FocusBot

GROUP 09

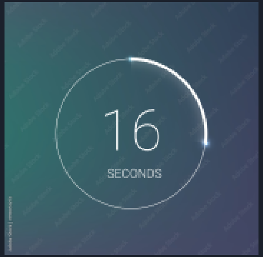
Michael Pang, Ajay Seethana,
Matthew Topping, Elijah Tynes

Original Problem Statement

Maintaining a high level of output over a long period of time remains a challenge in the software engineering field. Health factors such as vitamin intake, sleep, and energy levels play a huge role in the overall efficiency of software engineers. Creating a way for developers to know when they should take a break to hydrate, go for a walk, or stretch is essential for improving efficiency. This is what our solution attempts to solve.

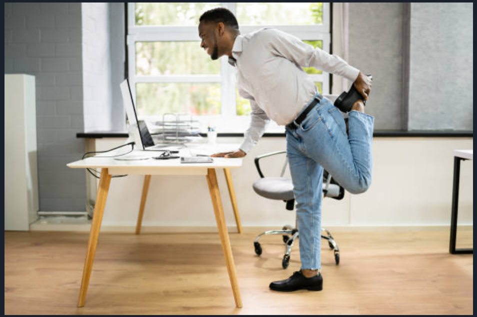
Explanation of Solution

- FocusBot web/mobile application to monitor software engineer productivity
- Tells user to take a break when productivity decreases or after a long duration of time
    - Monitors user activity (keystrokes, clicks, etc.)
    - Provides a lockout timer to ensure the user takes a sufficient break to reset
- Integrate health metrics from smart watches to advise moving / stretching



Rationale for Solution

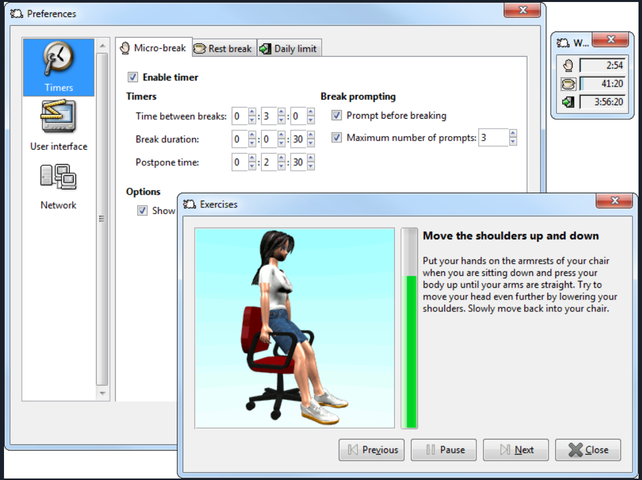
- More efficiency
- Higher quality of life for employees, as they receive as-needed breaks
- Help employees adopt a healthier lifestyle while working remotely
- Study by Cigna Healthcare (2023):
  - Declines in physical and mental health leads to huge losses
    - Total cost of UK economy of lost output among working-age people due to ill health was around $180 billion per year (~7% of GDP)



Related Works

Similar applications:

