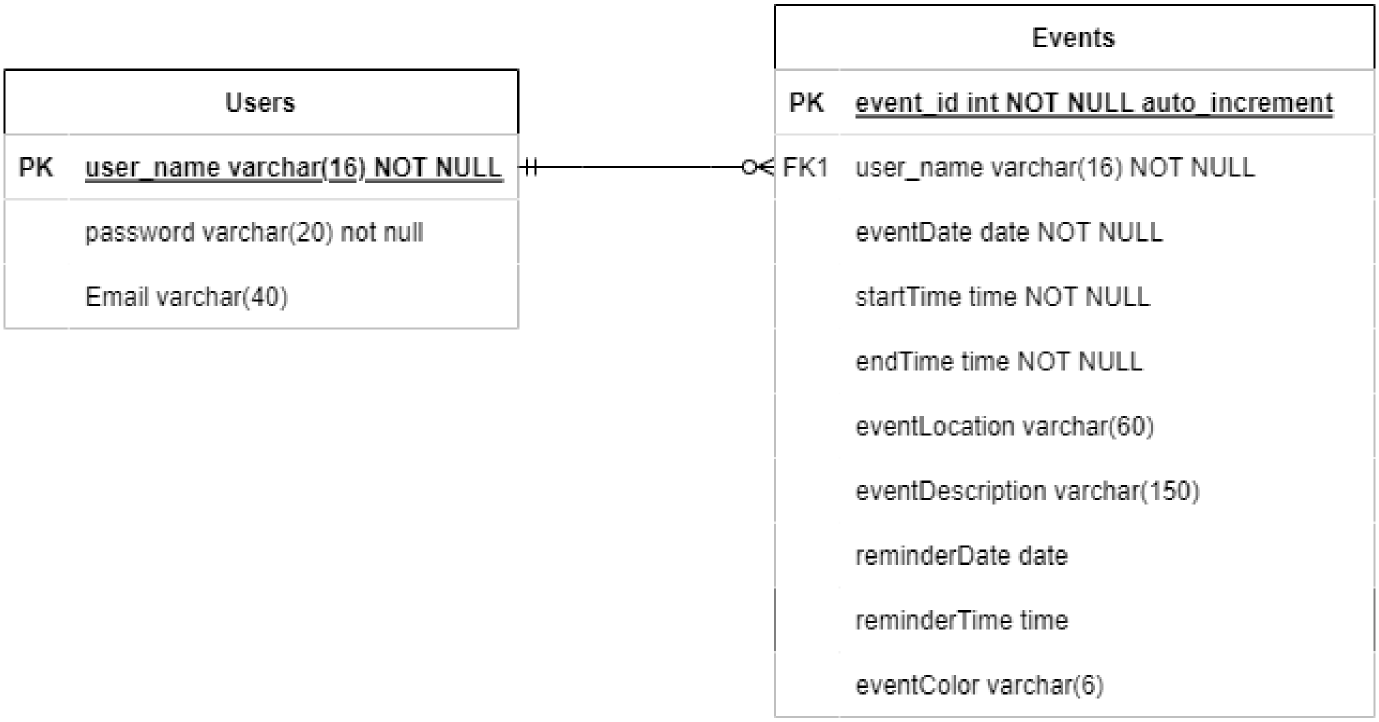


Figure 2.11 – Schedule View



Users	
PK	user_name varchar(16) NOT NULL
	password varchar(20) not null
	Email varchar(40)

Events	
PK	event_id int NOT NULL auto_increment
FK1	user_name varchar(16) NOT NULL
	eventDate date NOT NULL
	startTime time NOT NULL
	endTime time NOT NULL
	eventLocation varchar(60)
	eventDescription varchar(150)
	reminderDate date
	reminderTime time
	eventColor varchar(6)

