Brian Voo

Professor Penmatsa

CS 499

16 Nov 2025

3-2 Milestone Two - Enhancement One - Software Design and Engineering

For my Category One enhancement, I chose to improve my CS-360 Mobile App, an Android-based inventory management system originally written in Java. The initial version of the project allowed users to log in, add items, view inventory, and send SMS alerts. However, the authentication system stored passwords in plain text, the user interface allowed unauthorized access to certain actions, and the overall application flow lacked consistency and security. My goal was to enhance this artifact by redesigning authentication, implementing role-based access control, strengthening input validation, and improving the UI and session flow to better reflect professional software engineering standards.

The first major enhancement involved securing the authentication system. I replaced plain-text password storage with SHA-256 hashing so that user credentials are never stored or compared in raw form. To validate this workflow, I tested multiple login scenarios by confirming that newly created passwords were consistently hashed before storage and that authentication only succeeded when the hashed input matched the stored hash. I also verified that incorrect credentials always failed authentication, ensuring the hashing process functioned reliably. A default administrative account was seeded using a hashed password, allowing secure system management while preventing exposure of sensitive credentials.

Input validation was added across multiple layers of the application to improve data integrity and defensive programming. Usernames must meet minimum length requirements and cannot contain spaces, while passwords must meet defined complexity rules. Inventory quantity fields now enforce numeric and positive-value constraints. I tested these rules by intentionally submitting invalid inputs to confirm that the app blocked unsafe or malformed data before it could reach the database. These validation checks reduce the risk of runtime errors, corrupted records, and unintended application behavior.

Role-based access control was another core enhancement. I added a role attribute to user records and enforced admin-only permissions for modifying or deleting inventory items. Standard users can view inventory data but cannot alter it. This restriction is enforced both at the UI level and within the application logic, ensuring that unauthorized users cannot bypass restrictions through navigation or unexpected execution paths. Testing focused on confirming that standard users could not trigger update or delete operations even when UI elements were manipulated or reloaded.

I also refined the application's UI flow to improve usability and session security. Once authenticated, the login tab is hidden to prevent unnecessary navigation. A logout button was added to allow users to explicitly end their session, and the application now clears authentication state every time it is reopened, requiring users to log in again rather than resuming a prior session automatically. These changes were tested by closing and reopening the app to confirm that protected screens were no longer accessible without reauthentication.

One challenge during this enhancement was balancing usability with security. Implementing strict validation and session controls required careful testing to ensure the app remained intuitive while still enforcing safeguards. In future iterations, I would like to add more descriptive error messaging, audit logging for user actions, and improved session timeout handling. Overall, this enhancement demonstrates growth in secure software design, defensive programming, role-based authorization, and user-centered UI refinement, aligning strongly with the software engineering outcomes of the Computer Science program.