Aaryaman Akhouri , Taylor Mannion, Uday Talasila , Thomas Powderly

Project Milestone 2

Process Deliverable I

Our team plans to use the Incremental model for our project, FocusExtension. As such, there are certain requirements we need to prioritize for the next milestone. In the requirements elicitation from Milestone 1 of the project, both interviews had the interviewee stating they would prefer to have all of the productivity tools such as Pomodoro timers, task management, calendars, and reminders in one application. This was the fundamental difference between our project proposal and existing productivity apps. With that in mind, it is the top requirement for us to prioritize. In that requirement, there are sub-requirements of the features promised, so our Chrome extension needs to have Pomodoro timers, a to-do list for tasks, a calendar, and reminders.

Both of our interviewees also mentioned getting distracted while doing work. The first interviewee expressed a desire for certain websites to be unavailable to them as they were working. To curb this kind of sidetracking from happening, we would like our Pomodoro timer to block distractions while the user is working. We imagine that websites and applications like social media apps, YouTube, video games, Netflix, Reddit, and Amazon would be preset on the list of distractions to block, and the user could edit this list to their needs. With distractions blocked, users can keep problematic impulses and temptations in check while working.

The second interviewee stated they would like a productivity app to have somewhere to put notes. While this seems like a very basic feature and there are many note-taking apps, word processors/text editors, and even plain paper notebooks that fulfill this purpose, the benefit of adding this feature comes from it being integrated with the rest of the productivity tools. As stated, the goal of our app is to have one centralized application that handles all of users' productivity needs, so they don't have to switch between apps. Therefore, we hope to add a section in our extension where users can jot down notes and possibly link those notes to events and tasks they've created.

Requirements Specification

Actor 1: I am a student user of FocusExtension who wants to have an event in my calendar with automatic reminders.

Acceptance Criteria:

1. **Given** the user have their reminder preferences set
2. **When** the user places an event into the calendar
3. **Then** the system will create reminders at the specified intervals before the event.

The subtasks and associated functional points for this story would be:

Establish a list of reminder objects that have times linked to them (5)
Establish a cycle to run through the reminder objects every 15 minutes to check if any should occur (5)
Create a method of displaying the reminder and then deleting it once it is done (7)

Actor 1 - I am a student user of FocusExtension who wants to have specific websites blocked when my Pomodoro timer is on.

Acceptance Criteria:

1. **Given** The user has input the URLs of websites to be blocked into the system.
2. **When** the user creates a Pomodoro timer
3. **The** system will then detect the IP addresses of the specified websites and intercept signals sent from them, essentially blocking them.

The subtasks and associated functional points for this story would be:

Establish a list of blocked URLs (3)
Establish a cycle that checks every second for incoming pings from the blocked URL IPs (8)
Establish a method of intercepting the received signals from those websites, causing the browser to display an error message for those URLs (7)

Actor 2: I am a professional who uses FocusExtension and wants reminders for the tasks on my to-do list.

Acceptance Criteria:

4. **Given** The user has enabled notifications within the application
5. **When** the user creates a task on the to-do list
6. **Then** the system will provide daily reminders of tasks to do.

The subtasks and associated functional points for this story would be:

Establish a database of app preferences that can be accessed by the todo list (3)
Establish a method for checking the database to see whether reminders should be created (3)
Establish a method for displaying reminders at the start of every day (5)

Actor 2 - I am a professional who uses FocusExtension and wants to have specific deadlines for the tasks on my to-do list.

Acceptance Criteria:

7. **Given** The user has set up the application.
8. **When** the user creates a task on the to-do list and sets a deadline for it
9. **Then** the system will automatically add a calendar event at the deadline to match the item on the to-do list.

The subtasks and associated functional points for this story would be:

Establish a to-do list object that contains task objects (3)
Establish an input field/buttons for taking user input for deadlines (5)
Establish a method for having calendar items link to to-do lists (4)

Requirements Analysis

Non-functional Requirements

1. In regards to performance, reminders should be displayed within 10 seconds of the time the user requested.
2. Regarding usability, the plugin should be easy to use with all options readily available to the user.
3. Regarding reliability, the system should have very few failures (200 MTBF) and the system should be able to be recovered with a mean downtime of 2 hours.
4. Regarding supportability, the system should be updated with every Chrome update to ensure the plug-in continues to work as expected.
5. Regarding implementation/constraints, the user is required to use Chrome as their browser to be able to use the plug-in.

Functional Requirements

1. Allow users to make a TODO list and check off items
2. Allow users to use a Pomodoro Timer to stay productive
3. Block certain websites when the Pomodoro timer is on a work session
4. A calendar where events on specific days can be added
5. Provide reminders if Chrome is open that a task is due soon

Use Cases

1. Use a Pomodoro timer to stay on track.

Main flow: The user will start the timer [S1], the timer will alternate between work periods and short breaks until it is time for a long break, and repeat [S2] until the timer is ended [S3].

[S1] User navigates to the Pomodoro Timer section of the extension and hits the start button.
[S2] The timer uses times of 25 minutes for a work period, 5 minutes for a short break, and 30 minutes for a long break, with 4 work periods before a long break.
[S3] User hits button to end timer.

2. The user creates a task on the to-do list.

Main flow: The user will be prompted to enter a name for the task and potentially a due date [S1]. The task will be saved with all other items in the to-do list.

[S1] Due date will be optional. Users can enter a deadline in MM/DD/YYYY format.

3. Users can block certain websites while the Pomodoro Timer is active.

Main flow: Users will have a window [S1] where they can add websites that they do not want to access [S2].

[S1] The window will be a recognizable tab in the extension that will display the list of currently blocked websites, an option to remove them, and an option to add others.

[S2] The User will press a + sign, enter a URL and press enter.

4. Users should get reminders that a certain task is due if it is coming up.

Precondition: A task should be due within 24 hours of the current time, Chrome should be open, and notifications should be enabled.

Main flow: Users will get an alert in Chrome that they have a task due at certain intervals leading up to the task [S1].

[S1] Intervals could be 24 hours before, 6 hours before, 1 hour before.

Alternative flows

[E1] No time is given on the task, in which case only two reminders may be given, one the day before, and one the day of.

5. Add an event to the calendar.

Main flow: The user enters the name of the event/item, date [S1], and time [S2], and it is added to the calendar [S3].

[S1] Date can be entered in MM/DD/YYYY format.

[S2] Time is optional.

[S3] The Event is created on the right day, and organized by time within that day.