Figure 3.1 – ERD Diagram

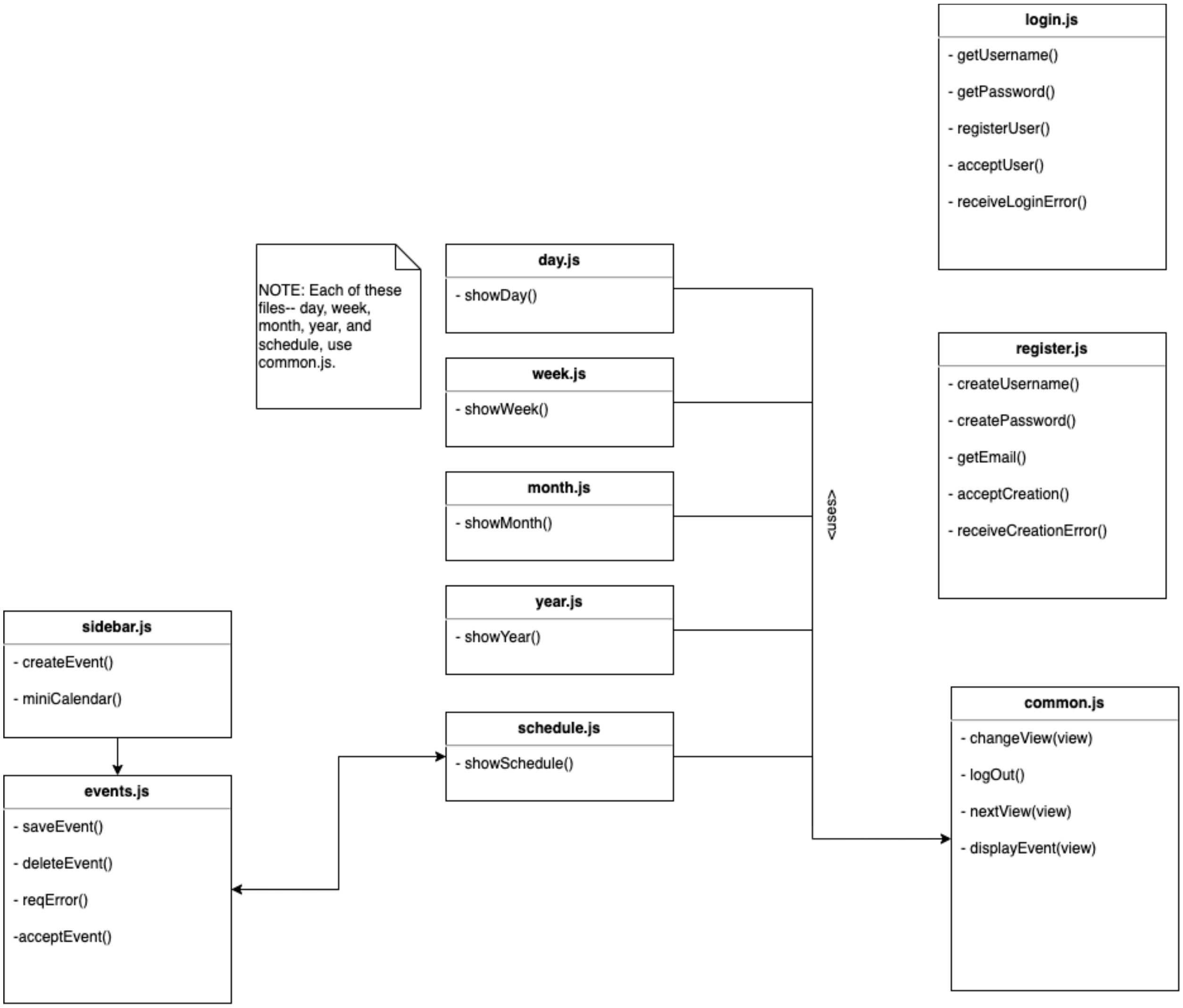


Figure 4.1 - Component Diagram