

Enhancement Three: Databases

By Ivy Pokorny

Artifact Description:

What is it? When was it created?

For my Algorithms and Data Structures enhancement project, I selected **Stock't**, an application developed in CS 360: Mobile Architecture and Programming, aimed at streamlining inventory management for flower shops. This app addresses critical user needs, including low stock alerts, easy inventory tracking, and efficient communication of inventory levels through SMS notifications. Users can enter and modify inventory information directly on their phones, adjusting counts as products are sold or lost. Additionally, the app notifies users when any product falls below a stock threshold of 50 units, ensuring they remain informed about inventory status. This project not only highlights my technical skills but also reflects my commitment to creating user-centric solutions that enhance operational efficiency. This project includes multiple screens. There is a Login screen and a Signup screen for entering into the User Database. The Home screen displays item analytics from Enhancement 1 of this class and it allows the Graph screen to be entered as well. The Items screen allows users to Delete, or adjust Quantity on the inventory. It also allows the user to Add Items on a separate screen. In the previous milestone another screen was added which allows changes made to the inventory to be undone or redone via a stack. Thus, any simple mistakes can be undone.

Justification for Inclusion

Why did you select this item? What specific components of the artifact showcase your skills and abilities in algorithms and data structure? How was the artifact improved?

I selected this artifact because it demonstrates my skills in databases, particularly through the implementation of a comprehensive History Database for tracking inventory changes. Previous functionality in this app included a database of the Lists of Items in the inventory, their name, price, count, and category, which were stored and updated using an SQLite database as well as an SQLite database for the user information. However, these databases did not provide a way to track the history of changes, making it difficult for users to review or analyze past adjustments. The new History Database not only enhances user experience by providing a detailed log of inventory changes but also showcases my understanding of efficient database design and data management within the application. Features of the database include adding unique Item IDs to records which ensures that all inventory adjustments are linked to specific items, enabling precise tracking and reporting. Adjustment timestamping uses a DateTime field to automatically log when changes occur, making it easy to analyze trends over time. And reason codes for adjustments allow the user to capture the rationale for each inventory change which adds valuable context for users and supports better decision-making.

The additional features can be better explained by the figures below. To begin with however the added database includes the following fields:

- **ID:** Primary key, auto-incremented for unique identification.
- **Item_ID:** Links the changes to specific items in the inventory database.
- **Item_Name:** Provides a clear reference to the affected item.
- **Quantity_Change:** Records the amount by which the inventory was adjusted.

- **Adjustment_Time:** A DateTime field, defaulting to the current timestamp, to log when changes occur.
- **Adjustment Reason:** Captures the rationale behind the change, such as “Sold,” “Damaged,” “Store Use,” “Other” or “Recieved Shipment.”

The addition of this database enables the app to maintain an accurate historical log of inventory adjustments. Users can now review the reasons and timing of changes, providing valuable insights into inventory trends and operational bottlenecks.

The Inventory screen was altered to make room for the improved quantity submission button which shows a reason pop-up. In figure 1 below you can see the submit button next to every inventory item. If the user alters the Item Count from what is in the database then the Submit button becomes teal. From this point pressing the submit button pops up a window asking the user to select a reason for the change. Touching any of those reasons completes the inventory change submission process and also adds the inventory change to the database.

