2024 Internship Report

The internship project, problem defined

During my 12-week internship at Microsoft, I worked on a project that extended the existing Get Help app in the Windows Operating System with a new functionality to analyze Wi-Fi connections and the surrounding environment to provide diagnoses and corresponding recommendations to improve the Wi-Fi quality.

Wi-Fi plays a vital role in modern computing, enabling users to connect to the internet, stream media, and collaborate online. However, Wi-Fi networks often face issues like congestion, low signal strength, outdated hardware, and ISP bottlenecks, which can degrade the user experience. The Get Help app, which contains the network diagnostics tool for the Windows OS, currently only diagnoses disconnection and unreliable Wi-Fi connections and is not able to offer any advice regarding slow or lossy Wi-Fi connections. 48% of Windows users who launched the Get Help app with an active WLAN interface were connected to Wi-Fi. That means the app was not able to help 48% of its WLAN users without my feature.

The Wi-Fi analyzer that I created scans the user's network and surrounding Wi-Fi networks to detect details like frequency band, channel, signal strength, and security protocol. It then analyzes the data and provides actionable recommendations to improve connectivity. For example, if a network is using a congested channel or an outdated Wi-Fi standard, the app suggests alternatives and upgrades. By integrating these capabilities, the Get Help app now offers users a more comprehensive solution for troubleshooting and optimizing their Wi-Fi connectivity. The app covers 8 common cases where the Wi-Fi connection can be improved and provides actionable recommendations without prolonging the diagnosing time. My team expect to integrate this feature into the next release of the app.

My Work & Role

I was the sole owner of this feature and I participated in every step of the production process. I started the design phase with a thorough competitor analysis, defined functional requirements, drafted a 17-page dev design spec, and had it peer reviewed. Moving into the implementation phase, I wrote code and regularly participated in code review every two to three days. I used unit tests, integration tests, and manual tests to validate my code. Since the app is owned by the Connection Awareness Team, which is my official team, but my feature intersects with the Wi-Fi team, I closely worked with both teams to understand the problem and ensured the successful completion of the project.

Challenges

- **Learning and Applying Networking Concepts**

Going into this internship, I did not have much knowledge about networking (I dropped COSC 60 Computer Networks, oops) and I certainly did not have industrial experience of networking. I

quickly ramped up by taking Coursera courses and asking questions on search engines and LLMs over the first week. I clarified confusions with my mentor and continued learning new things when I was coding. By the end of the internship, I can explain Wi-Fi concepts very well and play around with Windows WLAN APIs.

- **Drafting the Dev Design Spec**

This is the ultimate version of the design document of the final project of COSC 50: I was tasked to design something that no one knows how to do, alone. At the time that I drafted the spec, I knew nothing about what tools I could utilize, what APIs I could call, what parameters I needed, etc. I was afraid that I would design something that could not be implemented. To overcome this, I familiarized myself with the most common Windows WLAN APIs, such as querying WLAN interface and initiating an active Wi-Fi scan. After that, I kept track of two lists: what information I could gather and what information I need based on the functional requirements. I made sure that these lists matched each other at all times, so that I could get all the parameters I needed for the APIs and provide the most detailed and confident diagnoses.

- **Building on Top of an Existing App**

My task was to add a feature to an existing app, and that is a double-edged sword. I did not have to design all the components from scratch, but I had less freedom to do whatever I wanted. The most important thing at the beginning of my internship was to quickly understand the codebase and framework and know what to add, where to add, and how to add. I also needed to modify some existing data structure or methods. To follow industry standards and best coding practices, my new design had to align well with the current framework and cause minimum confusion or restructuring. That was what I had in mind when I wrote the design spec and the code.

Take Aways

- **The End-to-end Workflow of a Software Engineer**

One of the most valuable experiences during my internship was gaining insight into the complete workflow of a software engineer. From the initial planning and design phases through development, testing, and deployment, I experienced every step of the process. My manager pushed me to do a lot of tasks that only full-time employees do at Microsoft, such as writing specs and doing costing. This comprehensive involvement deepened my understanding of software development lifecycle practices, allowing me to appreciate the importance of thorough planning, collaborative coding, rigorous testing, and continuous iteration to deliver a high-quality product.

- **Hands-on Experience with Essential Microsoft Technologies**

Throughout my internship, I gained hands-on experience with essential Microsoft technologies like Azure, Visual Studio, and Microsoft 365. These tools not only enabled me to build and

deploy features efficiently but also provided me with a deeper understanding of Microsoft's technology stack. This experience has expanded my technical skill set and prepared me to work more effectively within the Microsoft ecosystem and similar enterprise environments.