- WorkRave
- RescueTime
- Stretchly
- StandApp
- EyeLeo



☑ Enable timer

Timers

Time between breaks: 0 : 3 : 0

Break duration: 0 : 0 : 30

Postpone time: 0 : 2 : 30

Break prompting

☑ Prompt before breaking

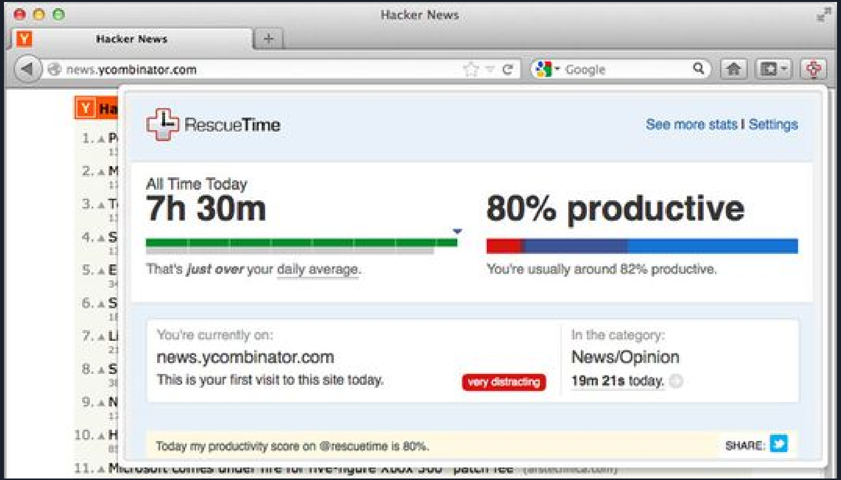
☑ Maximum number of prompts: 3

Options

☑ Show

Move the shoulders up and down

Put your hands on the armrests of your chair when you are sitting down and press your body up until your arms are straight. Try to move your head even further by lowering your shoulders. Slowly move back into your chair.



All Time Today

7h 30m

That's *just over* your daily average.

80% productive

You're usually around 82% productive.

You're currently on:

news.ycombinator.com

This is your first visit to this site today.

In the category:

News/Opinion

19m 21s today.

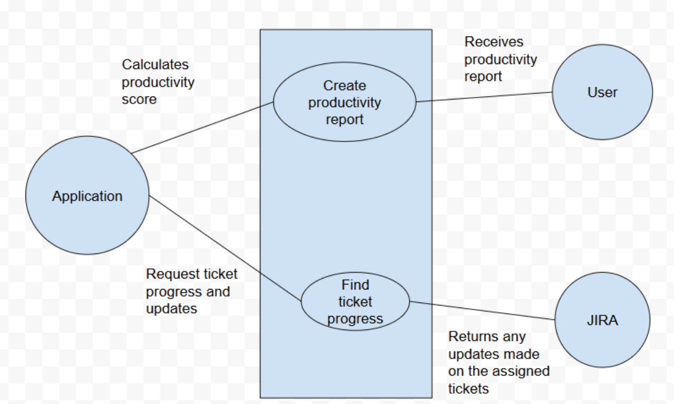
Today my productivity score on @rescuetime is 80%.

Use Case 1 - Productivity Tracker

[S1] Application will track the productivity of the user

[S2] Productivity will be tracked using the amount of time spent on completing tickets and how active the user is while at the computer

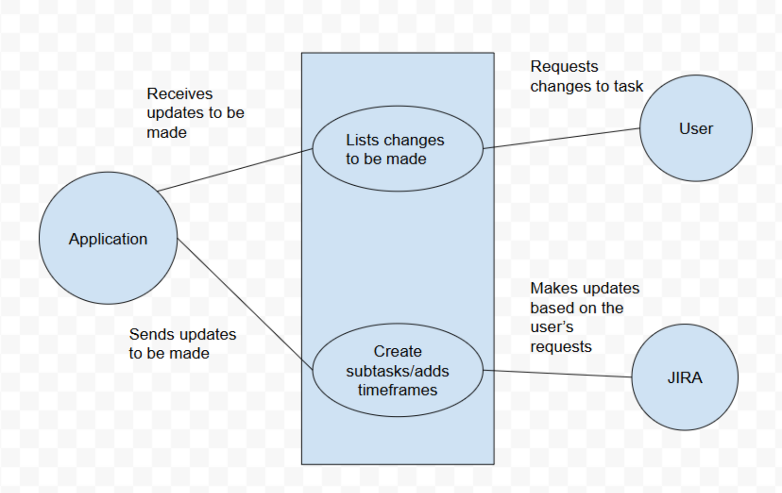
[S3] A score will be given to the user per period of time to praise/give advice to the user in terms of how well they've done over that period


Calculates productivity score
Create productivity report
Receives productivity report
User
Application
Request ticket progress and updates
Find ticket progress
JIRA
Returns any updates made on the assigned tickets

Use Case 2 - Access & Split Up Tasks

[S1] Users will be able to access, update, and split up tasks assigned on Jira through the application

