# Alpha Team

## Remote Work Discipline App: A Productivity Enhancement for Remote Software Engineers

By: Raunak Chitre, Ben Sullivan, Jeriah Valencia, and Shane Matthews

# Problem

## The Challenges of Remote Work for Software Engineers

- **Distractions**
  - People working remotely can be exposed to a variety of distractions that are not present in the office, such as other family members or pets in the home.
- **Accessibility**
  - Working remotely makes it more difficult to reach out to supervisors and coworkers when issues arise.
- **Isolation**
  - People working remotely lose opportunities for face to face collaboration with coworkers and miss out on company social events and networking opportunities that could benefit their careers.
- **Lack of resources**
  - Many homes do not include office space and materials that are needed to create a work environment.
- **Laziness**
  - Working from home allows people to live a more sedentary lifestyle, as daily physical activity such as walking or driving to work is no longer necessary.

# Proposed Solution

**Remote Work Discipline App**

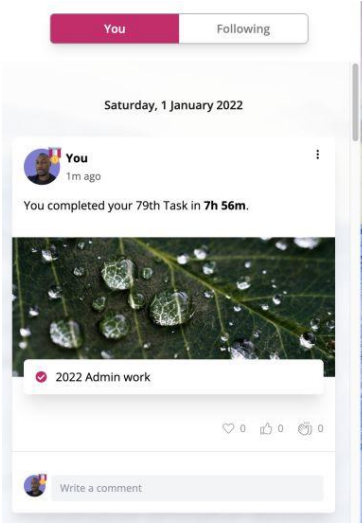
- Phone will vibrate to prompt the user to perform different tasks
  - Taking breaks to exercise
  - Can identify when someone is working on a computer or scrolling through a phone
  - Monitor distractions, such as background noise
  - Monitor work environment, such as room temperature
  - Monitor ergonomics, such as the person’s posture while working

# How The Solution Improves Software Engineering

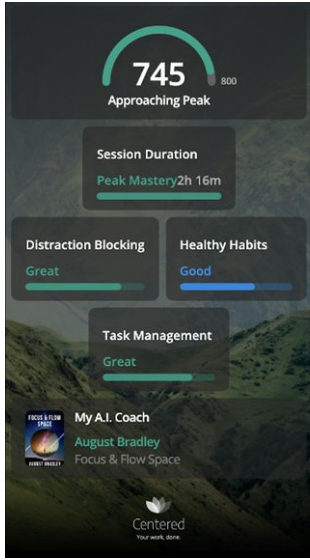
- **Optimizes Crucial Development Time**
  - ensures that the maximum amount of remote work time is put toward project development in an environment full of distractions
- **Improves Organizational Planning**
  - gives engineers the ability to more easily determine how to spend their time at work
- **Automates Administrative Overhead**
  - systematizes what is usually a complex part of a supervisor's role in the engineering process

# Related Work: Centered App

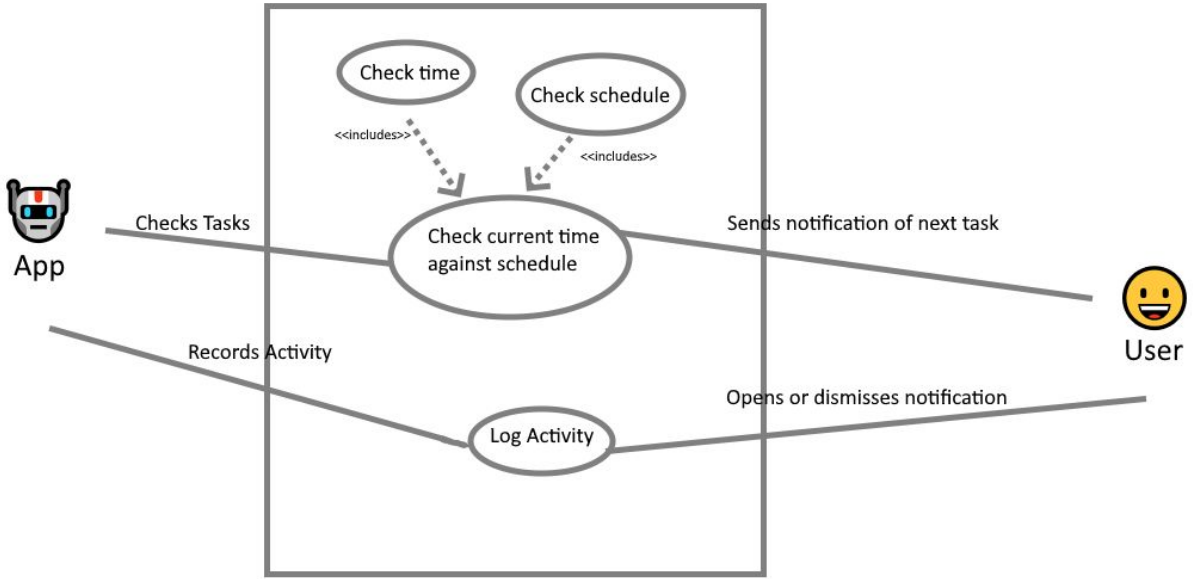
- A gamified task-organization app
- Considers time for breaks
- Minimizes distractions
- Does not feature environmental monitoring



