CS-499: Computer Science Capstone

Professional Journals Compilation

Ariana Mikhak

August 2025

——

CS-499: Computer Science Capstone

2-1 Journal: What Makes a Productive Code Review?

Ariana Mikhak

July 11, 2025

Code review is the process of examining code to find bugs, improve structure, and check for security or performance issues before merging it into the main project. It's a key part of professional software development because it helps catch problems early, improves overall code quality, and encourages collaboration between developers. Some best practices I found useful from the readings include keeping the reviews focused and small, using automated tools where possible, and offering feedback that's constructive and specific. The OWASP guide also emphasized the importance of reviewing for security issues like input validation and access control. Code reviews should happen once a piece of functionality is complete but before it's merged into the main branch, so fixes don't cause bigger problems down the line.

For my screencast, I'll be using the screencasting and editing feature of Descript to record and edit my walk through of the EventTrackerApp created in Android Studio. To stay on track during the review, I prepared a script and screenshots to cover each section of my presentation. I'll focus on the three enhancement categories—software engineering, algorithms and data structures, and databases—explaining how the current code works and what I plan to improve. This approach

helps me stay organized and makes sure I hit all the important review points from the rubric and checklist.

———

CS-499: Computer Science Capstone

3-1 Journal: Marketing With ePortfolios and Artifact Update

Ariana Mikhak

July 18, 2025

Part One – Reflection Questions

1. How might you use an ePortfolio for the benefit of self-promotion?

An ePortfolio allows me to present my skills, projects, and growth as a developer in a cohesive and public-facing way. I'm using GitHub to host my portfolio, where I'll showcase my capstone project, *EventTrackerApp*, along with detailed descriptions of the enhancements I've implemented. Each enhancement is clearly labeled by category (Software Design, Algorithms and Data Structures, Databases), making it easy for potential employers to see how I applied my technical knowledge. The ePortfolio serves as a concrete demonstration of my ability to design, build, and iterate on real applications.

2. How might you mitigate risks while maximizing the marketing potential of the ePortfolio?

To maximize the marketing impact of the portfolio while minimizing risk, I'm careful to avoid posting anything proprietary, insecure, or poorly documented. I've removed hardcoded credentials and am reviewing my commit history for anything that might appear unpolished or overly personal. I also plan to add a professional README file with clear instructions, a project

overview, and visual examples. To keep the tone professional and accessible, I'll include summaries for each enhancement so that even non-developers can understand the project's value.

3. Describe possible downsides or risks—for instance, the risks of posting intellectual property online for public consumption.

One risk of a public portfolio is having my work copied or reused without credit. Another is misrepresenting my skill level if I share incomplete, outdated, or poorly explained code. There's also a risk that the portfolio could be overlooked if it isn't well-organized or visually appealing. To reduce those risks, I'm being intentional about what I include, using clear structure, brief explanations, and eventually screenshots. I'm also following open-source licensing conventions and ensuring the repository presents a complete, professional impression.

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

So far, I've completed the Software Design and Engineering enhancement, which involved refactoring the app to use MVVM architecture, implementing an AI-powered title suggestion feature, and updating the UI to follow Material Design 3. These changes improved the app's structure, usability, and maintainability. I've also started work on the Algorithms and Data Structures enhancement by building an event conflict detection system using PriorityQueueand HashMap. The final enhancement—Databases—will involve refining SQLite queries and data validation to improve backend reliability.

