

CS25 Final Presentation: FocusBot

Dinachi Utah, Ruba Babiker, Alfredo Rodriguez, Minahil Malik,
Sophia Nadasy

Original Problem Statement

Problem of dwindling focus among students during lessons.

Subsequent impact on academic performance and overall growth.

Root Causes

- Distraction from digital gadgets.
- Personal challenges hindering concentration.
- Unengaging lesson delivery methods.

Current Solutions

- Inadequacy in addressing the core issues.
- Lack of universal effectiveness.
- Insufficient in catering to individual needs.

Call for Improvement

- Need for innovative, easy-to-adopt solutions.
- Aim for enhanced student engagement and attention.

Ultimate goal: Enriched learning experience for all.



Explanation & Rationale for Proposed Solution

The focus bot is an AI-driven tool designed to remove distractions by employing real-time monitoring and personalized feedback, which would foster improved focus and engagement.

Use Case 1: User Wants to Set a Pomodoro Cycle

1 Preconditions

Bot has a timer embedded in it

2 Main Flow

User will request a cycle and provide how many cycles they want to do, how long each break is, and how long each focus time is

3 Subflows

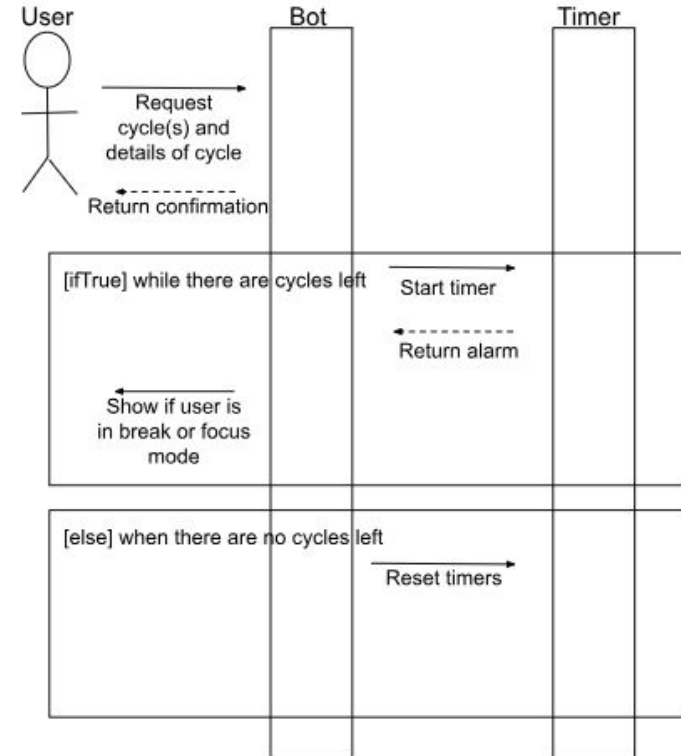
[S1] User provides number of cycles and time for breaks and focus time

[S2] Bot will return a confirmation

[S3] Bot will start pomodoro cycle and ring alarm once timer is up and show the user if they are in break or focus mode

4 Alternative Flows

[E1] No cycles in pomodoro are left so the bot will reset it all



Use Case 2: User Gets Distracted

1 Preconditions

The user is currently making use of a study cycle or has the bot active

2 Main Flow

[S1] The user is studying but gets distracted or goes inactive for a decent amount of time

[S2] The bot will show a small pop-up window asking the user if they are still studying or asking them to get back to studying if they are distracted

[S3] If the user is still active, they can close the pop up. The bot will proceed to count any time spent in unrelated tasks as distracted time

3 Subflows

[S1] The user is either not paying attention or is not at their computer.

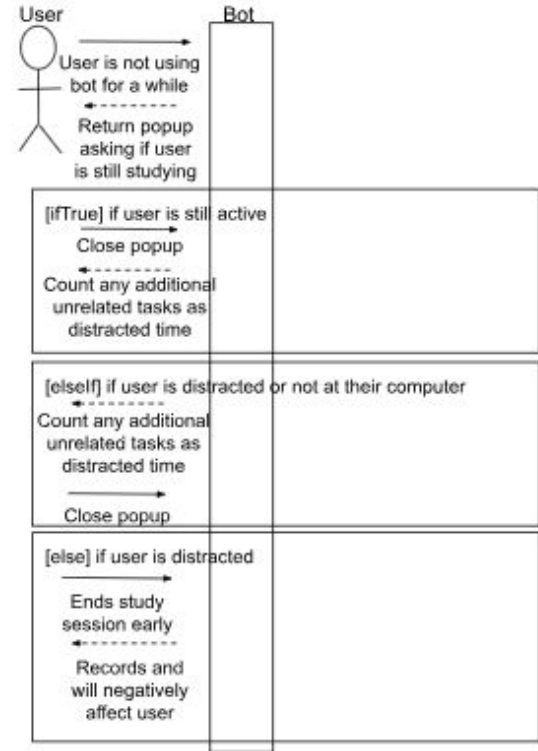
[S2] The user will not close the pop-up and all the time spent with the pop-up active will either count as distracted time or inactive time.

