CS-499-11684-M01

Professor Fitzroy Nembhard

Fernanda Coghlan

05 May 2025

# CS 499 Module One Assignment Template

Complete this template by replacing the bracketed text with the relevant information.

1. **Self-Introduction:** Address all of the following questions to introduce yourself.
   1. How long have you been in the Computer Science program?

I have been enrolled in the Computer Science program at Southern New Hampshire University (SNHU) **since January 2023**. This is my second undergraduate degree. My first degree was in Mechanical Engineering with a major in Industrial Production Engineering, completed in Brazil. The Computer Science program at SNHU is my first undergraduate degree in the United States.

* 1. What have you learned while in the program? List three of the most important concepts or skills you have learned.

Throughout the program, I’ve learned not only how to apply technical skills but also why they work the way they do. Unlike bootcamps, this program has helped me understand the theory and logic behind programming and system design. Three of the most important concepts or skills I’ve gained are:

* A strong foundational understanding of **databases and data management**.
* Awareness of **security best practices** and the importance of maintaining integrity in code.
* The role of **professional ethics** and responsible computing in the field of computer science.
  1. Discuss the specific skills you aim to demonstrate through your enhancements to reach each of the course outcomes.

For this capstone project, I plan to improve a mobile application originally developed for **CS 360: Mobile Architecture and Programming**. My enhancement plan includes:

* **Software Design and Engineering:** I will restructure the application to implement an MVVM architecture, separating UI, logic, and data layers.
* **Algorithms and Data Structures:** I will add a feature to calculate weekly weight trends using basic array processing and loops.
* **Databases:** I will integrate a Room database to replace in-memory data storage, demonstrating data persistence and schema design.

These enhancements will demonstrate my skills in building clean architectures, implementing algorithmic logic, and working with modern database technologies — all while aligning with the five course outcomes.

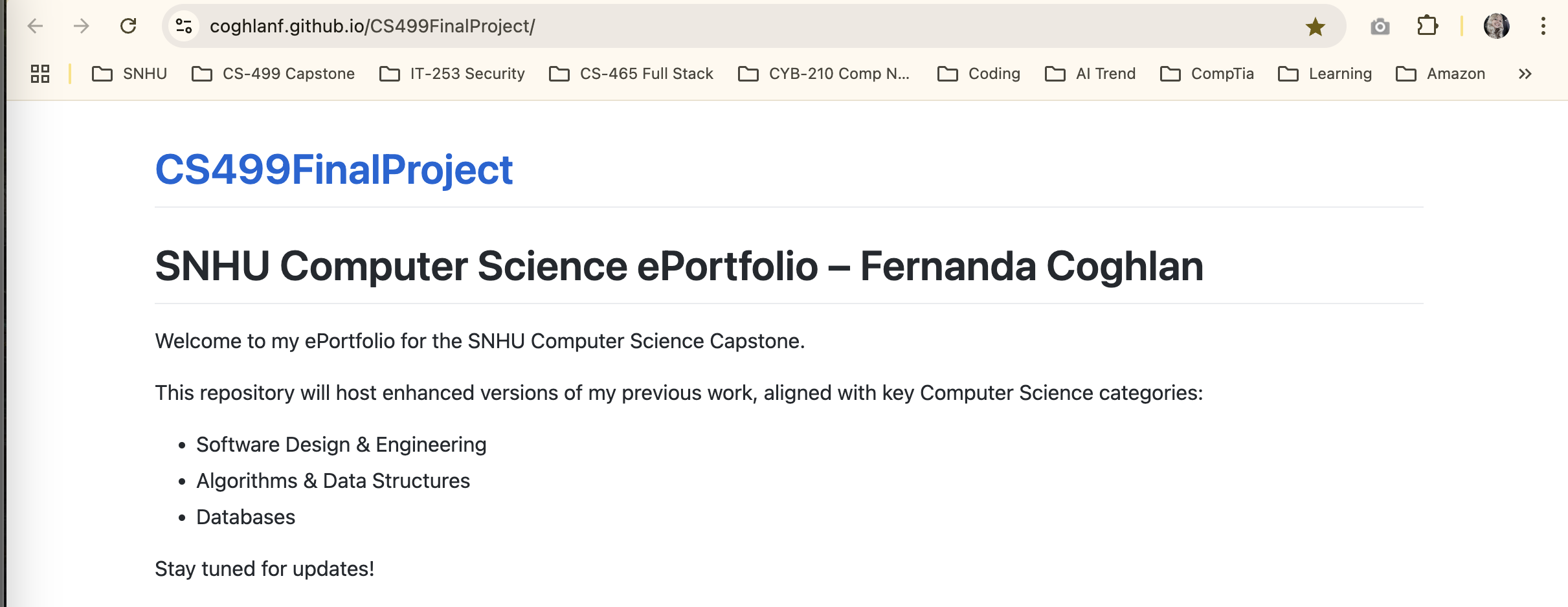
* 1. How do the specific skills you will demonstrate align with your career plans related to your degree?

