Remote Work Discipline App System Requirements

CS 3704 - Intermediate Software Design - Fall 2024

Raunak Chitre
Computer Science
Virginia Tech
Blacksburg, Virginia, USA
raunakc24@vt.edu

Jeriah Valencia
Computer Science
Virginia Tech
Blacksburg, Virginia, USA
jeriahv@vt.edu

Shane Matthews
Computer Science
Virginia Tech
Blacksburg, Virginia, USA
shanem64@vt.edu

Ben Sullivan
Computer Science
Virginia Tech
Blacksburg, Virginia, USA
bens21@vt.edu

Extreme programming calculation of project velocity:

1. Providing hypothetical non functional requirements for the system - assign 2 function points each because this is a fairly easy task, so 2*5=10 function points in total
2. Providing hypothetical functional requirements for the system - assign 2 function points each because this is a fairly easy task, so 2*5=10 function points in total
3. Writing five formal use cases for system - somewhat challenging, so 4*5=20 function points, use case or sequence diagrams take a moderate amount of time, so 3*5=15 function points.

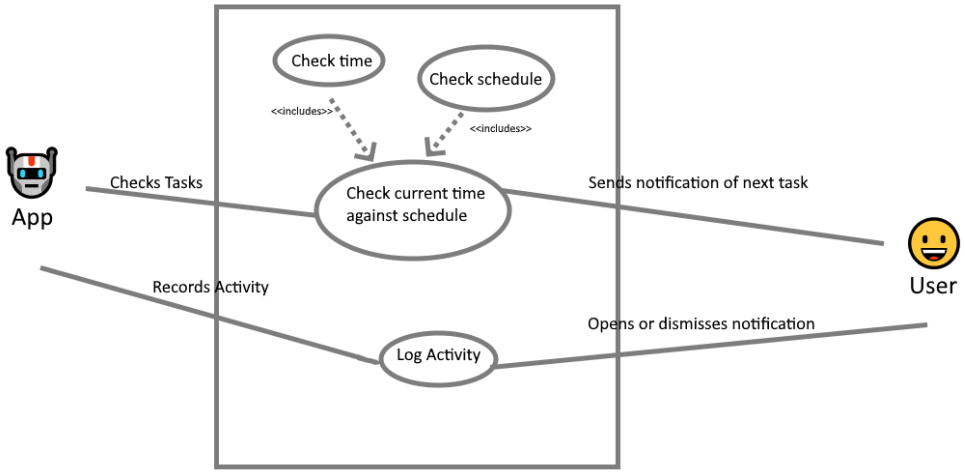
This is a total of 10+10+20+15 = 55 function points, with a majority of the function points coming from the third task.

Requirements Analysis:

1. Non-functional requirements: The remote work discipline app should have the following non-functional requirements:
    a. The vibration should be audible from a distance of at least 2 feet.
    b. The app should restart itself within 30 seconds of failing.
    c. The app should respond to suspicious movements within 2 seconds.
    d. The app should work regardless of whether the user has a stable internet connection.
    e. The app should not consume more than 5% of the phone's battery for each hour of use.
2. Functional requirements: The remote work discipline app should have the following functional requirements:
    a. The app should store data about user activity from the past hour, day, week, and year.
    b. The app should have a break mode, where vibrations and alerts can be temporarily paused upon the request of the user.
    c. There should be a points system based on user behavior, motivating users to maximize their productivity.

d. There should be a networking feature so that users can compete with each other through the app, with a leaderboard.
e. The app should make annoying noises for serious drops in productivity to redirect the user.

3. Use Cases:
    a. Task Reminder Notification
        i. Preconditions
            1. User must have the app installed and properly configured
            2. User must have set up tasks and reminders
        ii. Main Flow
            1. App checks the current time against user's schedule
            2. App determines a task is due within a certain time frame
            3. App sends a notification to user's phone informing them of the task
        iii. Subflows
            1. [S1] If the user dismisses the notification, the app logs the dismissal
            2. [S2] If the user opens the notification, the app marks the task as started
        iv. Postconditions
            1. User is reminded to complete their task
            2. App logs the user's interaction with the notification
        v. Alternative Flows
            1. [E1] If the app fails to send the notification, it retries after a set interval of time



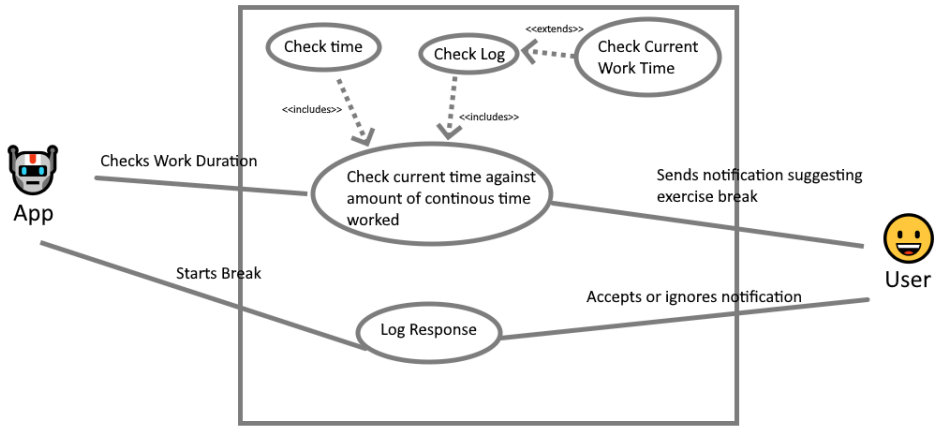
    b. Exercise Break Reminder Notification
        i. Preconditions
            1. User must have the app installed and properly configured
            2. User must have set up tasks and reminders
        ii. Main Flow