Checkpoint	Software Design and Engineering	Algorithms and Data Structures	Databases

Name of Artifact Used	EventTrackerApp (Android mobile app for creating, editing, and managing events)	Same	Same
Status of Initial Enhancement	Completed (MVVM refactor, AI suggestions, Material Design)	Completed (PriorityQueue, HashMap, Conflict Detection)	Not started
Submission Status	Submitted in Milestone Two	Upcoming	Upcoming
Status of Final Enhancement	Final updates planned (README/ screenshots polish)	Final tuning planned	In progress
Uploaded to ePortfolio	Not yet	Not yet	Not yet
Status of Finalized ePortfolio	Planned for final module	Planned for final module	Planned for final module

———

CS-499: Computer Science Capstone

4-1 Journal: Career Choice and Artifact Update

Ariana Mikhak

July 24, 2025

Part One – Reflection

This week I completed my Algorithms and Data Structures enhancement for EventTrackerApp. The work focused on improving efficiency and responsiveness by replacing list-based event management with a PriorityQueue for scheduling and a HashMap for fast lookup. I also added conflict detection methods to ensure that overlapping events are identified before being added or updated.

These changes not only improved the app's runtime performance but also made me think more carefully about scalability and user experience. I had to ensure that the new logic worked seamlessly with the existing LiveData structure and database access layer.

Finishing this enhancement gave me a greater appreciation for the role of efficient algorithms and data structures in mobile development. It was also a satisfying step toward preparing for the deeper technical work I'll be taking on as I begin my MCS at UIUC this fall.

Checkpoint	Software Design and Engineering	Algorithms and Data Structures	Databases
Name of Artifact Used	EventTrackerApp (Android mobile app for creating, editing, and managing events)	Same	Same
Status of Initial Enhancement	Completed (MVVM refactor, AI suggestions, Material Design)	Completed (PriorityQueue, HashMap, Conflict Detection)	Not started
Submission Status	Submitted in Milestone Two	Submitted in Milestone Three	Upcoming
Status of Final Enhancement	Final updates planned (README/ screenshots polish)	Final tuning planned	In progress
Uploaded to ePortfolio	Not yet	Not yet	Not yet
Status of Finalized ePortfolio	Planned for final module	Planned for final module	Planned for final module

———

CS-499: Computer Science Capstone

5-1 Journal: Computer Science Trends and Artifact Update

Ariana Mikhak

August 1, 2025

Part One – Reflection on Trends

Trend 1: Cloud-Integrated Databases

Cloud-integrated databases have become a cornerstone of modern applications. Their significance lies in providing seamless data synchronization, scalability, and high availability for applications that need to serve users across multiple devices and platforms. By leveraging cloud-hosted databases or offline-first patterns with cloud sync, developers can create robust apps that feel fast and reliable even in variable connectivity environments.

This trend is changing computer science by shifting data architecture toward distributed and fault-tolerant designs. It enables developers to focus on business logic while cloud infrastructure handles performance, replication, and scaling. For consumers and workers, it means that data is always accessible and up to date, whether they are on a mobile device, desktop, or browser.

This trend aligns closely with my career goals in full-stack and mobile development. Learning to design apps like EventTrackerApp with the potential for cloud integration prepares me for real-world applications where offline caching and cloud sync are standard.

Trend 2: AI-Enhanced User Experiences in Mobile Apps

AI-driven features in mobile apps, like intelligent event title suggestions in EventTrackerApp, are becoming a major differentiator in user experience. This trend's significance is in automating repetitive tasks, providing personalized recommendations, and improving decision-making within apps.

AI integration is reshaping computer science by making machine learning models a routine part of application workflows. For users, this translates to time savings, smarter recommendations,

and applications that feel more responsive to their needs. For workers, it streamlines productivity tools and opens opportunities for AI-assisted decision support.

This trend fits my career aspirations by bridging software engineering with AI integration, allowing me to create apps that are not just functional but intelligent and user-friendly.

Course Outcomes Achieved So Far

- Designed and evaluated computing solutions using algorithmic principles (PriorityQueue, HashMap, conflict detection).
- Applied innovative techniques and tools in computing practices (MVVM architecture, AI event title suggestions).
- Enhanced database reliability and performance (normalized schema, optimized queries, CSV export).

With my database enhancement completed this week, I now move into the final review and polish phase to prepare EventTrackerApp for my ePortfolio.