[S3] Once the user is back and closes the pop-up, the bot will stop counting the time as distracted or inactive if the user gets back to studying.

4 Alternative Flows

[E1] The user gets distracted, gets the pop-up alert, but ends the study session early instead of going back to studying.

[E2] The bot takes note of this, which will negatively affect the user's report.



Use Case 3: User Wants Performance Feedback

1 Preconditions

User must have studied while using the bot at least once before

2 Main Flow

[S1] User will ask the bot to make a report on their recent study sessions

[S2] Bot will take look at the sessions and take into account the focused study time, any moment the user got distracted by unrelated tasks, and how many of their current tasks they completed or worked on

[S3] The bot will return a visibly appealing report with everything the user got distracted with, how often they got distracted, and how much actual studying they got done while outlining the ost concerning problems in their study habits. The bot will also provide techniques or technologies so the user can make better use of their study cycles and get distracted less.

3 Subflows

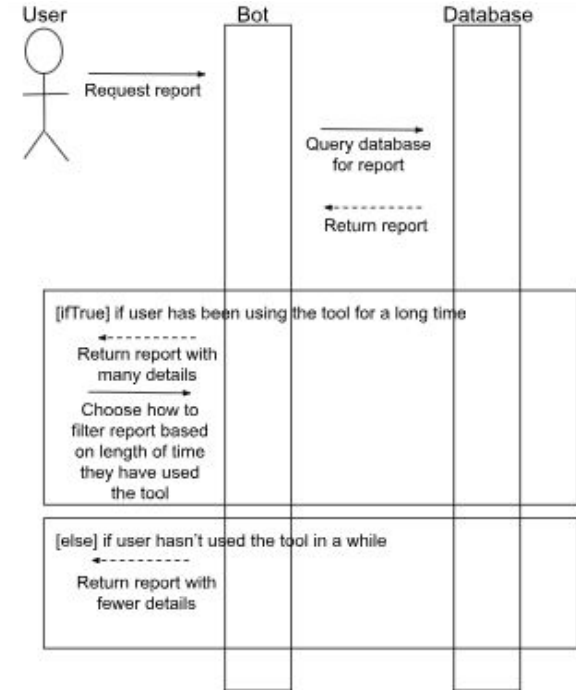
[S1] The user has been using the tool for a long time.

[S2] The user can choose to filter their report by the most recent week, month or longer depending on how long they have been using the tool.

4 Alternative Flows

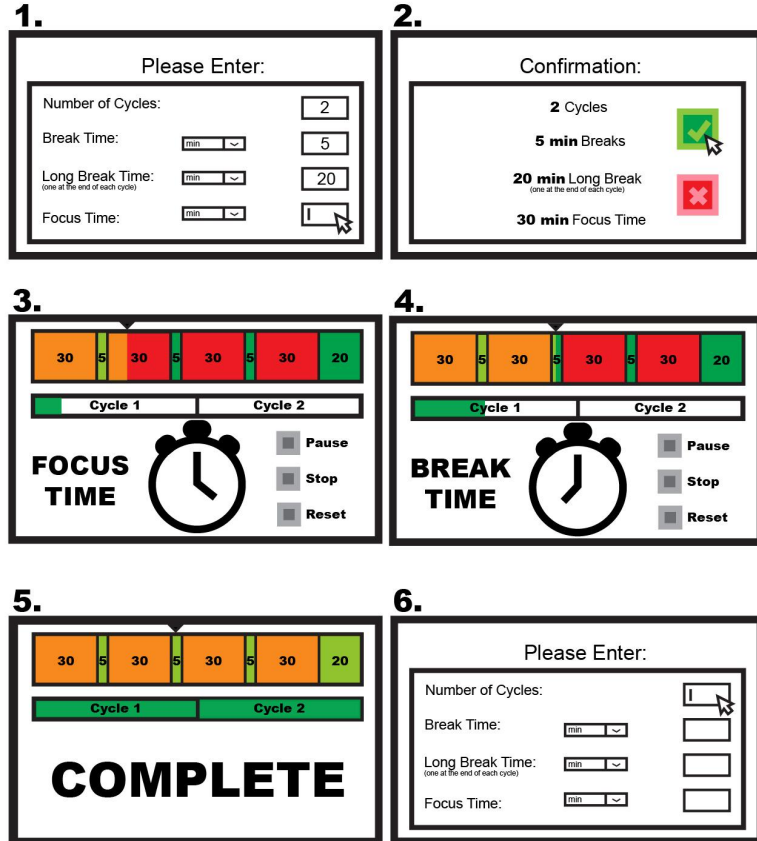
[E1] The user has not used the bot in a long time.

