CSE 170 Project

Alex Lessing

Needs observed

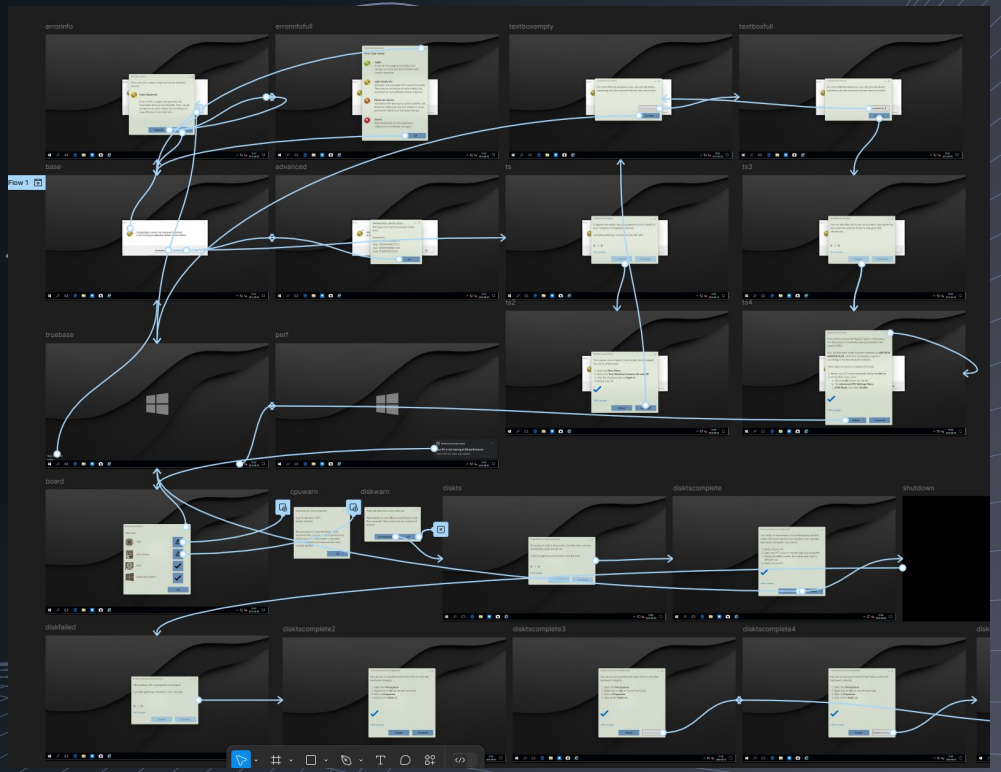
Users were observed struggling with computer issues via the Tech Support and Windows 10 forums on Reddit. From this, a need was inferred: The ability to troubleshoot intuitively, without feeling helpless and requiring outside assistance. Although the issues seen were understandably very diverse, a few key themes stood out repeatedly:

- Not understanding the practical meanings or causes of errors
- “Silent” issues, such as mysterious drops in performance
- Inability to judge the severity of an issue

Thus, a design was created with the aim of alleviating these issues.

