Artifact 3 Enhancements: Database CS 360 Inventory App – SQLite Database

The artifact I chose for database enhancements is an Inventory Management System developed as part of my coursework for CS360. This system is designed to help users manage and track inventory items, including their names, quantities, categories, and dates added. The initial version of this system was created during the course's earlier phases, focusing on basic functionality such as adding, updating, and deleting inventory items.

I chose to include this Inventory Management System in my ePortfolio because it represents my ability to design, implement, and enhance a real-world application using Android development tools and SQLite for database management. The enhancements made to this system, particularly the addition of category tracking and date management, demonstrate my skills in database schema design, SQL query formulation, and Android UI integration.

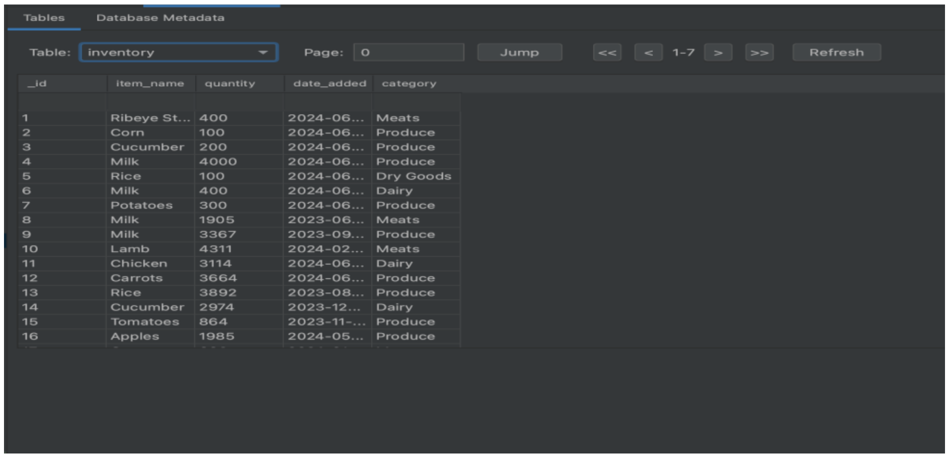
Database Schema Design

One of the significant enhancements I made was the addition of the `category` and date_added columns to the inventory table. This required careful planning and modification of the database schema.

1. Extended the Schema: Added new columns to the existing table, ensuring the database could store additional data without compromising existing functionality.

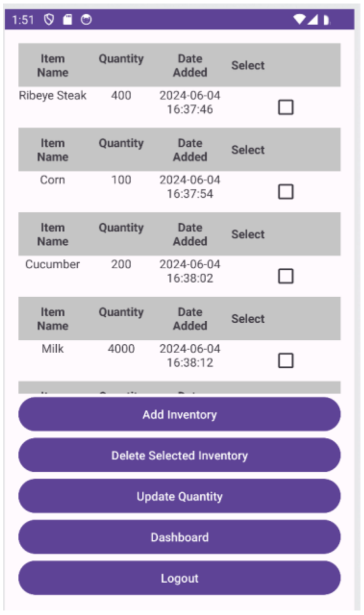
   - Category: Added a `category` column to categorize inventory items.

   - Date Added: Added a date_added column to record when each inventory item was added.



_id	item_name	quantity	date_added	category
1	Ribeye St...	400	2024-06...	Meats
2	Corn	100	2024-06...	Produce
3	Cucumber	200	2024-06...	Produce
4	Milk	4000	2024-06...	Produce
5	Rice	100	2024-06...	Dry Goods
6	Milk	400	2024-06...	Dairy
7	Potatoes	300	2024-06...	Produce
8	Milk	1905	2023-06...	Meats
9	Milk	3367	2023-09...	Produce
10	Lamb	4311	2024-02...	Meats
11	Chicken	3114	2024-06...	Dairy
12	Carrots	3664	2024-06...	Produce
13	Rice	3892	2023-08...	Produce
14	Cucumber	2974	2023-12...	Dairy
15	Tomatoes	864	2023-11-...	Produce
16	Apples	1985	2024-05...	Produce

Figure 1-1 shows the SQL database with the newly added columns date_added and ‘category’.



