Tasker

Lorenzo Gomez, Gemuele Aludino, Anton Ryjov

1. Introduction

Many people(including us) have creative pursuits; projects we want allocate time for; goals to accomplish. In our team, I write; Gem plays music. It's always been hard for us, and we suspect for many other people, to stay on task. More specifically, it's even harder to keep track of one's progress. The timers of our world—smartphones, desktops, etc—aren't designed to *accurately* time the tasks we do.

One of the reasons for this is that human error is inevitable. A writer like myself may set the timer to 30 minutes, but halfway through the writing session, I might go make tea. If one is doing it right, making tea takes time. I forget to stop my timer, which is supposed to keep me attached to the keyboard for 30 minutes. In the meantime, while time runs out, I'm setting up the pot, waiting for it to boil, take my tea bag to dip in, fetch milk, make toast. When I come back to my novel, the timer ran out!

Even though my timer counted 30 minutes, I only did 15 minutes of writing. This is not ideal. Tasker aims to solve this problem, or at the very least make our timers more accurate (hence mitigating human errors) by binding our timers to hardware.

2. Proposed System

With the explicit consent of the user, our system will track hardware interaction to provide a real-time timer that will only count down as the user is interacting with the peripheral device that they are using or interacting with at the moment—a writer would use a keyboard, a musician would be making noise. Not only would this time accurately measure our user's progress throughout their task, but over weeks and months our system can intelligently evolve into providing *personalized* data for our users so that they can more accurately assess how productive they are on a particular task.

We believe this system will add value to the user's productivity on a particular task; be a television writer or a musician.

1. *Functional Requirements*

Spec ID	Spec Type	Spec Description
In-001	Input	The user will input the amount of time they want to spend on this task. This task can be writing or composing music.
In-002	Input	The user will select what type of task they want to time themselves doing—writing, music or drawing.
InK-003	Input-keyboard	Temporarily buffer the user's interaction with keyboard for analysis.
Out-004	Output	Show how much time has been spent on the task(writing or composing music) compared to how much time the user has spent on this session.
InA-005	Input-Audio	Temporarily buffer the user's interaction with Audio/ Microphone for analysis.
Ink-007	Input-keyboard	Shortcut to make tasker appear in front of user—maximize tasker window.
Out-008	Output	Show the current state of Tasker's timer.
Out-009	Output	Show the current progress towards the set goal in a user-friendly manner. For example, "You have completed 50% of your task."
Out-010	Output	Show the current progress for a specific task(writing, composing music or drawing) for the past week.
Out-011	Output	Show the current progress for a specific task(writing or composing music) for the past month.

Out-012	Output	Show the current progress for a specific task(writing or composing music,) for the past year.
Out-013	Output	Plot in graphs the current progress for a specific task(writing, composing music, or drawing) for the past year.
Out-014	Output	Plot in graphs the current progress for a specific task(writing or composing music) for the past month.
Out-015	Output	Plot in graphs the current progress for a specific task(writing or composing music) for the past week.
Out-016	Output	Plot into graphs Milestones.
Out-017	Output	After two weeks of analyzing a task, Tasker can start personalizing the sensitivity of the timer.
PR-019	Output	Notify the user when they are done with their task. This means when productive time has reached the goal.
In-021	Input	The user will be able to create a commitment to play music, write or draw for an amount of time they'll specify. Such as from 1 hour/day to 4 times/week.
In-023	Output	The user will be able to view their commitments.
In-024	Input	The user may choose how to be notified at the end of their session.
PR-026	Process	Calculate the amount of keyboard activity during session for use in characters per minute.

Input-027	Input	Ask for the user’s consent before accessing any hardware peripheral—keyword or microphone.
PR-028	Process	Have an algorithm that accurately counts down the timer as the user interacts with keyboard—as they write.
PR-030	Process	Our application will only run on unix-based desktop.

2. *Non-Functional Requirements*

Spec ID	**Spec Type**	**Spec Description**
In-006	Input-Non	Timed Task(s) will be saved for future analysis.
PR-018	Process-Non	Proactively clear data that is not needed to ensure we take care of the user’s privacy when interacting with keyboard.
PR-020	Process-Non	Proactively clear data that is not needed to ensure we take care of the user’s privacy when interacting with microphone.
In-022	Process-Non	Tasker’s window will hide from screen when the timer starts.
Out-025	Output-Non	The user will be able to their progress on a per-session basis.
In-029	Process-Non	Allow the user to turn off consent for any hardware peripheral—keyboard or microphone.