Checkpoint	Software Design and Engineering	Algorithms and Data Structures	Databases
Name of Artifact Used	EventTrackerApp (Android mobile app for creating, editing, and managing events)	Same	Same
Status of Initial Enhancement	Completed (MVVM refactor, AI suggestions, Material Design)	Completed (PriorityQueue, HashMap, Conflict Detection)	Completed (Normalization, optimized queries, CSV export)
Submission Status	Submitted in Milestone Two	Submitted in Milestone Three	Submitted in Milestone Four

Status of Final Enhancement	Final updates planned (README/ screenshots polish)	Final tuning planned	Final review and polish for ePortfolio
Uploaded to ePortfolio	Not yet	Not yet	Not yet
Status of Finalized ePortfolio	Planned for final module	Planned for final module	Planned for final module

———

CS-499: Computer Science Capstone

6-1 Journal: Emerging Technology and Artifact Update

Ariana Mikhak

August 7, 2025

Part One – Reflection on Trends

Trend 1: Quantum Computing and Emerging Architectures

As we move beyond the limits of traditional silicon chips, emerging computing architectures such as quantum computing and neuromorphic technology are transforming the landscape of computer science. Quantum computing leverages qubits to process information in multiple states simultaneously, enabling breakthroughs in areas like cryptography, complex simulations, and optimization problems that are impractical for classical computers . Neuromorphic technology, meanwhile, mimics the structure of the human brain to perform low-energy, highly parallel computations that could revolutionize AI processing.

These trends matter because they signal the next paradigm shift in computing. Applications that once required vast server clusters may eventually run on far more efficient systems, and problems previously considered unsolvable could become tractable. For a future full-stack or mobile developer like me, understanding these trends highlights how current coding practices

may need to adapt to take advantage of these new processing models and how software design must anticipate rapid hardware evolution.

Trend 2: Datafication and Organizational Intelligence

Another defining trend is datafication, the transformation of nearly every human and organizational process into data-driven decision-making. Organizations now rely on interconnected devices, cloud infrastructure, and analytics pipelines to convert daily operations into “data objects” that guide strategy, improve efficiency, and support AI-driven insights .

For computer scientists, this trend underscores the importance of designing applications with data privacy, structure, and lifecycle management in mind. My work on EventTrackerApp aligns with this trend because the database enhancements I implemented—normalization, relational integrity, and CSV export—demonstrate my ability to create data that is organized, portable, and actionable. As data continues to drive decision-making across industries, developers who understand both the technical and ethical implications of datafication will have a competitive edge.

Course Reflection and Progress

Module Six has provided a brief opportunity to reflect and consolidate my work. Over the past five weeks, I:

- Completed all three artifact enhancements (Software Design & Engineering, Algorithms & Data Structures, and Databases).
- Strengthened MVVM architecture, efficient data structures, and a normalized relational database.

- Integrated AI-powered event title suggestions and CSV data export, showing awareness of trends like intelligent automation and data portability.

At this stage, all milestone enhancements are submitted, and my focus shifts to final polish, README documentation, and preparing my GitHub-hosted ePortfolio for Module Seven.

Checkpoint	Software Design and Engineering	Algorithms and Data Structures	Databases
Name of Artifact Used	EventTrackerApp (Android mobile app for creating, editing, and managing events)	Same	Same
Status of Initial Enhancement	Completed (MVVM refactor, AI suggestions, Material Design)	Completed (PriorityQueue, HashMap, Conflict Detection)	Completed (Normalization, optimized queries, CSV export)
Submission Status	Submitted in Milestone Two	Submitted in Milestone Three	Submitted in Milestone Four
Status of Final Enhancement	Preparing for ePortfolio polish	Preparing for ePortfolio polish	Preparing for ePortfolio polish
Uploaded to ePortfolio	Not yet	Not yet	Not yet
Status of Finalized ePortfolio	Planned for Module Seven	Planned for Module Seven	Planned for Module Seven