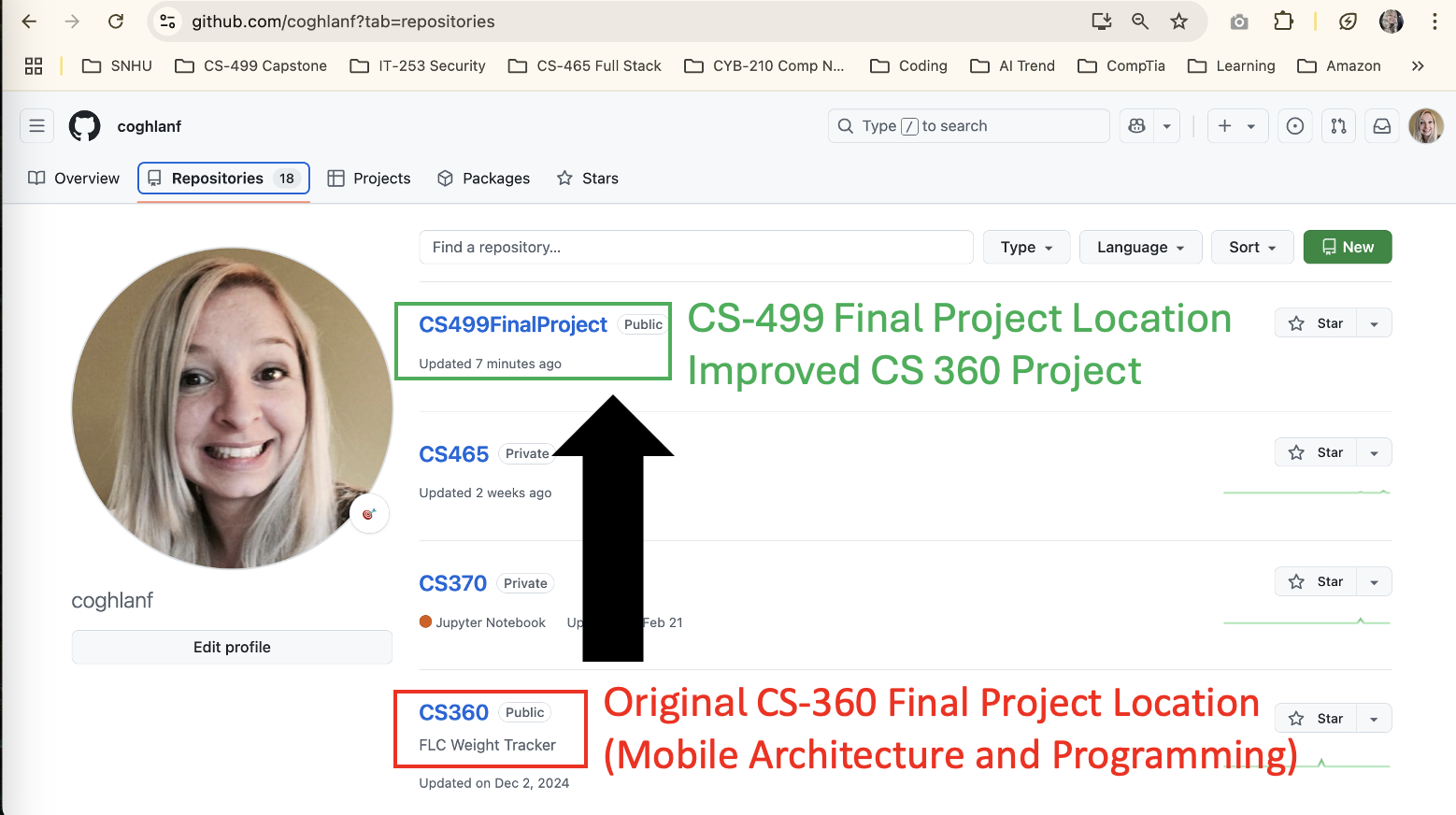
Currently, I work as a Mechanical Engineer in the aerospace industry. My career goal is to transition into the computer science field, ideally through an internal move or a new opportunity in tech. These enhancements allow me to showcase practical, cross-functional skills that can support that transition. By working on a complete mobile app, I’m demonstrating both technical versatility and real-world project experience — key qualifications for junior-level roles in software development or data-focused environments.

