Calendar Event and Task List Manager

Prepared by Team Thundercats

David Smith
Shubhangi Rakhonde
David Schechter
Zayd Hammoudeh

Table of Contents

1. Application Name

Event and Task List Manager

2. Application Description

This section provides an overview of the application, including its intended purpose.

2.1. Application Purpose

This application simplifies the management (e.g. viewing, modifying, prioritizing, etc.) of a user's daily calendar as well as the user's to-do/task list by integrating these items into a single, cohesive interface.

2.2. Application Overview

It is common for a person to have multiple, distinct, daily calendars stored on different, disconnected platforms. For example, a business professional may have a work calendar on his/her company's corporate network, a personal calendar as part of his/her Google account, and a social calendar on Facebook. Managing and visualizing these disparate calendars can be cumbersome and difficult; this application simplifies this otherwise burdensome task by integrating all of a user's different calendars into a unified calendar where the user can visualize and modify all of his/her calendars using a single, cohesive interface.

In addition to scheduled events, meetings, and appointments, an individual usually must also complete a set of tasks, chores, errands, etc. These tasks may be professional, personal, social, etc. This application also integrates the ability to create and manage the user's tasks in the form of an advanced "to-do list".

By juxtaposing in a single interface an individual's calendar with the tasks s/he must perform, a user is able to easily visualize and prioritize all of his/her daily activities. Moreover, the application will support the ability to provide user alerts (e.g. SMS text message or email) to serve as event reminders. Therefore, this application's integrated approach helps prevent the inefficiencies and issues (e.g. belated completion of tasks) associated with what today is an unstructured system.

3. Functional Requirements

This section reviews the event and task manager's functional features. The requirements have been organized into different categories based on their logical role in the application.

3.1. General Functional Requirements

A. Before accessing this tool's features, the user must enter his/her username and password correctly into the application.

B. If a user fails to enter valid login credentials, the application shall display an error message instructing the user to enter valid credentials or to create a new account.

3.2. Functional Requirements Related to Events and Tasks

A. The application shall allow two types of user items; the application shall track these differing types separately since there is limited overlap in their nature.
   i. The first type is a user calendar event, which is "synchronous" in nature in that such an event is associated with a specific time and date on the calendar. Examples of synchronous calendar

events would be a user's: doctor's appointment, business meeting, and birthday party; note that all of these synchronous events occur at specific times.

ii. The second type is a user task, which is "asynchronous" in that the user has significant flexibility regarding the time the task can be performed/completed. For instance, a possible asynchronous task could be to cut the grass or to go shopping since the user can do them when they see fit or have time. In contrast, if the user only had access to a lawnmower from 2-3pm on Saturday, mowing the grass could turn from being asynchronous to synchronous.

B. The application shall allow the user to create notifications regarding certain events. These notifications will be sent by the application to the user at specified times.

3.3. Functional Requirements Related to Event and Task Priority

A. Users shall be able to specify a priority for each synchronous calendar event and asynchronous task.

B. The user shall be able to order the asynchronous tasks according to different criteria including but not necessarily limited to: task priority, task creation date, and alphabetically by task name.

3.4. Functional Requirements Related to Synchronous Calendar Events

A. The user shall be able to integrate and modify their synchronous calendar events from other applications/services.

B. The user shall be able to create and modify synchronous calendar events that were not imported from another third-party application.

C. The user shall be able to specify that a synchronous calendar event occurs at a specified regular interval (e.g. once a month, once a week, every Tuesday and Thursday, etc.), and the application must automatically replicate such a task, based off the user specified time recurrence profile.

D. The application must support the ability to create and appropriately display multiple synchronous calendar events that overlap in time (e.g. two separate meetings scheduled at exactly the same time).

3.5. Functional Requirements Related to Asynchronous Task Completion

A. The application must support the ability to categorize asynchronous tasks as either uncompleted or completed.

B. Once an asynchronous task has been completed, the user shall be able to mark it as "Completed" by clicking a check box next to the task. Upon an asynchronous task's completion, the application must automatically remove the task from the set of uncompleted tasks and include it in the set of completed tasks.

C. Once a task has been marked as completed, the application must allow the user to specify the task's completion time. This completion time can be either the current time, or the application shall allow the user to specify another time.

4. Nonfunctional Requirements

The following are a set of nonfunctional requirements for the event and task manager. The requirements have been categorized depending on each one's logical type.

4.1. General Nonfunctional Requirements

A. The application shall be accessible to the user through a web browser. At the minimum, the application needs to support Google Chrome since it is themost used browser with support for other browsers prioritized based on their user base size.

B. As described in the section entitled "Functional Requirements Related to Events and Tasks", the application can send the user notifications. Types of notifications that shall be supported include but are not necessarily limited to: SMS text messages, browser pop-ups, and emails.

