Juan Cervantes Ortiz

CS-499

January 23, 2025

Milestone 2

For category 1, Software Engineering and Design I have chosen my CS-360 Inventory application. This is an app that allows the user to create an account and manage their inventory. The user can view, add, update and delete items from their inventory. Furthermore, the system will send the user a notification via text when an item is deleted or updated. The application is developed using the JAVA program language and uses two SQLite databases. There is a database for user login information and another for the user inventory. The application was programmed using Android Studio and was tested on the Virtual Android Emulator as well as a physical Motorola Stylus 2024 development device.

There are many reasons why I have chosen this item to be included in my ePortfolio. First it has various components that must work together, and the system needs to be designed with consideration for the users. This project shows my ability to properly create, read, update, and delete items using an SQLite database. The class functions must properly interact with the database to ensure the action is performed properly, but also that the app view updates properly. Second it shows that I can properly use the permissions manager and session manager that Android have. It also demonstrates that I can create a secure authentication system that allows users to login and stay logged in, as well as prevent incorrect input from crashing the app. The project also has a clear UI layout that makes it easy for users to use.

There are various improvements that were made to the program. First was the login method now keeps users logged in. In the original project the application would have the user log in, and then when the app was closed and opened it would have them log in again. The enhancement uses the Android Session manager to save the users login information, and once they are successfully logged in or signed up the user will be taken to the home page automatically instead of being asked to log in each time. The system begins by creating keys to store the user's information. The system will, at initiation, check to see if there is data in shared preferences. If there is data in shared preferences, then the system will jump the user to the main screen. If there isn't any information in shared preferences, then the system will check the user login information and if it is correct than it will put those values into shared preferences before taking the user to the home page. This makes it so the next time the user opens the app they will be taken to the home page without needing to sign in. Along with the new login method, there is also a log out button on the home page, which when pressed will clear out the values saved in the shared preferences and take the user back to the log in screen.

The next improvement that was made was to create a different notification system for the application. The original project had an SMS notification system that would alert the user if an item was updated or deleted. This notification method is ineffective because it would send a text message anytime anything was updated or deleted. The project should only send the user a message when an item quality is updated, or if something is deleted. For the enhancement, the system now only sends an app notification if an items quantity is updated and is less than or equal to 1. The system still uses the SMS method to send the user an SMS if something from the inventory is deleted. I decided to keep it that way so that users can have a record of what items were removed in case something was removed on accident.

Another improvement was to add an input validation class that ensures the input is valid before it is passed to the SQLite system. This was done by adding a separate input InputValidation.java class. The class has 3 methods that check the item inputs. If they are not empty, then the system will either return true or false. If the user input is true that means the input is empty so the system will set a default value of either “No name”, “N/A”, or “1” depending on input field that is empty. This class is also used in the update method.

I believe that I have met the course outcomes I planned to meet with this enhancement in Module One. The course outcomes that pertain to this enhancement are 2, 3, 4, and 5. The project focuses on anticipating users’ needs by providing them with features that they may want. The new project also has a separate validation class that checks user input to ensure they are safe before they get passed into the database. This makes it so there is one class that the others can use to validate their input. Using Android Studio with Java provides many different tools that allow for scalability and the launch of the application on the Google Play Store. Lastly the system uses input validation to ensure information is properly configured before the query is sent to the database.

Throughout the process of enhancing and modifying the artifact there are many things I learned and struggled with while creating and improving it. First, I had to learn how to use the notification manager to create and send notifications. Something that I learned was that I had to create the notifications in the class that I would use them in. Another thing I learned was how to use the session manager. This is what allowed me to keep the user logged in. The biggest struggle was just trying to get the app to work without crashing. Another struggle was implementing the changes to the code while ensuring that the new features work without disrupting the rest of the code.

References

Create a notification : views : android developers. Android Developers. (n.d.-c). https://developer.android.com/develop/ui/views/notifications/build-notification

GeeksforGeeks. (2024c, July 12). *Session Management in Android with example*. https://www.geeksforgeeks.org/session-management-in-android-with-example/?ref=header_outind