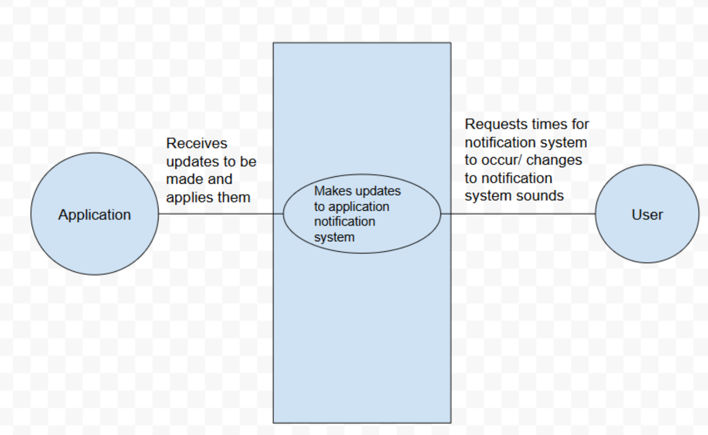
[S2] The user will be able to assign time frames for them to complete each subtask


Receives updates to be made
Application
Sends updates to be made
Lists changes to be made
Create subtasks/adds timeframes
Requests changes to task
User
Makes updates based on the user's requests
JIRA

Use Case 3 - Ability to Customize the Reminder Functionality

[S1] Users will be able to choose specific times that will vary in length based on the different amount of time from the previous break

[S2] There will also be different notification sounds that can be set for different notification types

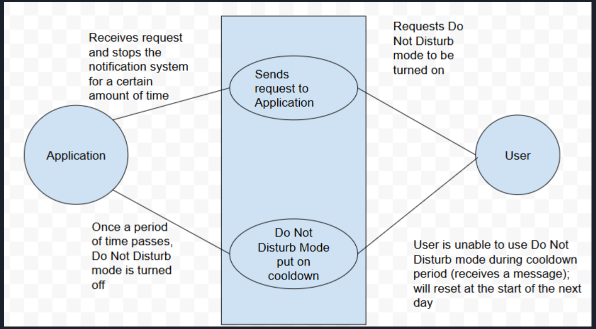

Application
Receives updates to be made and applies them
Makes updates to application notification system
Requests times for notification system to occur/ changes to notification system sounds
User

Use Case 4 - Do Not Disturb/Meeting Mode

[S1] Users will be able to set a Do Not Disturb/Meeting mode that will prevent the application from sending break time notifications

[S2] The mode will automatically turn off after a certain amount of time as to prevent the user from avoiding all break notifications

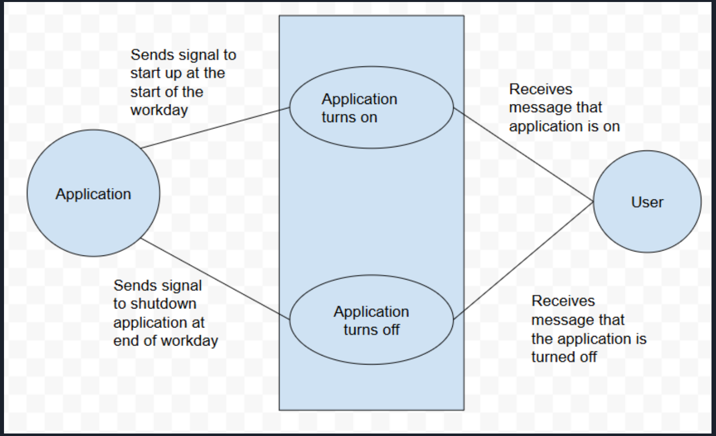
[S3] The mode will have a temporary cooldown period after use before it can be used again, with a limit to how much time that can be used per day (resets per day)


Receives request and stops the notification system for a certain amount of time
Sends request to Application
Requests Do Not Disturb mode to be turned on
Application
User
Once a period of time passes, Do Not Disturb mode is turned off
Do Not Disturb Mode put on cooldown
User is unable to use Do Not Disturb mode during cooldown period (receives a message); will reset at the start of the next day