1. App records user's work duration
2. App detects that the user has been working continuously for a set period of time
3. App sends a notification to user's phone, suggesting a break for exercise

iii. Subflows
1. [S1] If the user accepts the break reminder, the app starts a break timer
2. [S2] If the user ignores the reminder, the system logs the inactivity

iv. Postconditions
1. User is prompted to take a break and exercise
2. App logs the user's response to the notification

v. Alternative Flows
1. [E1] If the app fails to send the notification, it retries after a set interval of time



c. Break Mode Activation

i. Preconditions
1. User must have the app installed and properly configured

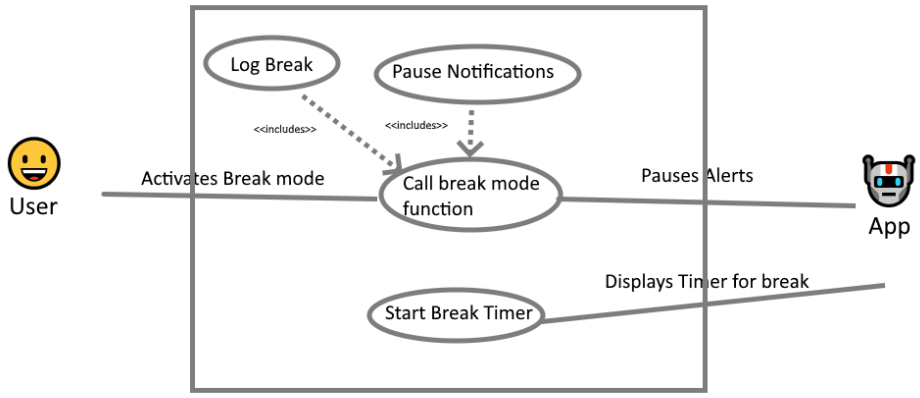
ii. Main Flow
1. User activates break mode through the app
2. App pauses all vibrations and alerts
3. App displays a timer for break duration

iii. Subflows
1. [S1] If the user sets a specific break duration, the app resumes normal operations after the timer expires
2. [S2] If the user manually ends the break, the app resumes normal operations immediately

iv. Postconditions
1. User can take a break without interruptions from the app
2. App logs the break mode activation and duration

v. Alternative Flows

1. [E1] If the app fails to start break mode, it retries after a set interval of time



d. Points System and Leaderboard
    i. Preconditions
        1. User must have the app installed and properly configured
        2. User must have an account to track points
    ii. Main Flow
        1. App monitors user behavior and assigns points based on productivity
        2. User views their points and ranking on the leaderboard
        3. App updates leaderboard in real-time based on user activity
    iii. Subflows
        1. [S1] If the user completes tasks in a shorter amount of time than expected, the app awards bonus points
        2. [S2] If the user views the leaderboard, the app displays the top users and their scores
    iv. Postconditions
        1. User is motivated to maximize productivity through the points system
        2. App maintains an up-to-date leaderboard
    v. Alternative Flows
        1. [E1] If the app fails to update points or the leaderboard, it retries after a set interval of time
e. Data Storage and Retrieval
    i. Preconditions
        1. The user has the app installed and properly configured
    ii. Main Flow
        1. App continuously monitors and logs user activity
        2. User requests to view their activity data

3. App retrieves and displays data for the specified time period (hour, day, week, year)

iii. Subflows
1. [S1] If the user requests data for a specific period, the app filters and displays the relevant data
2. [S2] If the user requests a summary, the app provides an overview of the relevant data

iv. Postconditions
1. User views their activity data for the selected time period
2. App logs the data retrieval request

v. Alternative Flows
1. [E1] If the system fails to retrieve the data, it prompts the user to retry

Requirements Specification:

1. User Stories:
   a. As a software engineer that works remotely, I want more discipline so that I can move up in my career.
      i. App only needs basic functionality: 5 function points to implement

      The app should include notifications in real time to provide focus, should be able to track the work habits of the user and provide feedback on productivity. The user should be able to view weekly and monthly productivity stats.
   b. As a college student completing homework in my dorm room, I want to block out distractions so that I can complete my work as efficiently as possible.
      i. App needs an additional feature that blocks background noise, 10 function points to implement

      The app needs to have a feature that blocks noise and mute distractions in the background. The user should be alerted of any distractions that have been found such as when the phone is used. The app should show user productivity and how long the user has been focusing.
   c. As a lawyer transcribing court documents, I want to reach a quota for how much work I do so that I can avoid being laid off.
      i. App needs to be able to measure work using clues such as keyboard taps, this may be more difficult to implement so 15 function points to implement.

      The app should be able to create daily summaries of productivity in relation to a set quota, the user should be given notifications if they are lacking on their quota. The app should track keyboard and mouse or any interactive activity to track how much progress the user has made.
   d. As a job candidate with an online assessment, I want to concentrate as much as possible on the exam so that I can maximize my chances of passing the interview.
      i. App only needs basic functionality: 5 function points to implement.

      The app should have a focus mode that takes away distractions. The user can customize how long focus mode runs for, and during this time only important notifications should be given.