**CS 499 Module Six Journal**

Andrew Emilio DiStefano

SNHU

Professor Neil Kalinowski

June 14, 2025

**Part One:**

1. **What is the identification and description of each technology?**

The three technologies which I choose to discuss in this journal assignment are solid state drives, Arm processors, and the Android operating system.

Before Solid State Drives (SSD) became widely accessible, Hard Disk Drives (HDD) were used in most consumer-grade computers (Brant, 2025). Interaction with data stored on these memory devices was slower and more resource intensive. Then disks of HDDs also needed to physically spin while the system was reading from them, which meant that the machine needed to remain stationary in order to avoid risking costly damage to the system. SDDs were first popularized in the 1970s, but the price point for these devices was so much higher than that of HDDs that it was not feasible to use them in consumer-grade computers (Brant, 2025). The use of SDDs became much more prevalent in the late 2000s and onward, as the price point dropped significantly in relation to that of HDDs (Brant, 2025).

Arm processors are a game-changing type technology which enables complex computing while working faster and consuming significantly less power than processors commonly used on laptops and consumer-grade computers. Due to their resource efficiency, Arm processors are often used for mobile devices and embedded systems (Arm Editorial Team, 2020). Their competitive prices and complex capabilities despite low power consumption has encouraged developers to use them in situations where energy consumption is a major concern.

The Android operating system was developed in the early 2000’s (GeeksForGeeks, 2024) and uses a Linux kernel to produce an open-source mobile operating system which has become the most widely used of all mobile OSs worldwide (Sherif, 2025). Android has done a great deal to make smartphones more accessible to the general public.

1. **What are the likely impacts on computer science or your career?**

Hard Disk Drives (HDD), which were used in most consumer-grade computers before Solid State Drives (SSD) became widely accessible, have several significant drawbacks for application in the robotics industry. The physical disk(s) within an HDD need(s) to rotate without being disturbed by sudden movements which can interfere with operation being performed on stored data. Without SSDs, many of our modern robotics innovations would have been virtually impossible.

Arm processor technology is also incredibly important to the robotics industry. The quick response times and low energy consumption afforded through the use of the RISC (Reduced Instruction Set Computing) architecture used in Arm processors (Johnson, 2024) has made their use almost mandatory in many subcategories of the robotics industry. The Raspberry Pi, which is notorious for its use in robotics projects (particularly popular for prototyping and hobbyist-level robotics projects), uses Arm processors for the aforementioned reasons.

While the Android operating system is not directly involved in most robotics projects, there are certainly ways in which thai prevalent OS has influenced the robotics industry. Smartphones are embedded systems which prioritize low energy consumption and fast loading and response times, which means that their architecture shares several key similarities with that of autonomous mobile robots (AMR). I have even seen developers post videos online of simple autonomous mobile robots which use smartphones with the Android operating system as their control systems (Ingmar Stapel, 2020). It is reasonable to deduce that developers of robotics and embedded systems have almost certainly drawn inspiration from the system designs of devices meant to run the Android operating system.

1. **How might the three technologies impact humans, communities, or the world?**

Hard disk drives have become the standard in computer production, and I have personally not seen a machine with a hard disk drive since the early 2010s. This significant advancement in data storage hardware has made computing more feasible in more industries, including for applications which require machines to be in constant motion. This has of course enhanced a wide variety of industries on which humanity relies to fulfill basic human needs.

Arm processors have also certainly been a game-changer in a wide variety of industries. Without Arm processors, mobile devices would not function as effectively and efficiently as they do today. This would limit the ability for individuals around the world to maintain a connection to the internet.

As the most widely used mobile OS worldwide, the Android operating system has lowered the financial barriers involved in owning a smartphone to the point where many more people on earth have them than do not (Sherif, 2025). I have personally seen smartphones which run the Android operating system for sale for the equivalent of less than thirty dollars in developing nations. As a low-cost option with great development support due to widespread adoption, this open-source smartphone operating system has had a significant positive impact on people around the world.