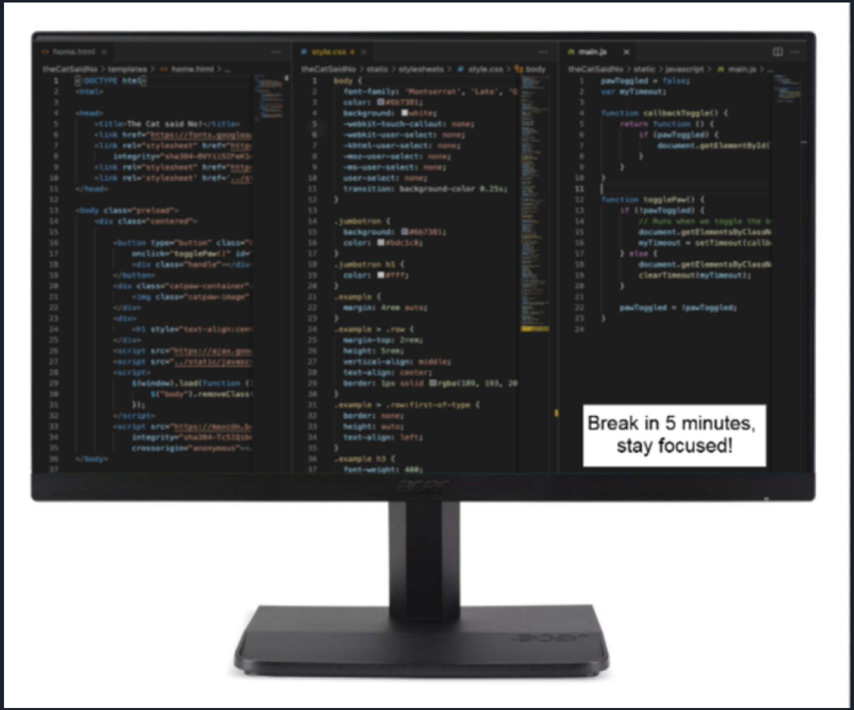
Use Case 5 - Application Automatically Starts/Stops

[S1] The application will automatically enable itself at the start of the workday

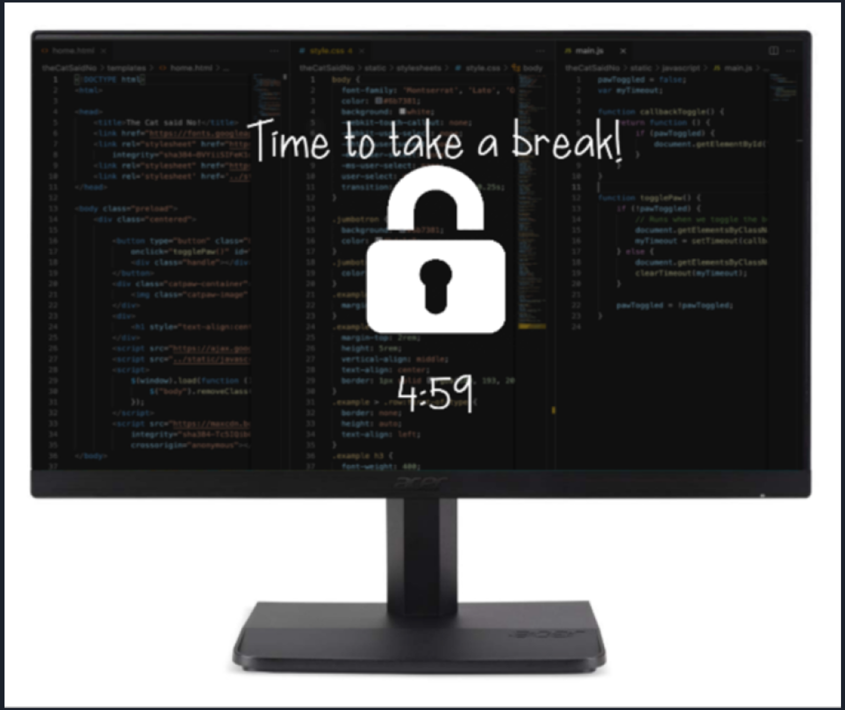
[S2] Additionally, it will shut itself down once the workday is over


