

VS-Saturn Project Requirements

Goals:

- Time management using a Pomodoro schedule.
- Customizable task checklist to track progress.
- Non-intrusive notifications for time management and a snooze function.

Software elements:

- Timer: set some amount of time for work and breaks and switch from work timer to break timer and vice-versa.
- Snooze: delay current timer cycle for some user-specified amount of time.
- Reset: The user can reset the cycles back to the beginning if they want to start over.
- Task list: takes some strings as input and creates a checklist based on these strings
- Notification: on each change of timer cycle, gives some non-intrusive cue that the current timer is up
 - Change border color.

Quality measures:

- The timer should track time accurately and consistently.
- The task list should be able to convert the input strings into checklist items in a presentable way (i.e. no graphical errors).
- Notifications must be noticeable while maintaining non-intrusiveness. These should behave the same way on a consistent basis.
- The snooze button must be able to delay the timer for a set amount of time.
- Specifications for the time settings should be changeable by the user at any time, with immediate impact on behavior. These should be set to standard Pomodoro technique specifications by default.

Technical Requirements

1. Keep track of user's goals
 - 1.1. Checklist of "work" tasks
 - 1.1.1. Ability to add/edit/remove (or mark complete) tasks
 - 1.1.2. Visible during work periods
 - 1.2. Checklist of "break" tasks
 - 1.2.1. Ability to add/edit/remove (or mark complete) tasks
 - 1.2.2. Visible during break periods
 - 1.2.2.1. User can also add to this list while working
2. Keep user on-task (and on-schedule)
 - 2.1. Display progress within larger pomodoro cycle
 - 2.2. Display time remaining in current work/break period
 - 2.3. Ability to pause/resume current timer
 - 2.3.1. Keyboard shortcuts for these functions
 - 2.4. Ability to reset cycle progress if necessary
 - 2.5. Non-intrusive notification when work-timer expires
 - 2.5.1. Option to "snooze" timer for some amount of time (i.e. if the user is in the middle of something uninterruptible, they can snooze the timer to delay the start of the break period)
3. User configurability
 - 3.1. Users can choose...
 - 3.1.1. Whether the timer should continue if VSCode is minimized
 - 3.1.2. Whether confirmation is required before moving on to the next period when a timer expires
 - 3.1.3. How many tasks make up one larger pomodoro cycle
 - 3.1.4. How long work periods should be
 - 3.1.5. How long break periods should be
 - 3.2. User's settings are saved and reloaded when VSCode is quit & reopened.