* 1. How does this contribute to the specialization you are targeting for your career?

My long-term goal is to work in data science. While this capstone doesn’t focus exclusively on data science tools, it gives me a foundation in the skills that underpin data-driven applications: clean software design, structured data storage, and basic trend analysis. These enhancements will show that I’m not only building technical skills but also preparing for a specialization that depends on thoughtful system architecture and data literacy. At 49, successfully pivoting into this field — even if not in a pure data science role at first — would be a rewarding milestone in my professional journey.

1. **ePortfolio Set Up:**
   1. Submit a **screen capture** of your ePortfolio GitHub Pages home page that clearly shows your URL.
      1. You already have a repository in GitHub where you uploaded projects in previous courses. Your ePortfolio will reside in GitHub but can link to work at other sites, such as Bitbucket.
   2. Use the GitHub Pages link in the Resource section for directions on:
      1. How to create your GitHub website and publish code to GitHub Pages
      2. Issues, such as adding links to other sites
   3. Paste a screenshot of your GitHub Pages home page with your URL clearly showing in the space below.

****



1. **Enhancement Plan:** 
   1. **Category One:** Software Engineering and Design
      1. **Select an** **artifact** that is **aligned with** **the** software engineering and design **category** and explain its origin. Submit a file containing the code for the artifact you choose with your enhancement plan.

**Artifact:**

The original artifact is a mobile application named **FLC Weight Tracker**, developed for the course **CS 360: Mobile Architecture and Programming** at SNHU. It was built using Android Studio with Java and followed a basic activity-to-activity architecture. The application allowed users to input and view weight entries, but it lacked a clean architectural structure and long-term data persistence.

The enhanced version will be developed in the **CS499FinalProject** repository and hosted on GitHub Pages.

Note: Your artifact may be work from the following courses:

* IT 145: Foundation in Application Development
* CS 250: Software Development Lifecycle
* CS 260: Data Structures and Algorithms
* IT 315: Object Oriented Analysis and Design
* CS 320: Software Testing, Automation, and Quality Assurance
* CS 330: Computational Graphics and Visualization
* CS 340: Advanced Programming Concepts
* CS 350: Emerging Systems Architectures and Technologies
* **CS 360: Mobile Architecture and Programming**
* IT 365: Operating Environments
* IT 380: Cybersecurity and Information Assurance
* CS 405: Secure Coding
* CS 410: Reverse Software engineering
* IT 340: Network and Telecommunication Management
* IT 380: Cybersecurity and Information Assurance
  + 1. **Describe** a practical, well-illustrated **plan** for enhancement in alignment with the category, including a pseudocode or flowchart that illustrates the planned enhancement.

**Enhancement Goal:**

Restructure the app using the **MVVM (Model-View-ViewModel)** architecture to separate UI, logic, and data responsibilities for cleaner code, better testing, and future scalability.

**Plan Summary:**

* **Add a ViewModel** class to encapsulate UI-related data logic.
* **Create a Repository** layer to manage data sources (starting with in-memory or Room DB).
* **Refactor the Activity** to observe LiveData from the ViewModel rather than handling logic directly.
* **Use ViewBinding or DataBinding** to simplify UI updates.