[E2] The bot might still provide feedback based on the information it has gathered but it might not reflect the current study habits of the user.



Visual Representation of Project (Mock UI)

Team CS25 Storyboard User Primary Task



Please Enter:

Number of Cycles:

2

Break Time:

min



5

Long Break Time:

(one at the end of each cycle)

min



20

Focus Time:

min



|



Confirmation:

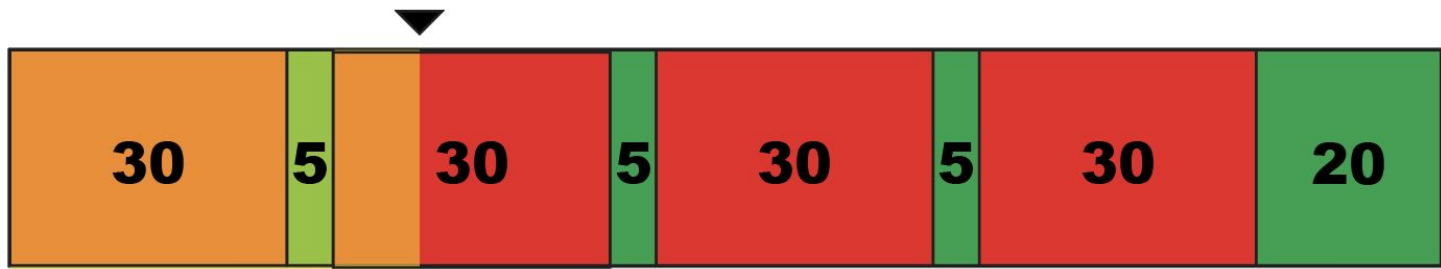
2 Cycles

5 min Breaks

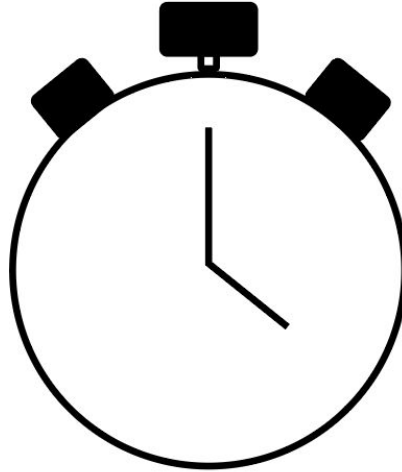
20 min Long Break
(one at the end of each cycle)