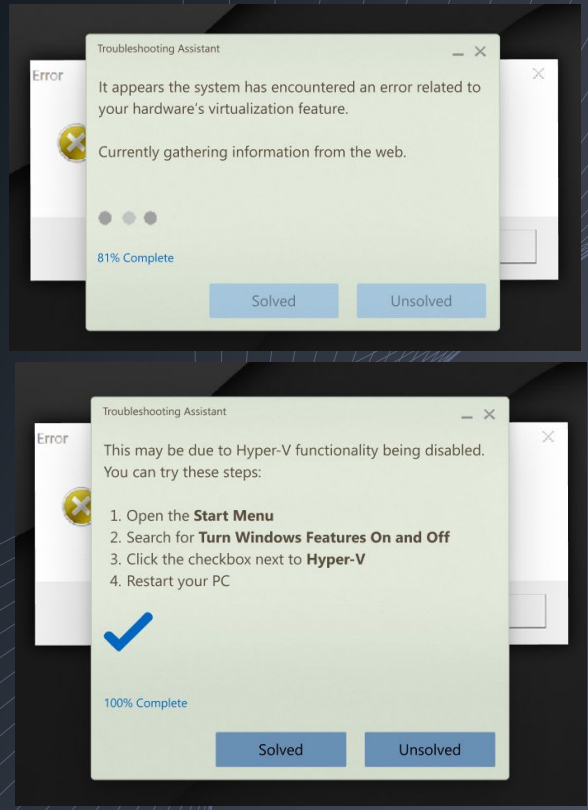
Prototype

Figma was used to prototype a troubleshooting software suite that handles the aforementioned types of issues in a more proactive way. The context is a simulated Windows desktop.

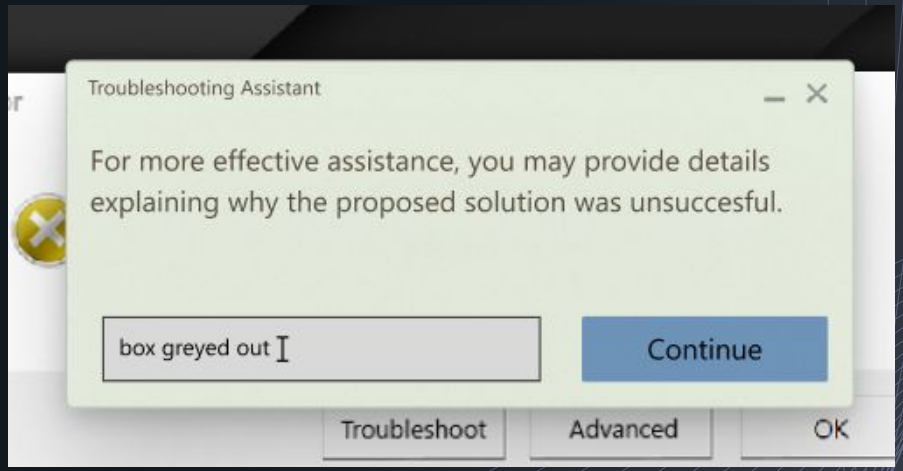


Need: Error solutions

Upon encountering errors, users can open an online, AI-assisted troubleshooting window. This automates the process of manually needing to search/post about the issue, leading to a much less frustrating experience. After a solution is found, the problem can be marked as solved, or unsolved – leading to iterative troubleshooting. Additionally, because this functionality is installed on the user's computer, its hardware specifications and configuration are known and considered during the troubleshooting process.