1. **Which course outcomes have you achieved so far, and which ones remain?**

At this point I have achieved all of the provided course outcomes, and I will continue to address these outcomes as I add the finishing touches to my two artifacts over the next two weeks. I have consistently updated my GitHub repository for each project after every meaningful enhancement. I have also used open-source libraries created by other developers in order to accomplish complex tasks like JSON interaction. These practices show my ability to employ strategies for building collaborative environments that enable diverse audiences to support organizational decision-making in the field of computer science. Throughout the development process, I have kept a list of dependencies which is located in the folder which holds these dependencies. This documentation contains citations and attributions to the authors of these open-source tools. My performance of these tasks demonstrates my ability to design, develop, and deliver professional-quality oral, written, and visual communications that are coherent, technically sound, and appropriately adapted to specific audiences and contexts. My ability to create and use complex indexing methods such as the inverted index in the MemoryBot project shows my capacity for designing and evaluating computing solutions that solve a given problem using algorithmic principles and computer science practices and standards appropriate to its solution while managing the trade-offs involved in design choices. This is the same data structure used by popular search engines for its ability to efficiently search input strings for specific word combinations. The implementation of data structures used in real-world applications demonstrates my ability to use well-founded and innovative techniques, skills, and tools in computing practices for the purpose of implementing computer solutions that deliver value and accomplish industry-specific goals. My limiting of user input to as few variations as possible and my use of type conversion functions to assure that variable types are configured correctly show my capacity for developing a security mindset that anticipates adversarial exploits in software architecture and designs to expose potential vulnerabilities, mitigate design flaws, and ensure privacy and enhanced security of data and resources.

**Part Two:**

* **Software design and engineering**

Over the past few days, I have finished the implementation of the HumidityMachine class for the humidity control state machine, and everything is up and running. I was able to effectively connect another LCD screen to my circuit despite the limited available GPIO pins by sharing most of the pins with the other LCD screen on the circuit. The system now functions as intended with sensing and control for air temperature and air humidity. I have used a combination of asynchronous functions and Python ‘multithreading’ in order to optimize the efficiency of my program despite the drawbacks associated with writing a program like this in Python (especially the lack of efficiency which can be achieved without optimizing memory management and data allocation). Seeing both LCD screens showing the appropriate information felt great, and I am now quite confident of my ability to design and implement Raspberry Pi circuits with software which interprets sensor data and performs the appropriate operations based on said input.

* **Algorithms and data structures**

I am proud to say that my Inverted Index implemented in the MemoryBot system is now up and fully functional. This was quite honestly one of the most satisfying *eureka* moments of my entire software engineering journey. I had actually been pondering what would be the best data structure and search algorithm for this program ever since I had developed it. I noticed that the program would always output the answer for the first question in the data store. This makes for an effective conversation simulator when there are only a few different question and answer pairs, but responses become less relevant as the number of question-answer (key-value) pairs becomes larger.

Thanks to my research into how tech industry leaders achieve similar goals, I was able to identify the inverted index as the most appropriate data structure for this program. The map which stores every word in every string in the database along with how many times each of those tokens (words) appear in each question is an incredibly efficient way to achieve these particular system goals involving comparison of user input strings with saved strings word-for-word. One of the issues which I had been having with the search algorithm was that I had failed to understand that what makes this data structure so powerful is the use of a linear merge algorithm for querying the data structure. Correctly implementing this algorithm made everything work as intended.

I have been trying this program with different JSON files loaded with question-answer pairs created by free AI tools. For example, asking an AI algorithm to produce a JSON file loaded with question-answer (key-value) pairs representing an interview with an ancient Roman soldier who hated Juluis Cesar actually does a pretty good job of simulating a conversation with such an individual! I included this JSON file in the project uploaded to my eportfolio and you are more than welcome to try it. The precision with which the program is able to produce the best answer based on input is somewhat chilling, and I can definitely see how a JSON file with 10,000 question-answer pairs rather than the less than 100 which are in the current file. The more question-answer pairs are added, the more closely the conversation begins to approximate a real conversation. I am really proud of this, and I hope that you enjoy reviewing this in my final project!

