Milestone Two Narrative

Christopher Carnell

Southern New Hampshire University

CS-499 Computer Science Capstone

Professor Maryann Krupa

November 17, 2024

Milestone Two Narrative

The first artifact is the MainActivity.java file from my WeightWatcher Android application, serving as the main activity that handles user authentication. I developed this file in October 2024 as part of a project to build a weight-tracking app. It allows the user to log in, register, and navigate to other parts of the app based on their authentication status.

I selected MainActivity.java for my ePortfolio because it encapsulates key aspects of software development, including user authentication, code refactoring, and exception handling. Specific components that showcase my skills are the code refactoring to create reuseable code, adding exception handing, removing unused variables, adding comments. Specifically, I removed the unused REQUEST\_SMS\_PERMISSION variable, I removed the input validation from the login and register methods that were duplicated and created a helper method named validateInputs. I created helper methods handleLogin and handleRegistration to separate the logic for login and registration. I wrapped the database operations in try-catch blocks in handleLogin and handleRegistration methods to implement exception handling. I added method level comments to explain their purpose, parameters, and return values. Additionally, I addressed a bug where the user wasn’t automatically logged in after registration by creating a new user object after a successful registration and using that object to proceed as if the user had just logged in.

The enhancements I made to this artifact met the course outcomes I planned to address. By refactoring the code and improving documentation, I aligned with course outcome two, demonstrating my ability to design and develop professional-quality software. Fixing the automatic login bug allowed me to showcase my problem-solving skills, meeting course outcome three. Implementing exception handling contributed to developing a security mindset, aligning with course outcome five. An additional item I addressed was the automatic login bug which I found during testing which didn’t always work and sometimes caused the application to crash.

I learned I need to plan out my coding better instead of throwing things together on the fly. For example, I had reused the same code in multiple places, had I planned it out, or at least conducted my own code review, I would have seen how much extra work I was doing. I learned I need to review my code with an eye towards refactoring to make things more maintainable. I also need to use more comments, not only inline and method level, but headers that explain the purpose of the code. The only real challenge was refactoring the code without creating new issues. I overcame the challenge by fixing one thing at a time and then testing the application after each change.

The second artifact is the DataDisplayActivity.java file which is a central component of my application. It was initially created during my CS-360 course in October 2024. This activity displays the user’s weight entries and goal weight. Allows the user to add new entries, set or change their goal weight, and track their progress. It also checks for goal achievement and sends SMS notifications if the user has opted in.

I selected this artifact because it demonstrates my skills in software development in enhancing code robustness and maintainability. Specifically, I added exception handling around the database operation to prevent crashes and improve stability. I added method-level comments and removed unused constants which enhanced the code readability and facilitates easier maintenance. Finally, I improved the user experience by adding feedback through informative messages. Example pictured below of the method checkAndDisplayGoalWeight before and after enhancement with exception handling.

pre-enhancement:

Pre-enhancement


Post-enhancement: A screen shot of a computer program

Description automatically generated

I met the outcomes I planned to address with these enhancements. By enhancing exception handling and code documentation, I demonstrated my ability to develop professional quality software that is reliable and maintainable which aligns with course outcome two. By implementing exception handling I solved the problem of potential crashes due to unhandled exceptions which showcases my problem-solving skills which align with course outcome three.

What I learned from the process was the importance of adding exception handling, it’s critical to anticipate and manage potential errors which improves the application stability. Challenge was ensuring I didn’t add new bugs while enhancing the code.

The third artifact is DataAdapter.java file, which is an adapter class used in my weight tracking application. This artifact was created during my CS-360 course in October 2024. It binds weight entry data to the RecyclerView in the DataDisplayActivity, managing the display of weight entries in a list format. It handles user interactions such as deleting entries from both the list and the database.

I selected this artifact because it highlights my skills in software development, particularly in optimizing code performance and enhancing user experience. Specific components that showcase my abilities include exception handling, performance optimization, user experience enhancements, and code quality. Exception handling was implemented by adding try-catch blocks around the deletion operation to handle potential exceptions improving stability. Performance was optimized by removing unnecessary notifyItemRangeChanged call which optimized the RecyclerViews performance during item deletion. User experience was enhanced by adding a confirmation dialog before deleting an entry to prevent accidental deletions which demonstrates attention to user-centered design. Code quality was enhanced by removing unused imports and adding clear documentation to the code.

I learned about the importance of performance optimizations by understanding how unnecessary method calls can impact performance. I also learned of the importance of adding confirmation dialogs to prevent unintended user actions. The challenge faced was adding the confirmation dialog, particularly regarding the use of the correct context and handling adapter positions. I overcame the challenge by referring to the Android developer documentation and testing changes as I made them.

The last artifact is AddEntryDialog.java, which is a dialog fragment in my application that was created for the CS-360 class in October 2024. This component allows users to add new weight entries by entering their current weight. The dialog collects the user’s weight, records the current data automatically, and passes the data back to the DataDisplayAdapter to be saved in the database.

I chose this artifact because it highlights by abilities to enhance application robustness and improving user experience through input validation and error handling. Specifically, skills include input validation, exception handling, and user experience enhancement. I displayed input validation skills by implementing validation to ensure that weights entered are valid, positive numbers, and preventing invalid data from being processed. Adding try-catch blocks around critical operations to handle potential exceptions such as NumberFormatException, enhanced the applications stability and displayed my exception handling skills. Providing informative feedback messages to guide the users in entering correct data, I also improved usability.

I met the outcomes I intended to address with these enhancements. Course outcome two by improving input validation and error handling, and course outcome three by addressing potential crashes due to invalid inputs.

I learned about the importance of input validation, for example during testing I realized that I could enter anything I wanted as a weight, for example I entered 0 and it worked. I also learned that the user may not always understand exactly what they should be entering so feedback is important.

The challenge I faced was accounting for various invalid inputs such as empty strings, non-numeric characters, and negative numbers. The challenge was overcome by having a think on all the possible inputs a user could enter. Another challenge I had was the on-screen keyboard that stopped working somewhere along the way, this one I couldn’t figure out, so I switched to testing on a physical device where the keyboard worked fine. I was playing with making a custom input keyboard only show valid inputs.