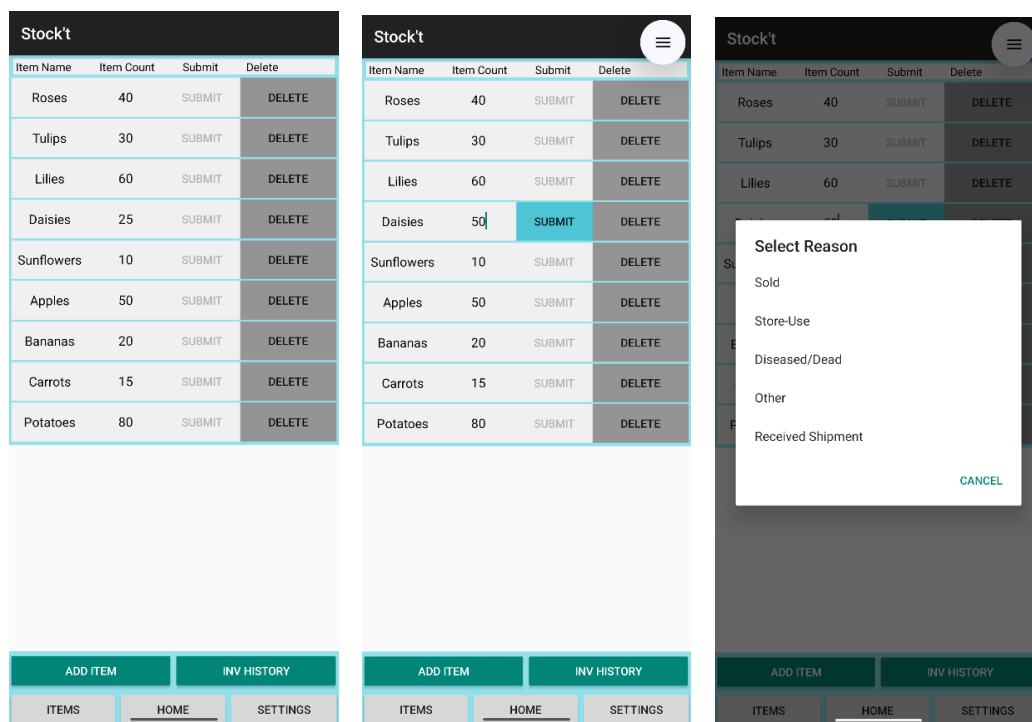


Figure 1: Inventory submission flow diagram

The new inventory/item graph screen allows for simple data visualization of one item through the history of the items' inclusion within the database. Selecting the item's name in the Inventory Screen will open the corresponding item's graph. Figure 2 below shows the history of sales with the x-axis showing the time and date, and the y axis showing either the number of items sold or the monetary value sold. Once in this screen the user can pinch-to-zoom for closer or broader examination of trends. The bottom toggle button is what allows the user to easily swap between dollars and count sold.

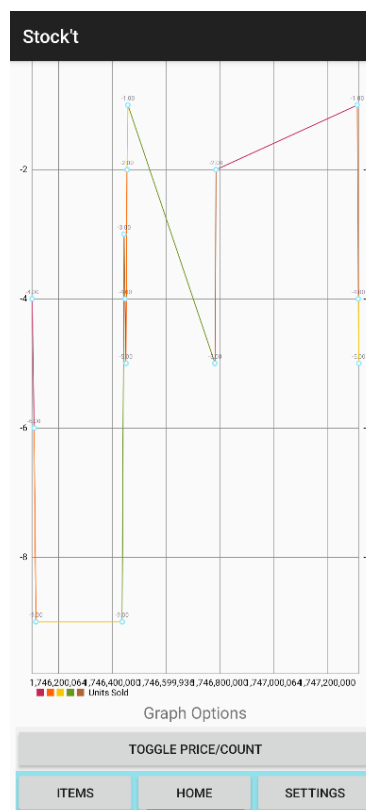


Figure 2: Sales over Time graph

This enhancement provides flower shop owners with actionable insights, enabling them to identify sales patterns and adjust their inventory strategies accordingly.

Course Outcomes and Updates

Did you meet the course outcomes you planned to meet with this enhancement in Module One?

Do you have any updates to your outcome-coverage plans?

Through these enhancements, I successfully met several course outcomes, particularly in designing, developing, and evaluating computing solutions using databases.

The addition of the History Database exemplifies my ability to design database structures that support scalability and data analysis. While these modifications were not originally planned in Module One, feedback from Modules One and Two helped shape the final implementation.

Looking ahead, my future plans include better data visualization. I'd also like to use more advanced querying in order to generate detailed reports, such as identifying the most frequently adjusted items or the reasons for inventory losses. Hopefully, in the future this database can be used to feed predictive analytics with the goal of helping users forecast inventory needs.

Reflection on the Enhancement Process

What did you learn as you were creating it and improving it? What challenges did you face?

The enhancement process has been a valuable learning experience, particularly in the realm of database design and optimization.

One of the key challenges I faced was ensuring that the database and the data handling was consistent particularly with DateTime in SQLite. I needed to make sure that the data type that java used to hold the date time was accurate and useful for sorting the data in the java implementation. I used LocalDateTime for this so I could easily convert to and from String format and to convert to EpochSeconds for use in the graph. Through this process I learned about

the different SQL date or time type that were available to me, as well as the java date and time classes available. I used DateTime for SQLite instead of Timestamp because I thought that for the time being the added time zone would complicate the process unnecessarily. Additionally I found myself searching for the best graph to use to visualize the data. Bar charts could have proven useful given a more complex data structure for handling the grouping of data, but the line graph allowed me to display every sale or adjustment easily and clearly over time.

Through this process, I gained a deeper understanding of database design principles, the nuances of working with date and time data in SQLite and Java, and the importance of thoughtful data visualization. These challenges not only enhanced my technical proficiency but also reinforced my ability to make informed design decisions that balance complexity, usability, and functionality. Overall, the enhancement process has strengthened my confidence in developing scalable and user-focused database solutions. It has also provided me with valuable insights that I will carry into future projects, particularly those requiring robust data management and visualization capabilities. This experience has solidified my commitment to creating software that is both technically sound and user-centric.