4.2. Nonfunctional Requirements Related to Calendar Events

A. The user shall be able to import calendar eventss from at least the following set of third-party platforms: Google Calendar, Facebook, and Apple's calendar application.

B. When creating an event, the application must support the ability for the user to specify the following event information:
   i. Event name
   ii. Event time and date
   iii. Event description (if any)
   iv. Invitee List (if any)
   v. Event recurrence (e.g. once a week, every Tuesday/Thursday - if any)

4.3. Nonfunctional Interface Requirements

A. The application shall display synchronous events and asynchronous tasks in a side-by side two panel view as shown in figure 1.

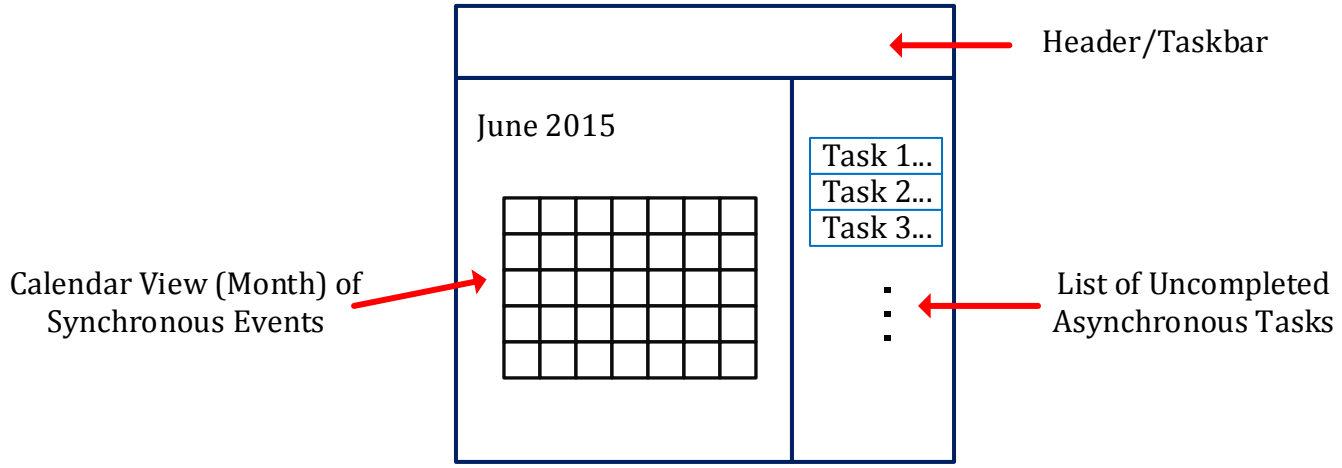

Figure 1 – Basic Structure of the User Interface

B. Tasks and events with different priorities shall be color coded in the user interface. The color scheme used must make higher priority events and tasks appear more prominently than their lower priority counterparts.

C. The rating system used to prioritize tasks shall be on a scale of zero to five stars, with zero stars being the lowest priority and five stars being the highest priority.

D. When creating a synchronous event, the application form shall resemble the structure shown in figure 2.

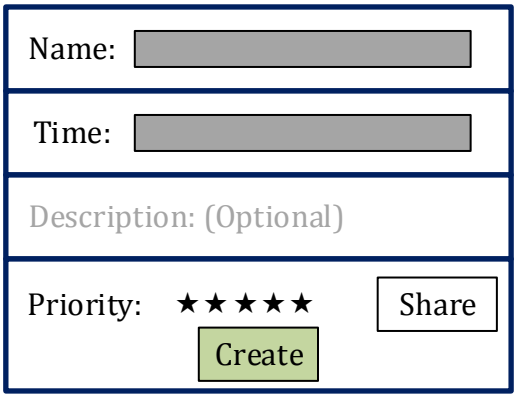

Figure 2 – Synchronous Event Creation Form Model

E. As described in the section entitled "Nonfunctional Requirements Related to Calendar Events", the event description field is optional. If the user clicks on the description field shown in figure 2, the form shall lengthen to allow the user to enter an event description as shown in figure 3.

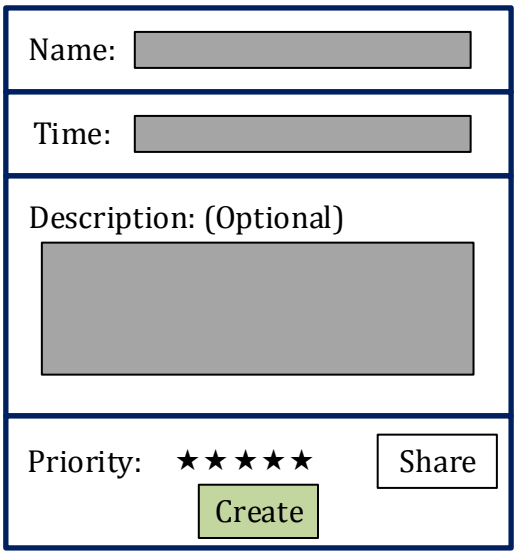

Figure 3 – Synchronous Event Creation Form Model with Expanded Description Entry Form

F. The application's "Share" button (shown in figures 2 and 3) shall open another form to allow the user to share the event with others via email or third party platforms (e.g. Facebook, Google Calendar, Apple's calendar application).