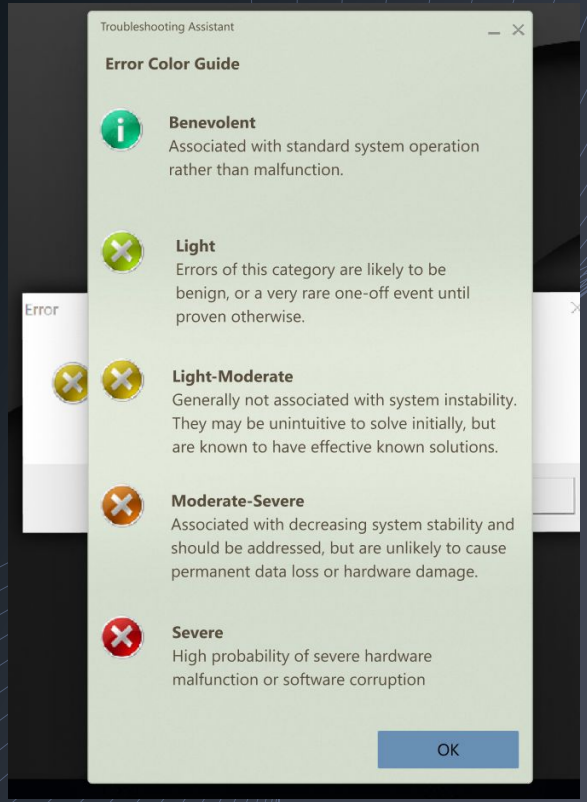


Need: Error solutions



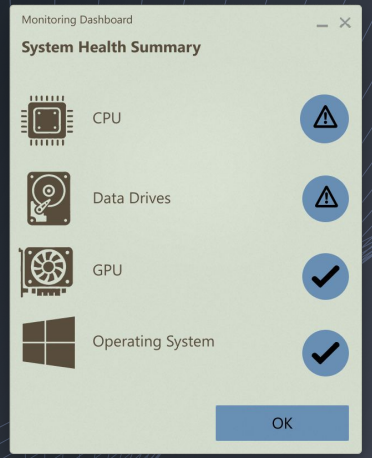
Need: Judging error severity

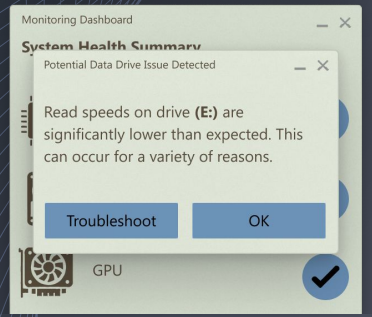
It was noted that multiple users mistook benevolent system messages as symptoms of malicious software. There was also anxiety relating to how problematic an error actually was. To address this, a color-coding system was created that heuristically evaluates how severe an issue is likely to be. This gives users a realistic sense of what they're dealing with, and sets expectations regarding the gravity of the solution.

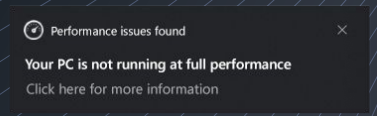


Need: Monitoring performance

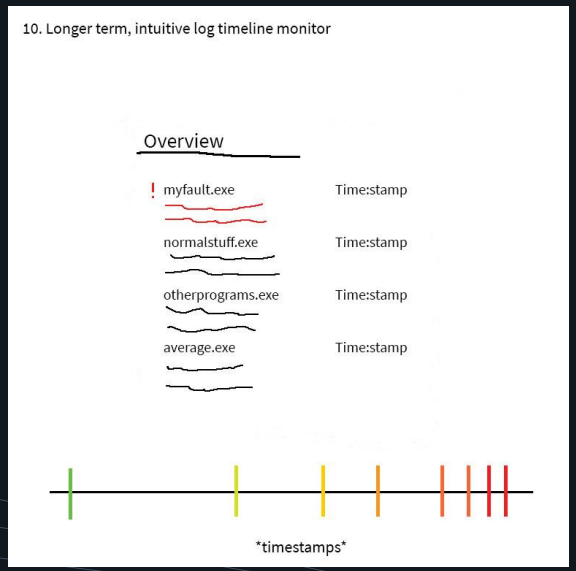
Sometimes, observed users noticed their computer running more slowly or crashing without explicit error messages. As this is typically due to anomalies in hardware or software operation, a simple dashboard is implemented that allows the user to check the status of their system's components, and work through potential issues using the previously mentioned AI troubleshooting assistant. Additionally, it is proactive; a notification informs the user if anything is running out of usual range and directs them to the dashboard.