**Pseudocode:**

// ViewModel class to manage UI-related data in a lifecycle-conscious way

class WeightViewModel extends ViewModel {

private MutableLiveData<List<WeightEntry>> weightData; // Observable data holder for weight entries

// Getter that returns observable LiveData to be observed by the UI

public LiveData<List<WeightEntry>> getWeights() {

if (weightData == null) {

weightData = new MutableLiveData<>(); // Initialize LiveData

loadWeights(); // Load initial data from repository

}

return weightData;

}

// Method to fetch weight entries from a data source (e.g., Room DB, in-memory list)

private void loadWeights() {

// TODO: Implement logic to fetch data from repository and post to weightData

}

// Adds a new weight entry and updates LiveData so UI reflects the change

public void addWeightEntry(WeightEntry entry) {

// TODO: Add entry to data source and refresh weightData with updated list

}

}

// MainActivity observes the ViewModel and updates the UI when data changes

public class MainActivity extends AppCompatActivity {

private WeightViewModel viewModel; // Reference to the ViewModel

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

// Get instance of ViewModel scoped to this Activity

viewModel = new ViewModelProvider(this).get(WeightViewModel.class);

// Observe the weight data and reactively update the UI when it changes

viewModel.getWeights().observe(this, weights -> {

// TODO: Bind the list of weights to a RecyclerView or UI element

});

}

}

For this category of enhancement, consider improving a piece of software, transferring a project into a different language, reverse engineering a piece of software for a different operating system, or expanding a project’s complexity. These are just recommendations. Consider being creative and proposing an alternative enhancement to your instructor.

Think about what additions to include to complete the enhancement criteria in this category. Since one example option is to port to a new language, that is the kind of scale that is expected. This does not mean you need to port to a new language but instead have an equivalent scale of enhancement. Underlying expectations of any enhancement include fixing errors, debugging, and cleaning up comments, but these are not enhancements themselves.

* + 1. Explain how the planned enhancement will **demonstrate** specific **skills** and align with course outcomes.
       1. Identify and describe the specific skills you will demonstrate that align with the course outcome.

**Skills Demonstrated:**

* Clean architectural thinking using **MVVM**.
* Improved UI/data separation and reusability.
* Use of **LiveData, ViewModel**, and **lifecycle-aware components** in Android.
* Proper refactoring of legacy code into modular and testable components.
  + - 1. Select one or more of the course outcomes below that your enhancement will align with.

**Course Outcomes Addressed:**

* **Design and evaluate computing solutions** using sound software design principles.
* **Demonstrate an ability to use innovative tools and techniques** (e.g., MVVM, LiveData, ViewModel) that deliver real-world application value.
* **Develop a security mindset** by structuring the app for easier input validation and error management through separation of concerns.

Course Outcomes:

1. Employ strategies for building collaborative environments that enable diverse audiences to support organizational decision-making in the field of computer science.
2. Design, develop, and deliver professional-quality oral, written, and visual communications that are coherent, technically sound, and appropriately adapted to specific audiences and contexts.
3. Design and evaluate 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.
4. Demonstrate an 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.
5. Develop 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.
   1. **Category Two:** Algorithms and Data Structures
6. **Select an artifact** that is **aligned with the** algorithms and data structures **category** and explain its origin. Submit a file containing the code for the artifact you choose with your enhancement plan. You may choose work from the courses listed under Category One.

**Artifact:**

The artifact is the same Android mobile app project, **FLC Weight Tracker**, originally created for **CS 360: Mobile Architecture and Programming**. The app allows users to record and display their weight history, but it currently lacks analytical features that process or summarize the data.

For this category, the focus is on enhancing the **logic layer** by adding features that compute trends or insights based on user input.

1. **Describe** a practical, well-illustrated **plan** for enhancement in alignment with the category, including a pseudocode or flowchart that illustrates the planned enhancement.

**Enhancement Goal:**

Add a feature that calculates **weekly weight trends**, such as average weight change per week, using arrays and simple statistical analysis. This demonstrates understanding of data structures (e.g., List<WeightEntry>) and algorithmic thinking.

**Plan Summary:**

* Store all weight entries as objects in a List<WeightEntry> sorted by date.
* Implement logic to:
* Group entries by week.
* Calculate weekly averages.
* Compute the difference between the first and last week for trend detection.