Sends signal to start up at the start of the workday
Application turns on
Receives message that application is on
Application
User
Sends signal to shutdown application at end of workday
Application turns off
Receives message that the application is turned off

Visualized User Interface



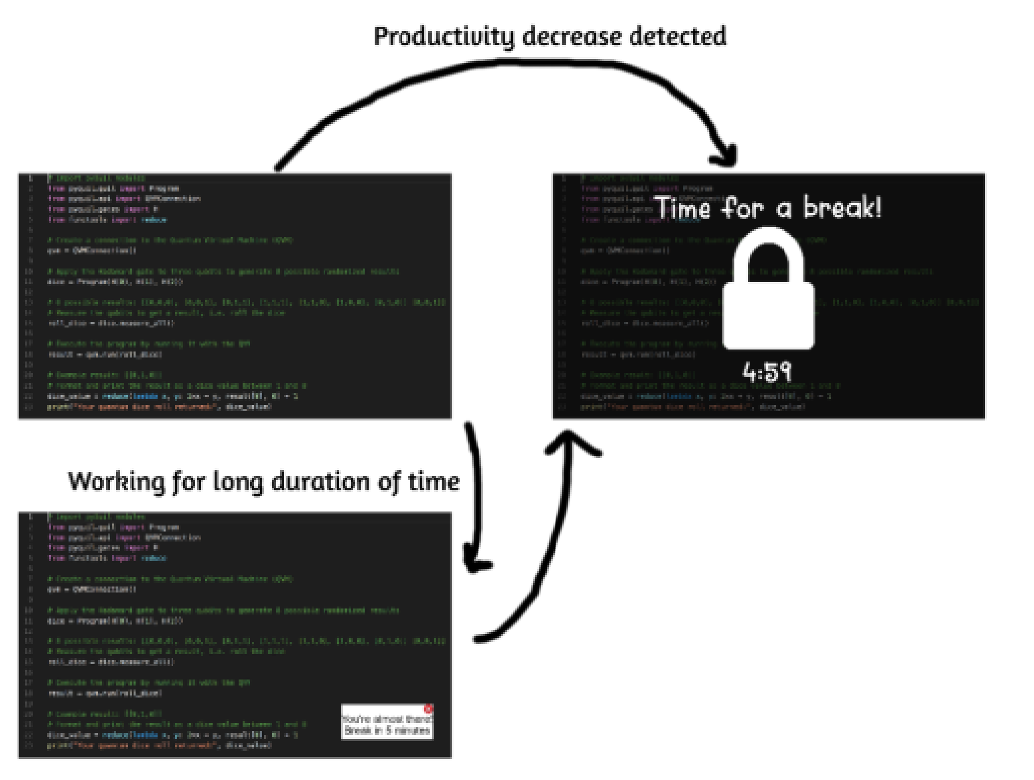
Break in 5 minutes, stay focused!



Time to take a break!

4:59

Visualized User Interface (2)



Productivity decrease detected
Time for a break!
4:59
Working for long duration of time
You're almost there!
Break in 5 minutes

```
# Import pyQuil modules
from pyquil.quil import Program
from pyquil.api import QVMConnection
from pyquil.gates import H
from functools import reduce

# Create a connection to the Quantum Virtual Machine (QVM)
qvm = QVMConnection()

# Apply the Hadamard gate to three qubits to generate 8 possible randomized results
dice = Program(H(0), H(1), H(2))

# 8 possible results: [[0,0,0], [0,0,1], [0,1,1], [1,1,1], [1,1,0], [1,0,0], [0,1,0]] [0,0,1]]
# Measure the qubits to get a result, i.e. roll the dice
roll_dice = dice.measure_all()

# Execute the program by running it with the QVM
result = qvm.run(roll_dice)

# Example result: [[0,1,0]]
# Format and print the result as a dice value between 1 and 8
dice_value = reduce(lambda x, y: 2*x + y, result[0], 0) + 1
print("Your quantum dice roll returned:", dice_value)
```