- **Improving Workflow and Efficiency Using AI Powered Tools**

I improved my workflow and efficiency by incorporating tools like Copilot into my daily routine. Copilot played a crucial role in streamlining code development, allowing me to write and optimize code more efficiently. By leveraging this AI-powered tool, I was able to accelerate my coding process, reduce errors, and focus more on the strategic aspects of problem-solving and feature development. This has significantly enhanced my productivity and will continue to benefit my approach to software development in the future.

Favorite Part of the Project

My favorite part of the internship project was undoubtedly the coding. There's something incredibly satisfying about seeing lines of code transform into a functional product, and it was the part of the project where I felt most at home. What made it even more enjoyable was how AI tools, like Copilot, have streamlined the coding process. With these tools, I didn't have to write everything from scratch, which gave me more time to focus on higher-level problem-solving. It's like steering the ship rather than paddling—allowing me to guide the project's direction while letting the AI handle some of the heavy lifting.

However, using AI didn't mean the process was entirely hands-off. I frequently encountered errors in the code generated by Copilot that required careful attention to fix. It became a fun challenge to debug these issues, simplify the code with smarter algorithms, and integrate better coding practices to ensure everything worked seamlessly. This blend of AI-assisted coding and hands-on problem-solving made the process dynamic and kept me engaged throughout.

Another highlight was watching the code I wrote come to life. There's a unique satisfaction in creating something new and useful from scratch. Even when things didn't go as planned and the program crashed, I found joy in diving into the call stack, examining variables, and tracking down the source of the problem. Debugging, in a way, felt like solving a puzzle, and each success brought a sense of accomplishment.

Least Favorite Part

While there were many enjoyable aspects of my internship, the least favorite part for me was working with public APIs. It was often a frustrating process, especially when it came to finding the right documentation. Most of the documentation was written in C++, while my project was in C#, and bridging that gap wasn't always straightforward. The differences between the languages, combined with discrepancies in how certain functions were implemented, made integration more challenging than I anticipated.

This disconnect between the documentation and my C# code meant I had to spend a significant amount of time troubleshooting and experimenting to get things to work properly. It often felt like I was piecing together a puzzle with missing instructions, which slowed down my progress. While it was a valuable learning experience in terms of problem-solving and perseverance, it was undoubtedly the most tedious part of the project.

Impact On My Career Plans

I committed to the Computer Science major in my sophomore year, which was comparatively late. This internship, being the first and only one for me, had a tremendous and positive impact on my confidence and dedication to the Software Engineering position and the tech industry. It allowed me to strengthen my technical skills, deepen my understanding of large-scale software development, and collaborate with talented professionals in the industry.

I felt comfortable, competent, and rewarded doing my work. I looked forward to going to work and learning something new every day. I believed that what I did was meaningful and has impact. More specifically, I felt comfortable working at a big tech company and interacting with my manager, my mentor, and my team members. Although I still do not find systems and networks to be the most interesting thing in the world, I am very content with this internship and I would be glad to return to the same team or same company as a full-time employee.

Manager's assessment

To whom it may concern:

Selena recently completed an internship on the Mobile Connectivity team at Microsoft, and I was her manager. Selena did an excellent job ramping up on Windows networking and tools to produce a design for Wi-Fi troubleshooting that builds upon the existing Get-Help app. Selena held the design for peer review and incorporated their feedback. From there she produced cost estimates for the work and moved into an implementation phase.

Selena implemented the required code in C# within her estimates and was able to implement eight of the Wi-Fi scenarios. Her work culminated in two separate demos for the intern program and our broader team. Selena gave a clear presentation including the context of her project within the broader market, challenges faced and recommendations for future work. Selena's code was high quality, and she was able to smoothly incorporate code review feedback from the team. Selena self-tested as she made progress on the implementation, including work with Wi-Fi hardware.

Selena learns quickly and became proficient with Microsoft tools and technologies, including C# and Visual Studio over the course of the internship. Selena was a good communicator, presenter and collaborated well with other team members, including developing a working relationship with the Wi-Fi team. Her mentor was impressed with the level of independence she had over the course of the internship. We plan to productize Selena's project in a future release and expect to leverage parts of her design and implementation.

André Schoorl

Development Lead, Microsoft

aschoorl@microsoft.com