**Pseudocode:**

// Create a map to group weight entries by their corresponding week number

Map<Integer, List<Float>> weeklyWeights = new HashMap<>();

// Loop through each weight entry and organize them by week

for (WeightEntry entry : weightEntries) {

int weekNum = getWeekOfYear(entry.getDate()); // Get the week of the year for the entry

weeklyWeights.putIfAbsent(weekNum, new ArrayList<>()); // Initialize list if week not yet in map

weeklyWeights.get(weekNum).add(entry.getWeight()); // Add weight to the appropriate week's list

}

// Create a list to store weekly average weights

List<Float> weeklyAverages = new ArrayList<>();

// Loop through each week's entries and calculate the average

for (int week : weeklyWeights.keySet()) {

List<Float> weights = weeklyWeights.get(week);

float avg = calculateAverage(weights); // Custom method to compute average of weights

weeklyAverages.add(avg); // Store the average in the list

}

// Calculate the overall trend: difference between last and first week's average

float overallTrend = weeklyAverages.get(weeklyAverages.size() - 1) - weeklyAverages.get(0);

For this category of enhancement, consider improving the efficiency of a project or expanding the complexity of the use of data structures and algorithms for your artifact. These are just recommendations. Consider being creative and proposing an alternative enhancement to your instructor. Note: You only need to choose one type of enhancement per category.

Think about what additions to include to complete the enhancement criteria in this category. Since one example option is to port to a new language, that is the kind of scale that is expected. Perhaps you might increase the efficiency and time complexity of an algorithm in an application and detail the logic of the increased time complexity. Remember, you do not need to port to a new language but instead have an equivalent scale of enhancement. Underlying expectations of any enhancement include fixing errors, debugging, and cleaning up comments, but these are not enhancements themselves.

1. Explain how the planned enhancement will **demonstrate** specific **skills** and align with course outcomes.
   1. Identify and describe the specific skills you will demonstrate to align with the course outcome.

**Skills Demonstrated:**

* Use of **data structures** (Lists, Maps) to organize information.
* Implementation of **custom algorithmic logic** (e.g., grouping, averaging, and time-based calculations).
* Understanding of **time complexity** and efficiency in loops and data handling.
  1. Select one or more of the course outcomes listed under Category One that your enhancement will align with.

**Course Outcomes Addressed:**

* **Design and evaluate computing solutions** using algorithmic principles.
* **Use innovative techniques and tools** to derive meaningful patterns from user data.
* **Communicate results effectively** through UI elements (e.g., weight trend summaries, progress insights).
  1. **Category Three: Databases**
     1. **Select an artifact** that is **aligned with the** databases **category** and explain its origin. Submit a file containing the code for the artifact you choose with your enhancement plan. You may choose work from the courses listed under Category One.

**Artifact:**

The artifact is the **FLC Weight Tracker** mobile application, originally created in **CS 360: Mobile Architecture and Programming**. In the original version, weight entries were stored in memory only, which meant all data was lost on app close. To enhance its capability and realism, I will implement **Room**, Android’s SQLite-based persistence library, for managing the weight records in a structured and queryable database.

* + 1. **Describe** a practical, well-illustrated **plan** for enhancement in alignment with the category, including a pseudocode or flowchart that illustrates the planned enhancement.

**Enhancement Goal:**

Integrate a **Room database** into the app to enable full local data persistence, structured schema management, and lifecycle-safe data access.

**Plan Summary:**

* Define an **Entity** class to represent the WeightEntry table.
* Create a **DAO (Data Access Object)** interface with @Insert, @Query, and @Delete methods.
* Build a **RoomDatabase** class to initialize the database.
* Modify the **Repository layer** to interact with Room instead of memory.
* Ensure **ViewModel** accesses data via Repository and **LiveData**.

**Pseudocode:**

// Entity class representing a row in the 'weight\_entries' table

@Entity(tableName = "weight\_entries")

public class WeightEntry {

@PrimaryKey(autoGenerate = true) // Auto-incremented unique ID

private int id;

private float weight; // Weight value entered by the user

private Date date; // Date of the weight entry

// Standard getter and setter methods

}

// DAO (Data Access Object) defines the queries and operations on the database

@Dao

public interface WeightEntryDao {

@Insert

void insert(WeightEntry entry); // Insert a new weight entry

@Query("SELECT \* FROM weight\_entries ORDER BY date DESC")

LiveData<List<WeightEntry>> getAllEntries(); // Retrieve all entries in descending order by date

@Delete

void delete(WeightEntry entry); // Delete a specific entry

}

// Room Database configuration class for the local database

@Database(entities = {WeightEntry.class}, version = 1)

public abstract class AppDatabase extends RoomDatabase {

public abstract WeightEntryDao weightEntryDao(); // Expose DAO to other components