Future Work

- Smartwatch Integration
  - Track health information of the user
- Long term pattern recognition and reporting
  - AI
- Eye Tracking

Limitations

- Privacy Concerns
  - Since FocusBot monitors developers in real-time through different metrics it could raise concerns of personal privacy.
  - Developers might perceive the system as invasive, which could lead to mistrust or resistance from developers
- One Size Fits All
  - All developers are different. A system that works better for one developer, might not work better for another.
- Notification Fatigue
  - Frequent notifications, even if well-intentioned, could overwhelm developers, leading to notification fatigue and diminished productivity.
- Measuring Success
  - Determining whether FocusBot effectively enhances productivity can be difficult, as improvements in focus or well-being are subjective and vary between individuals.

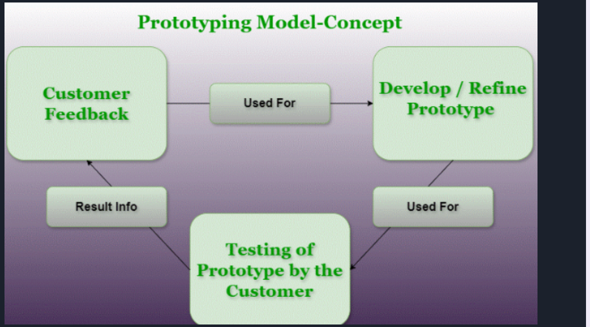
Processes and Tools Used

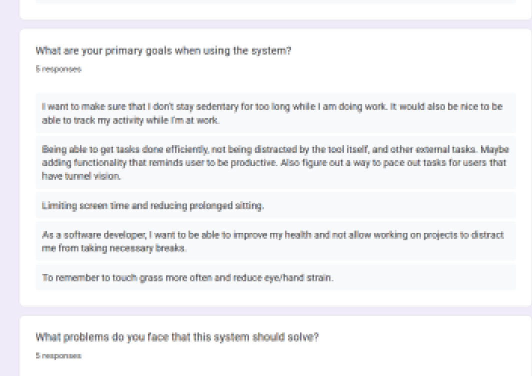
Processes:

- Agile Software Engineering Process
  - Prototyping Model
- Requirements Engineering Process
  - Elicitation -> Analysis -> Specification -> Prioritization

Tools:

- GitHub
  - Version control
- Paint/Pixlr
  - Prototyping


Prototyping Model-Concept
Customer Feedback
Used For
Develop / Refine Prototype
Used For
Testing of Prototype by the Customer
Result Info



What are your primary goals when using the system?

5 responses

I want to make sure that I don't stay sedentary for too long while I am doing work. It would also be nice to be able to track my activity while I'm at work.

Being able to get tasks done efficiently, not being distracted by the tool itself, and other external tasks. Maybe adding functionality that reminds user to be productive. Also figure out a way to pace out tasks for users that have tunnel vision.

Limiting screen time and reducing prolonged sitting.

As a software developer, I want to be able to improve my health and not allow working on projects to distract me from taking necessary breaks.

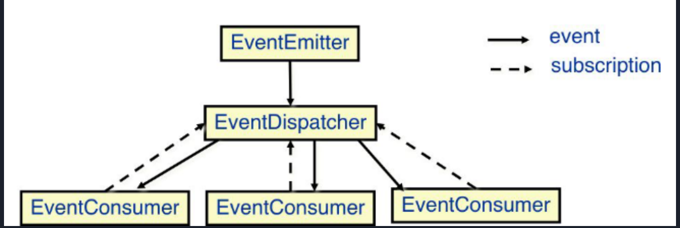
To remember to touch grass more often and reduce eye/hand strain.

What problems do you face that this system should solve?

5 responses

Things We Learned

- Applying High-Level Design and Low-Level Design Patterns to the Software Engineering Process
  - Event-based Architecture
  - Behavioral Design Pattern Family
- Applying the Requirements Engineering process to a real-life project
  - Different methods for Requirement Elicitation
  - How to perform Requirement Analysis
    - How to create formal use cases



Questions?