Item Name	Quantity	Date Added	Select
Ribeye Steak	400	2024-06-04 16:37:46	☐

Item Name	Quantity	Date Added	Select
Corn	100	2024-06-04 16:37:54	☐

Item Name	Quantity	Date Added	Select
Cucumber	200	2024-06-04 16:38:02	☐

Item Name	Quantity	Date Added	Select
Milk	4000	2024-06-04 16:38:12	☐

Figure 1-2 shows what it looks after enhancement from inventory fragment. It now tracks the data the inventory was added. Allowing for more precise tracking.

SQL Query Formulation

Implementing inventory trend tracking and total inventory by category required writing complex SQL queries.

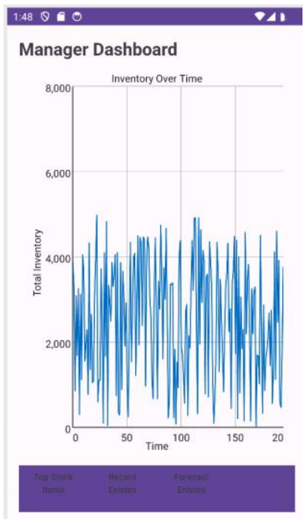


Figure 1-3 shows the Inventory Trending Tool

This involved:

1. Inventory Trend Tracking: Created SQL queries to group inventory data by the date_added field, allowing users to see trends over time.

   - Example: `SELECT date_added, SUM(quantity) as total FROM inventory GROUP BY date_added ORDER BY date_added`

2. Total Inventory by Category: Wrote queries to sum the quantities of items in each category.

   - Example: `SELECT category, SUM(quantity) as total FROM inventory GROUP BY category`

These tasks demonstrate my proficiency in SQL and my ability to manipulate and retrieve data efficiently. Integrating new database features into the Android UI involved updating the data display logic and ensuring a seamless user experience.

This included:

1. Updating Data Display Logic: Modified the application to retrieve and display the new category and date_added data fields.

- Ensured that inventory items are displayed with their categories and dates in the UI.

2. Enhancing User Interaction: Implemented UI components to allow users to view trends and categorize items effectively.

This highlights my skills in Android development, particularly in connecting the backend database with the frontend user interface to provide a cohesive user experience.

Error Handling and Debugging

Throughout the enhancement process, I encountered and resolved various issues, such as handling missing columns and ensuring the correct initialization of cursors. Specific challenges and solutions included:

1. Handling Missing Columns: Ensured that the application could handle scenarios where the new columns might be missing during the schema upgrade.

- Added conditional checks and migration scripts to add missing columns if necessary.

2. Cursor Initialization: Fixed issues related to cursor initialization to prevent crashes and ensure reliable data retrieval.

- Corrected the SQL queries and column references to match the updated schema, avoiding IllegalStateException and similar errors.

The enhancements made to this artifact align with the course objectives outlined in Module One. Specifically, I aimed to deepen my understanding of Android development, improve my database management skills, and learn to write more complex SQL queries. Through this process, I have successfully met these objectives, gaining valuable experience in extending and refining a software application.

Enhancing and modifying this artifact was a challenging yet rewarding experience. Some key learnings and challenges include:

1. Learning Outcome:

- Database Management: I learned how to effectively design and modify database schemas, ensuring data integrity and efficient retrieval.

- Complex Queries: Writing and debugging complex SQL queries taught me the importance of careful planning and testing in database operations.

- Integration and Testing: Integrating new features into the existing Android application required thorough testing and validation to ensure seamless functionality.

2. Challenges Faced:

- Schema Migration: Modifying the database schema while preserving existing data was challenging. I had to carefully plan the migration process to avoid data loss.

- Debugging SQL Issues: Encountering SQL errors due to missing columns or incorrect queries required in-depth debugging and understanding of SQL and SQLite behaviors.

- Ensuring UI Consistency: Updating the UI to reflect new data fields and ensuring consistency across various parts of the application required meticulous attention to detail.