4.4. Nonfunctional Security Requirements

A. The application must encrypt all of the user's data including his/her events, tasks, username, password, and the login credentials for third party tools whose calendar events are accessed by this application.

B. The application shall keep each user's data separate and unviewable by other users.

5. Use Cases

This section summarizes a set of uses cases the event and task manager must support. They have been included on separate pages for improved readability.

5.1. Log in User

Use Case Name: Log in User
Goal: The user logs into the application so that s/he can access his/her account information and use the application.

User Action	System Action
The user attempts to access the application in the browser.	The system prompts the user to enter his/her username and password.
The user enters login information and submits it to the system.	The system verifies whether the user specified login credentials match any known credential. **System Action Variation #1:** If there is a match, the system gives the user access to the corresponding account's data and opens the application's home interface. **System Action Variation #2:** If the specified login information does not match any existing accounts, the application prompts the user that the specified login is invalid and allows the user to enter new login information (note this step would repeat with the user action).

5.2. Add a Third Party Calendar

Use Case Name: Add a Third Party Calendar
Goal: The user adds a calendar managed by another provider to this application for unified management.

User Action	System Action
The user clicks the button to add a third party calendar to their account.	The system displays the supported third party calendars (e.g. Google, Facebook, Apple) and asks the user which type of account s/he would like to add.
The user selects the type of account calendar (e.g. Google) s/he wants to add.	The system prompts the user to enter the login information for an account of the user specified type.
The user enters the login information for the account s/he wants to add and clicks the submit button.	The application attempts to verify the login information with the third party provider. **System Action Variation #1:** If the information is successfully verified, the system alerts the user that the account was successfully added and syncs the data from the third party. **System Action Variation #2:** If the account is not successfully verified, then it prompts the user that the third-party login credentials were invalid and allows the user to enter credentials again (note this step would repeat with the user action).

5.3. Sort Asynchronous Tasks

Use Case Name: Sort Asynchronous Tasks
Goal: The user views the asynchronous tasks sorted according to a specified criteria.

User Action	System Action
In the asynchronous task viewer pane, the user clicks the "Sort" button.	The application drops down a menu showing the allowed sorting criteria (e.g. alphabetical by name, priority, task creation date).
The user selects the sorting criteria according to which s/he would like to see the tasks ordered.	The application hides the sorting criteria drop menu. The applications then sorts the tasks according to the user specified criteria. The application updates the user's view with the tasks appropriated sorted.

5.4. Mark Asynchronous Tasks Completed

Use Case Name: Mark Asynchronous Tasks Completed
Goal: The user updates an asynchronous task in the application to be marked as completed.

User Action	System Action
The user clicks on the task s/he wants to mark completed.	The application displays the task information, including a check box the user can check to mark the task completed.
The user clicks the "Completed" check box.	The application displays the message, "Are you sure you want to mark this task as completed?". The available options being either "Yes" or "No".
User Action Variation #1: The user clicks "Yes". **User Action Variation #2:** The user clicks "No".	**System Action for User Variation #1:** The system prompts the user to enter a task completion time with the default being the current time. **System Action for User Variation #2:** The operation terminates, and the application returns the user to the task view menu, without updating the task.
The user enters the task completion time and clicks the "Next" button to continue.	The system marks the task completed with the user specified time. It returns to the task view menu and removes the task from the list of uncompleted tasks and adds it to the list of completed tasks.

5.5. Create Calendar Event

Use Case Name: Create Calendar Event
Goal: The user creates a new calendar event for his/her account.

User Action	System Action
The user clicks on the button to create a new calendar event.	The application displays the create calendar event form similar to what is shown in figure 2.
The user enters the calendar event information and clicks the "Create" button.	The system checks the calendar event information for completion. **System Action Variation #1:** If the form information is incomplete, it prompts the user regarding the missing information and allows the user to reenter the information. Note that for this variation, the use case remains on this step with the user able to repeat his/her action again. **System Action Variation #2:** The application adds the new calendar event to the user's account data and returns to the event calendar view with the new calendar event displayed.
Additional User Action Variation #1: While adding event information, the user clicks on the "Description" field to enter an event description. Note this variation can occur at any time the user is entering the form information so given its asynchronous nature, it is denoted separately,	**System Response to Additional User Variation #1:** The create event form lengthens to allow the user to enter an event description as shown in figure 3.