**Android Inventory Application**

User Guide for Inventory Managers: Menu Usage and Error Handling

**Introduction**

The Android Inventory Application is designed to assist inventory managers and staff in managing inventory items efficiently on Android devices. The app provides a user-friendly interface for user authentication, viewing and editing inventory items, and switching between local and remote database modes. It uses a SQLite database for local storage and a RESTful API for remote data access, with secure API key authentication.

**User Input Handling**

Input Types:

* Username (text), password (text), and password confirmation (text for registration). Inputs are case-sensitive.
* Title (text, max 20 characters), description (text, max 40 characters), quantity (numeric, default 1 if invalid). Button clicks for increment, decrement, delete, and toggle operations.

Validation:

* Login: Invalid credentials trigger: “Incorrect, please try again.”
* Registration: Mismatched passwords show: “Supplied passwords not equal.” Duplicate usernames display: “Pick another Username.”
* Add Item: Overly long title/description fields are truncated.

**Errors:**

* Connection Failure: If the server is unreachable, the app displays: “Cannot connect to remote database.” Ensure an active internet connection and retry.
* Invalid API Key: Server-side validation failures result in HTTP 401 (Unauthorized), causing the app to retry or fall back to local mode.
* Server Errors: HTTP 404 (Not Found) or 400 (Bad Request) responses trigger toasts: “There was a problem, please try again.”

**Design Choices**

User-Friendly Interface: Clear prompts (e.g., “Login”, “Register”) and toasts guide non-technical users. The inventory list displays items in a grid with intuitive buttons (“+”, “-”, “Delete”).

Local/Remote Flexibility: The toggle option allows seamless switching between local and remote modes, accommodating offline use or server-based operations.

Security: Passwords are hashed (SHA-256) in User, and API requests use encrypted keys. SQLite queries are parameterized to prevent injection.

Error Feedback: Toasts provide immediate feedback without overwhelming users, while detailed logs aid developers in troubleshooting.

**Performance**

Local Operations: SQLite queries (via InventoryRepo) are nearly instantaneous (< 100ms for fetching 100 items). Database writes (create/update/delete) are optimized with parameterized queries.

Remote Operations: HTTPS requests depend on network latency, typically 200–500ms for a stable connection. Asynchronous tasks prevent UI freezes.

Scalability: The app handles hundreds of items efficiently in local mode. Remote mode performance depends on server capacity, but client-side caching (items list) reduces redundant requests.

**Limitations**

Login Sensitivity: Users must enter exact usernames and passwords, with no “forgot password” feature, which may frustrate staff unfamiliar with credentials.

Error Guidance: Toasts like “There was a problem, please try again” identify issues but don’t suggest specific fixes.

Remote Dependency: Remote mode requires a stable internet connection and a running server. Downtime or network issues force fallback to local mode.

Manual Data Correction: Invalid item data (e.g., overly long titles) is rejected without detailed guidance, requiring users to retry with correct inputs.

Database Reset: Local database corruption requires clearing app data, potentially losing unsynced items unless backed up remotely.

**Instructions for Use**

* Launch the App
* Login or Register
* Modify Quantity: Tap “+” to increment or “-” to decrement an item’s quantity. A toast notifies if quantity reaches zero.
* Delete Item: Tap “Delete” to remove an item. Confirm success via the updated list or a toast if it fails.
* Add Item: Tap the Floating Action Button, enter title (≤ 20 chars), description (≤ 40 chars), and quantity (numeric), then tap “Submit”. Success shows “Success”; failures show “There was a problem, please try again.”
* Toggle Database: Tap the toolbar menu and select “Toggle Remote/Local”. The title updates (e.g., “Database is Remote”), and the list refreshes. If remote fails, a toast shows “Cannot connect to remote database.”
* Exit: Press the device’s back button or close the app via Android navigation.

**Troubleshooting**

Login Issues: Verify username/password or register a new account.

Database Errors: For remote issues, check internet connection or switch to local mode. For local database crashes, clear app data (Settings > Apps > Inventory App > Storage > Clear Data) and re-login.

Item Errors: Ensure title/description lengths and quantity are valid. Retry if toasts indicate failures.

**Conclusion**

The Android Inventory Application provides a robust, user-friendly solution for managing inventory items, with flexible local and remote database support. By following this guide, inventory managers and staff can navigate the app’s menus, handle inputs, and address common errors effectively. While limitations like login sensitivity and remote dependency exist, the app’s feedback and secure design make it suitable for non-technical users.