**Created:** CS 360 – Mobile Architecture and Programming

### **Artifact 1: YourWeightHero Mobile App (Software Engineering and Design)**

**Description:**  
 YourWeightHero is a mobile Android application developed in Java using Android Studio. It allows users to track daily weight changes, offering a login screen, SQLite data storage, dynamic UI, and SMS notification features.

**Justification for Inclusion:**  
 I chose this artifact to represent my software engineering and design capabilities because it demonstrates a complete cycle of app development—from user interaction and security to data management and usability. The enhancements I made, such as input validation, UI restructuring, and SMS integration, significantly improved the app’s user experience and reliability. These enhancements reflect my growth in designing scalable, maintainable, and professional-quality applications.

**Reflection on the Enhancement Process:**  
 Enhancing this app taught me the importance of modular design and clear UX structure. I learned how minor UI improvements and feedback elements (like button disabling) significantly improve usability. The biggest challenge was implementing SMS functionality across various Android versions, which required me to dig into documentation and compatibility issues. I also incorporated instructor feedback to clarify user input flows and optimize the visual hierarchy.  
 **Course Outcomes Met:**

* Demonstrated secure, scalable software engineering
* Applied best practices in mobile UI design and modular code structure
* Delivered professional-quality application aligned with industry expectations  
   **Partially Met:** None—this artifact strongly aligned with all intended outcomes.

### **Artifact 2: Custom Sorting Logic (Algorithms and Data Structures)**

**Description:**  
 This artifact consists of custom sorting functionality embedded within the YourWeightHero app. It allows sorting of weight entries by date or value and handles edge cases like duplicates and empty fields.

**Justification for Inclusion:**  
 I selected this component because it showcases my ability to apply algorithmic principles in a real-world context. The sorting algorithm was enhanced to improve efficiency, handle multiple sorting criteria, and gracefully manage invalid inputs. These changes emphasized my grasp of list operations, comparators, and performance tuning in Java.

**Reflection on the Enhancement Process:**  
 The enhancement process required deep review of sorting methods like selection sort and leveraging Java's Comparator interfaces. I learned how algorithm refinements can affect user experience, especially in responsiveness and accuracy. Testing edge cases helped me become more thorough in quality assurance. The major challenge was balancing simplicity with performance, and I addressed this by removing redundant operations and improving logic readability.  
 **Course Outcomes Met:**

* Applied data structures and algorithmic concepts to practical problems
* Managed design trade-offs between performance and code clarity  
   **Partially Met:** Fully met all related outcomes for this category.

### **Artifact 3: SQLite Integration in YourWeightHero (Databases)**

**Description:**  
 This component of the YourWeightHero app implements SQLite for local data storage. It includes table creation, data insertion/deletion, and query logic, all linked with dynamic UI updates.

**Justification for Inclusion:**  
 I selected this artifact to demonstrate my ability to design and manage databases in mobile applications. The enhancements involved optimizing queries, integrating input validation to prevent bad data, and refining UI interactions based on database state. This showcases my skills in relational database integration, CRUD operations, and UI-data syncing.

**Reflection on the Enhancement Process:**  
 One of the most important lessons was realizing how fragile database interactions can be when input handling is not properly designed. Debugging issues related to UI not reflecting DB changes taught me how to structure cleaner database helper classes and improve feedback mechanisms. Incorporating user feedback and testing showed the importance of graceful error handling and data integrity.  
 **Course Outcomes Met:**

* Demonstrated competence in designing and managing databases
* Integrated database logic with UI and ensured dynamic updates  
   **Partially Met:** Initially lacked some indexing performance improvements, but enhancements corrected that.