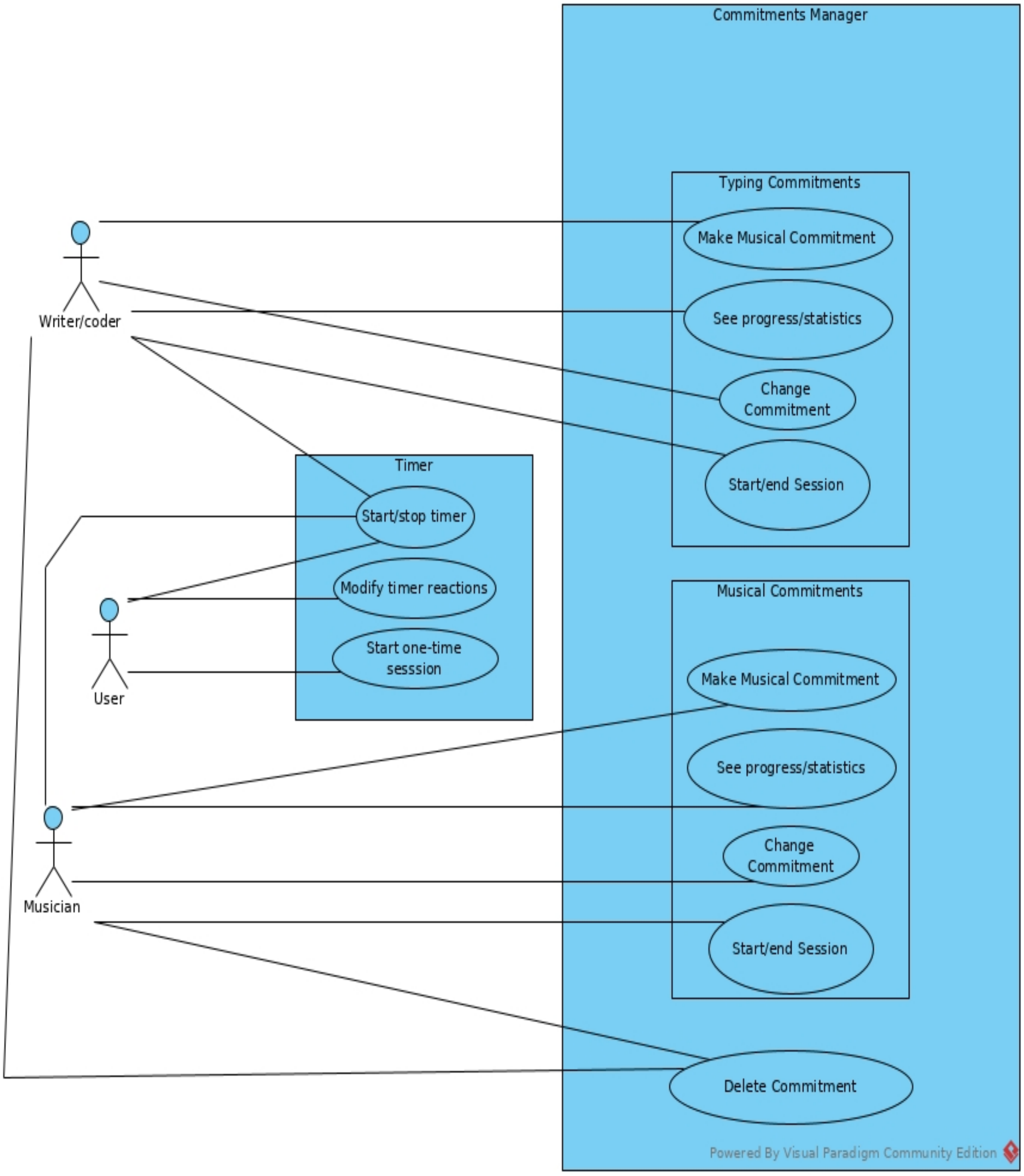
3. *System Models*

1. Use Cases

Use Case Name	MakeCommitment
Participating Actors	User OR Writer OR Musician
Flow Of Events	User opens the app, clicks the Make Commitment button, and is shown options for the commitment, including: • How much time User wants to commit to the task • Whether the timer for this commitment should be sensitive to audio, keyboard, or both
Entry condition	App open on home page
Exit conditions	User clicks OK or Cancel, or closes the app
Quality requirements	commitment data must be persistent

Use Case Name	ShowStatistics
Participating Actors	User OR Writer OR Musician
Flow Of Events	User opens a commitment and clicks Show Statistics The User is shown a gui displaying previous sessions attached to this commitment, each including; • The date of the session • The total time of the session • The productive time of the session • The unproductive time of the session
Entry condition	User clicks Show Statistics from a commitment page, User finishes a session
Exit conditions	User clicks Back, or closes the app
Quality requirements	If no session has been recorded yet, show an explanation to the user

2. Use Case Diagrams



3. Glossary

Session: *The total amount of time the user takes to accomplish a task. This includes productive time and unproductive time.*

Task: *This can be any task that the user may specify at the beginning of a session. This task can be a writing or a musical task.*

Commitment: *A long-term commitment to a task. For example; a writer may want to write 30 minutes a week.*

Productive time: *Time that we detect was spent on the task that the user is committing to.*

Unproductive time: *Time that the user does NOT spend on the task.*

Milestone: *A significant increase in user's performance on a specific commitment.*

X: *Positive reinforcement for when user stays committed.*

4. Reference

Visual Paradigm → https://www.visual-paradigm.com/

QT → https://www.qt.io/

C++

Apple's Health Sleep Tracker