30 min Focus Time

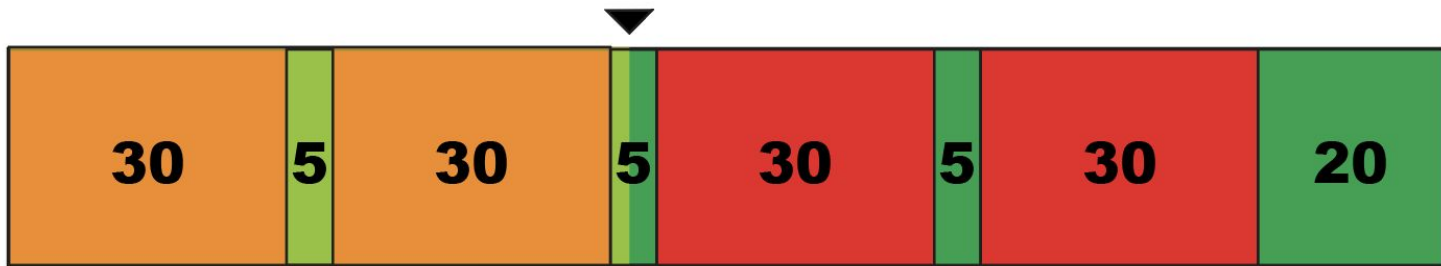




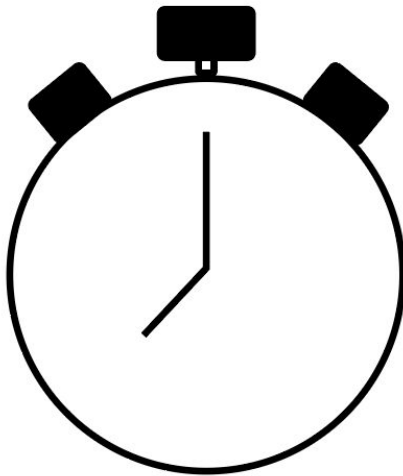
**FOCUS
TIME**






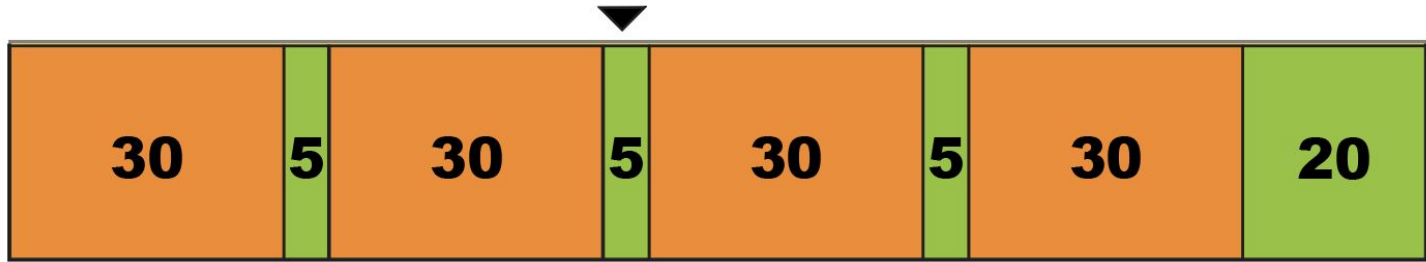
-  **Pause**
-  **Stop**
-  **Reset**



**BREAK
TIME**




-  **Pause**
-  **Stop**
-  **Reset**



COMPLETE

Please Enter:

Number of Cycles:



Break Time:

Long Break Time:
(one at the end of each cycle)

Focus Time:

Discussion on Limitations & Future Work

Current Limitations:

- Limited context awareness: Focus Bot may not fully understand complex software engineering tasks.
- Adaptability issues: Challenges in adapting to diverse work styles and environments.
- Dependency on user input: Reliance on user feedback for accuracy, which may be inconsistent.

Future Work:

- Enhanced AI algorithms: Improving context recognition to better understand software development tasks.
- Personalization: Developing more sophisticated personalization features to adapt to individual work styles.
- Integration with development tools: Seamless integration with popular software engineering tools for real-time assistance.

Support for Software Engineers:

- Distraction Reduction: Helps maintain a distraction-free environment, crucial for coding and problem-solving.
- Personalized Feedback: Offers tailored suggestions to improve workflow and efficiency.

Related Work

RescueTime:

- Overview: A time management application that tracks computer use to provide insights into productivity.
- Relevance: Similar to Focus Bot, RescueTime helps users understand their work habits and eliminate distractions.

Toggl Track:

- Overview: A simple time tracking tool that allows users to monitor how much time they spend on various tasks.
- Relevance: Offers data-driven insights, aiding software engineers in managing time more effectively, akin to Focus Bot's monitoring capabilities.

Processes & tools used

Before beginning to work on our project we had many discussions for **requirement elicitation**. This was useful so that we could have starting point on what should be included in our first prototype.

We used **prototyping** as our main process in order to develop this bot. Our team would meet and rework our prototype in order to address our problem statement. We often would think of criteria that our users would want/need. This was useful so that we were all on the same page on how we should build our bot.

In order to split the work evenly we used **scrum and kanban**. We had a shared Google Keep note of all the tasks we needed to complete for the next milestone and would share over slack our progress after class weekly.

Tools Used: Google Slides, Figma, Google Keep

[Link to Final Prototype](#)

Things We Learned

- Project Management: Understanding how to plan, organize, and manage a project, including time management and meeting deadlines.
- Teamwork Skills: Learning how to collaborate effectively with others and how to contribute positively.
- User Experience (UX) Design: Understanding how to design a user-friendly interface that is intuitive and effective for users. This includes learning about layout, color schemes, and interaction design.