}

// Repository class abstracts data operations and provides a clean API to the ViewModel

public class WeightRepository {

private WeightEntryDao dao;

public WeightRepository(Application app) {

// Build the Room database instance named 'weight\_db'

AppDatabase db = Room.databaseBuilder(app, AppDatabase.class, "weight\_db").build();

dao = db.weightEntryDao(); // Get the DAO to access data operations

}

public LiveData<List<WeightEntry>> getAllEntries() {

return dao.getAllEntries(); // Expose entries to the ViewModel

}

public void insertEntry(WeightEntry entry) {

// Run this operation on a background thread (not shown here)

}

}

For this category of enhancement, consider adding more advanced concepts of MySQL, incorporating data mining, creating a MongoDB interface with HTML/JavaScript, or building a full stack with a different programming language for your artifact. These are just recommendations; consider being creative and proposing an alternative enhancement to your instructor. Note: You only need to choose one type of enhancement per category.

Think about what additions to include to complete the enhancement criteria in this category. Since one example option is to port to a new language, that is the kind of scale that is expected. Perhaps you might increase the efficiency and time complexity of an algorithm in an application and detail the logic of the increased time complexity. Remember, you do not need to port to a new language but instead have an equivalent scale of enhancement. Underlying expectations of any enhancement include fixing errors, debugging, and cleaning up comments, but these are not enhancements themselves.

* + 1. Explain how the planned enhancement will **demonstrate** specific **skills** and align with course outcomes.
       1. Identify and describe the specific skills you will demonstrate that align with the course outcome.

**Skills Demonstrated:**

* Design and implementation of a **relational database schema** (Entity class).
* Use of **Room** ORM for efficient local data access in Android.
* Applying **LiveData** to observe database changes in real time.
* Understanding of **data persistence**, threading, and lifecycle-aware architecture.
  + - 1. Select one or more of the course outcomes listed under Category One that your enhancement will align with.

**Course Outcomes Addressed:**

* **Design and evaluate computing solutions** by implementing structured storage and access logic.
* **Use well-founded tools and techniques** (Room, DAO, LiveData) to deliver industry-aligned functionality.
* **Anticipate security concerns** by reducing data loss and enforcing input validation at the database level.

1. **ePortfolio Overall Skill Set**
   1. Accurately describe the **skill set** to be illustrated by the **ePortfolio** **overall**.
      1. Skills and outcomes planned to be illustrated in the code review

In the code review, I will examine the original and enhanced versions of my mobile application, focusing on architecture, code quality, and design decisions. I will highlight how I refactored procedural code into an **MVVM architecture**, introduced **Room for persistent storage**, and implemented **data grouping and analysis logic.** Through this, I will demonstrate:

* Structured software design using MVVM
* Effective use of data structures and algorithms
* Clean, maintainable, and well-commented code
* Security and error-handling improvements

**Course Outcomes Addressed:**

* Evaluate and improve computing solutions using software engineering principles
* Use modern tools and techniques (e.g., ViewModel, Room)
* Anticipate vulnerabilities through separation of concerns and modular design
  + 1. Skills and outcomes planned to be illustrated in the narratives

In the three enhancement narratives, I will explain:

* Why each enhancement was needed
* What changes were made and how they work
* How each change demonstrates specific technical skills and course outcomes

These narratives will focus on **communication and reflection**, ensuring I explain technical decisions in a way that is accessible and professionally relevant.

**Skills Demonstrated:**

* Technical articulation of software changes
* Connection between code-level changes and strategic goals
* Audience-appropriate documentation (developer, reviewer, employer)

**Course Outcomes Addressed:**

* Deliver technically sound and professionally written communications
* Relate enhancements to organizational and user value
* Align technical decisions with audience needs
  + 1. Skills and outcomes planned to be illustrated in the professional self-assessment

In my self-assessment, I will reflect on my journey from starting the CS program to completing this capstone. I will highlight:

* The foundational and advanced skills I’ve developed
* My shift in mindset from mechanical engineering to computing
* How I plan to apply what I’ve learned to future roles in data science or software development

**Course Outcomes Addressed:**

* Demonstrate lifelong learning and self-directed improvement
* Align personal growth with field expectations
* Position myself professionally with a clear, honest self-assessment