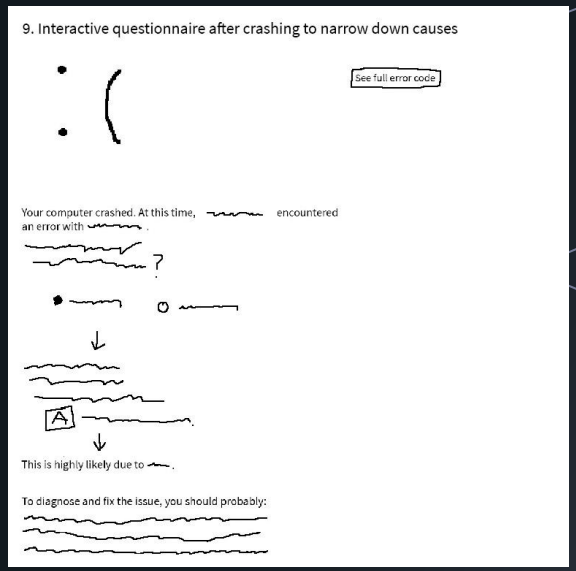






Alternatives considered

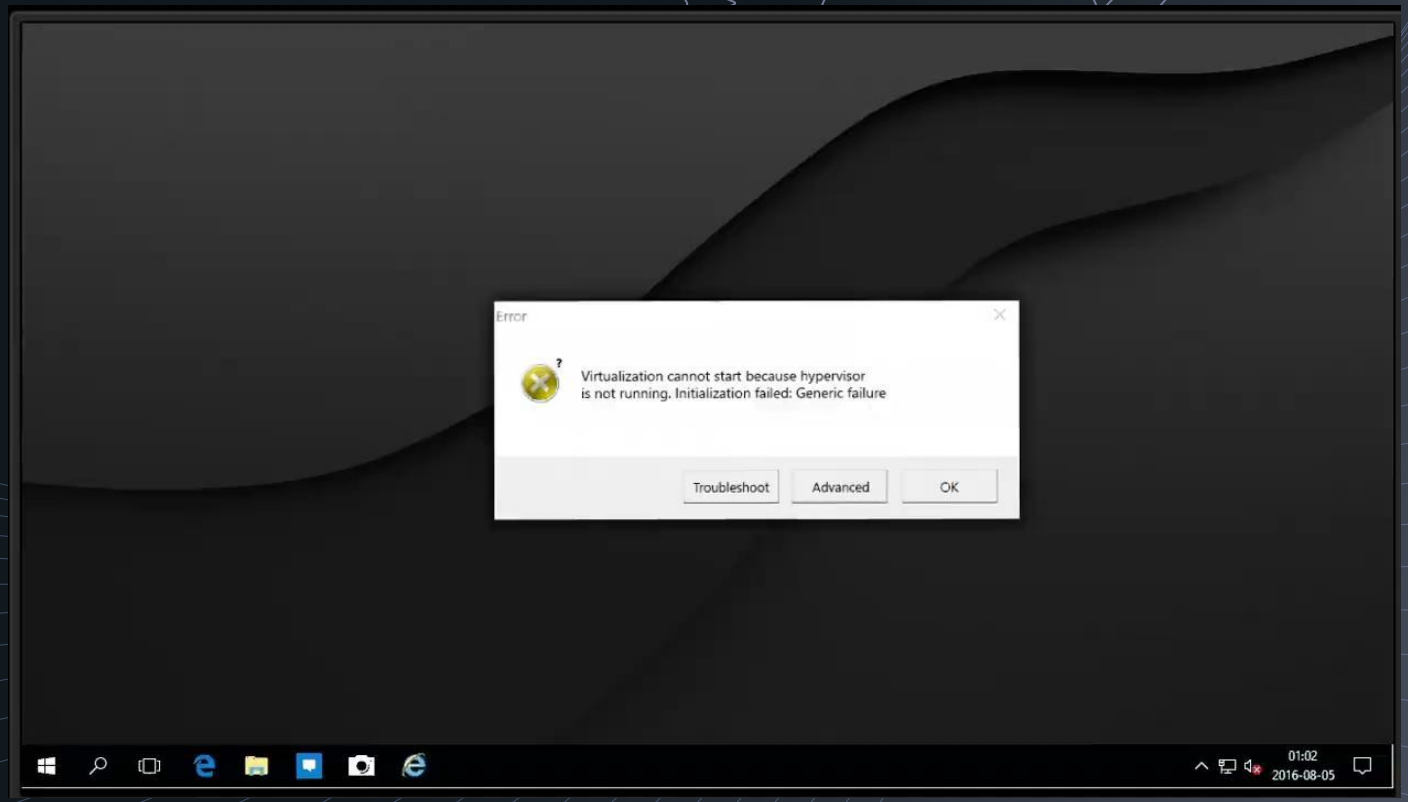




Alternatives considered included:

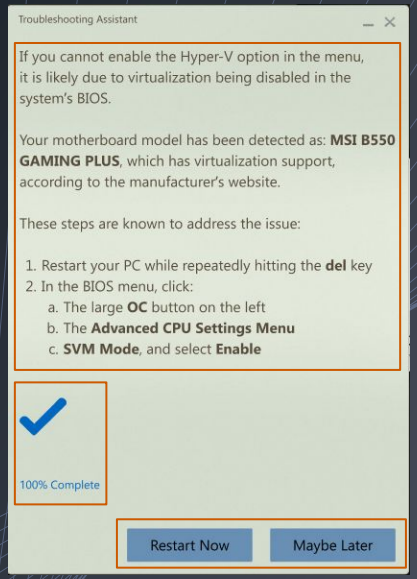
- Longterm error tracking timeline monitor (Unnecessary information and perhaps unintuitive to view)
- A **survey-based** AI troubleshooter (Users may not want to be bombarded with questions; their input was made optional)
- The monitoring system using a **large pop-up with immediate directions** (Possibly too overwhelming compared to a more subtle notification linking to a dashboard with thorough information)

User Flow Example

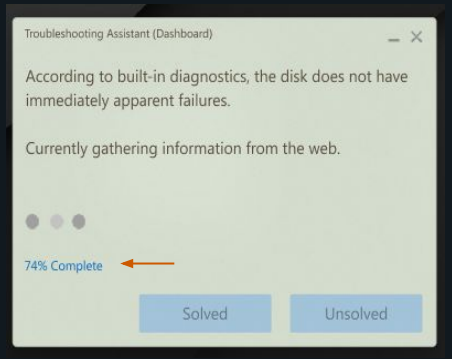


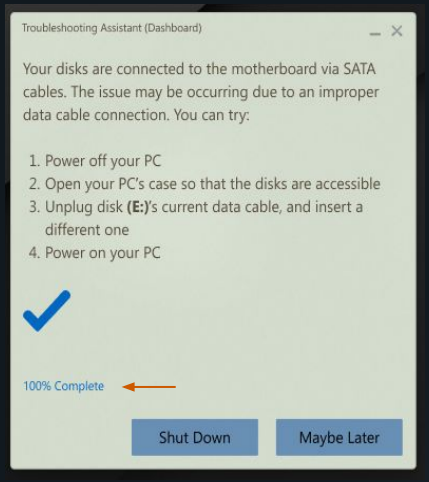
Decisions P1: Visuals

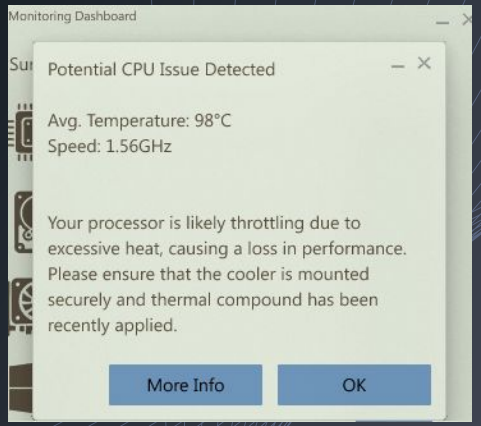
The static elements of the application use tan/brown tones, while the dynamic ones use blue. The buttons appear as blue rectangles. This creates **contrast** and **consistency**, making it clear where the user can interact. Additionally, the instructional text, buttons, and progress indicators stay within their own groups, adhering to **proximity**. Overall, the design is minimalist and intuitive.



Decisions P2: Information







Visibility of system status can be seen as the troubleshooter displays its progress, keeping the user informed in real time.

Also, the dashboard feature directly monitors the status of the user's computer.

User Testing Insights

During moderated usability testing, a few concerns were consistently highlighted:

- The dynamic dashboard status icons were red/green, introducing inconsistency and making their interactivity unclear (They are now blue and resemble the application's other buttons)
- The input text box being integrated into the "Unsolved" button felt awkward (It is now in its own dedicated window)
- The question mark icon near the initial error sign associated with the color-coding information was too difficult to press (The hitbox is expanded, and now covers the entire error sign as well)

Original text box

Thank you!