* **Databases**

I am incredibly close to successfully implementing the database functionality of the PlantSitter application! I have developed a respectable understanding for how the ReductStore database works, how to implement it locally on a single-board computer, and how these techniques are used in the robotic industry for the storage and interpretation of visual sensor data. As soon as I fix an issue relating to the use of the function which saves information to the database as an asynchronous function, the program should be able to successfully interact with the databases configured for the storage of timestamped temperature and humidity data.

The implementation of database functionality for the MemoryBot program (or rather for the JSON functionality which could easily be set up to work with virtually any database) is certainly working in terms of reading data. The InvertedIndex class’ writeToDatabase function should be able to successfully write new entries to the JSON file, but I have yet to test this function by calling it in my conversation loop. I believe that I will be able to implement functions for adding and removing entries from the inverted index relatively easily.

### **Status Checkpoints for All Categories**

| **Checkpoint** | **Software Design and Engineering** | **Algorithms and Data Structures** | **Databases** |
| --- | --- | --- | --- |
| **Name of Artifact Used** | PlantSitter | MemoryBot | PlantSitter |
| **Status of Initial Enhancement** | Initial enhancements have all been implemented. | Initial enhancements have all been implemented. | Initial enhancements have all been implemented. |
| **Submission Status** | 95% finished | 90% finished | 89% finished |
| **Status of Final Enhancement** | Everything is working, but I still need to go through the added class and proofread my documentation. | The inverted index data structure and linear merge algorithms are up and running, but I still need to find the best way and place to call the functions from adding or removing entries. | Everything is in place and very close to working, but I still need to correct the logic of my use of asynchronous functions for interacting with the ReductStore database. |
| **Uploaded to ePortfolio** | Yes | Yes | Yes |
| **Status of Finalized ePortfolio** | 100% up to date | 100% up to date | 100% up to date |

**Sources:**

Arm Editorial Team.. (2020, September 14). *Next Generation*

*Robots Powered by Arm*. Arm.

<https://newsroom.arm.com/blog/arm-powered-robotics>

Brant. T. (2025, March 31). *SSD vs. HDD: What's the Difference?*

PC Mag. [https://www.pcmag.com/comparisons/ssd-vs-hdd-](https://www.pcmag.com/comparisons/ssd-vs-hdd-whats-the-difference)

[whats-the-difference](https://www.pcmag.com/comparisons/ssd-vs-hdd-whats-the-difference)

GeeksForGeeks. (2025, February 24). *History of Android*.

<https://www.geeksforgeeks.org/history-of-android/>

Ingmar Stapel. (2020, November 18). *OpenBot - Turning*

*Smartphones into Robots* [Video]. YouTube.

<https://www.youtube.com/watch?v=gHHTZAcu9bU>

Johnson, C. (2024). *ARM Processing and Its Impact on MSPs*.

Axcient. [https://axcient.com/blog/arm-processing-and-](https://axcient.com/blog/arm-processing-and-its-impact-on-msps)

[its-impact-on-msps](https://axcient.com/blog/arm-processing-and-its-impact-on-msps)

Sherif, A. (2025, March 11). *Market share of mobile operating*

*systems worldwide from 2009 to 2025, by quarter*.

Statista.

[https://www.statista.com/statistics/272698/global-](https://www.statista.com/statistics/272698/global-market-share-held-by-mobile-operating-systems-since-2009/)

[Market-share-held-by-mobile-operating-systems-](https://www.statista.com/statistics/272698/global-market-share-held-by-mobile-operating-systems-since-2009/)

[since-2009/](https://www.statista.com/statistics/272698/global-market-share-held-by-mobile-operating-systems-since-2009/)

Sherif, A. (2025, March 24). *Smartphone penetration*

*worldwide as share of global population 2016-*

*2024*. Statista. [https://www.statista.com/statistics](https://www.statista.com/statistics/203734/global-smartphone-penetration-per-capita-since-2005)

[/203734/global-smartphone-penetration-per-](https://www.statista.com/statistics/203734/global-smartphone-penetration-per-capita-since-2005)

[capita-since-2005](https://www.statista.com/statistics/203734/global-smartphone-penetration-per-capita-since-2005)