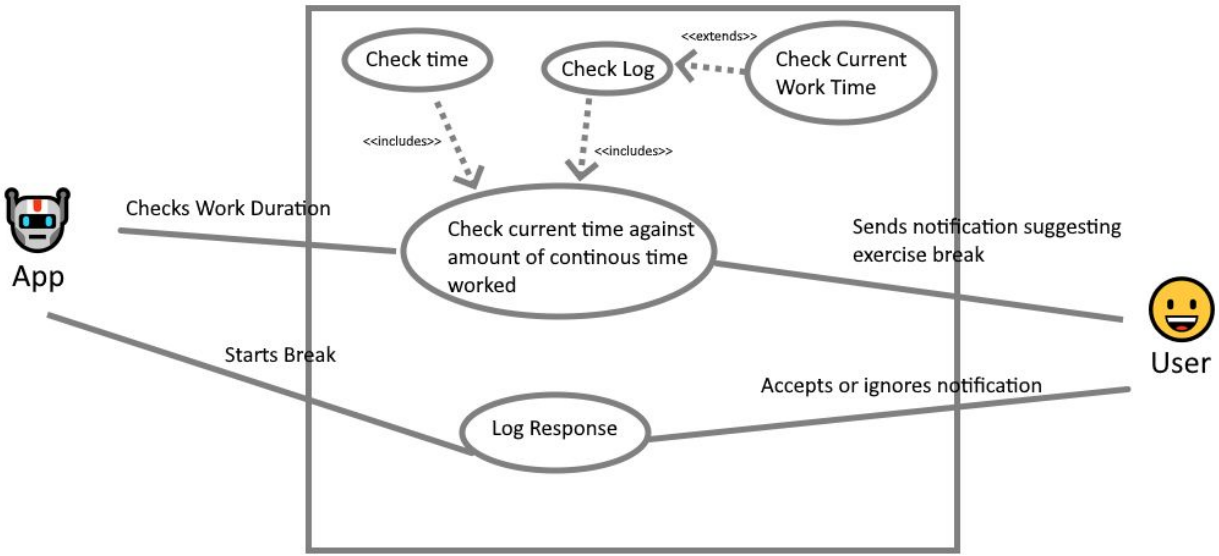




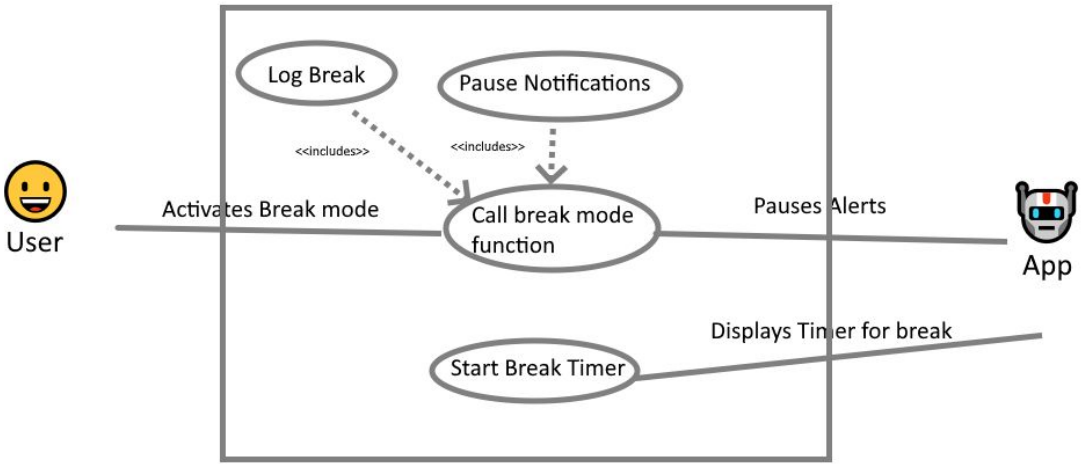
# Task Reminder Notification Use Case



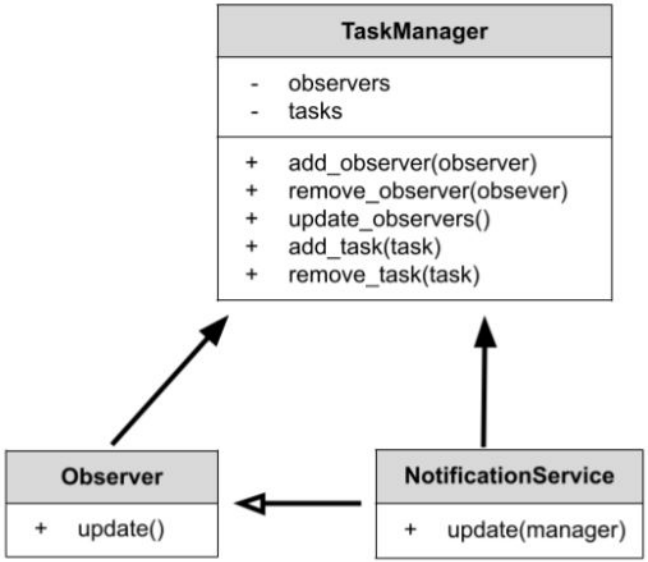
# Exercise Break Reminder Notification Use Case



# Break Mode Activation Use Case



# Low Level Design



# Class Concepts Used for the Project

- **Extreme programming** - estimating the project velocity for the next milestone
  - useful because this helped the team prepare for future milestones
- **Use Cases** - describing different situations in which our app would be used
  - useful because if we wanted to actually implement the app, we would already have some considerations significantly planned out in our design
- **Git version control** - accessing assignments related to the project through Git
  - unuseful because we did all of the actual work in Google slides and Google docs, where the team would work synchronously

# Future Work

- Actually implement the app
- Test the app on users
- Gather feedback from the users
- Improve the app based on feedback and repeat the cycle
- Once in a workable state, the app could be fully released