**Prismatica Android**

Offline-first Android app for tracking devices/items with fast CRUD and optional SMS alerts. Users sign up/log in locally, manage a grid of items (with notes), and—if granted—send test SMS notifications. Designed to satisfy CS360 Project Three rubric.

**Overview**

* **Offline-first:** all data stored locally in SQLite (no network).
* **Auth:** local signup/login.
* **CRUD Grid:** add, edit, delete, view items in a RecyclerView grid.
* **Notes:** per-item notes persist.
* **SMS (optional):** runtime permission; app works fully if denied.

**Quick Start**

1. **Open in Android Studio** (Giraffe+).
2. **Run on an emulator** (API ≥ 26).
3. **Sign Up** a user → then **Log In**.
4. **CRUD:** tap **+** to add; tap pencil/delete on a card to edit/remove.
5. **SMS test screen:** tap the **“Devices”** header to open **SmsNotificationsActivity**.
   * **Deny** → app continues; send is disabled.
   * **Allow** → tap **Send Test SMS**.
     + For real delivery, run a **second emulator** and use its number (e.g., send to 5556).

**Requirements**

* **Min SDK:** 26 (Android 8.0)
* **Target SDK:** per module config (e.g., 36)
* **Tooling:** Android Studio, Gradle

**Build**

* **Debug APK:**  
  Build → Build Bundle(s) / APK(s) → Build APK(s)  
  Output: app/build/outputs/apk/debug/app-debug.apk

**App Structure (high level)**

app/

├─ java/com/prismatica/iotInterface/

│ ├─ ui/

│ │ ├─ LoginSignup.java # local auth screen

│ │ ├─ DeviceList.java # grid + dialogs for CRUD

│ │ └─ SmsNotificationsActivity.java # runtime SMS permission + send test

│ ├─ data/

│ │ ├─ Item.java # model: id, name, qty, notes

│ │ ├─ ItemRepository.java # CRUD for items (SQLite)

│ │ └─ UserRepository.java # signup/login (SQLite)

│ └─ AppDatabaseHelper.java # schema + migrations

└─ res/

├─ layout/

│ ├─ activity\_login\_signup.xml

│ ├─ activity\_device\_list.xml

│ ├─ activity\_sms\_notifications.xml

│ ├─ dialog\_item.xml # Add/Edit dialog (etItemName/Qty/Notes)

│ └─ item\_row.xml # Row: tvName, tvQty, (tvNotes), edit/delete

└─ values/

├─ strings.xml # user-visible strings

└─ colors.xml

**SQLite schema**

* users(username TEXT PRIMARY KEY, password TEXT NOT NULL)
* items(id INTEGER PK AUTOINCREMENT, name TEXT NOT NULL, qty INTEGER NOT NULL DEFAULT 0, notes TEXT)

**Permissions**

<uses-feature android:name="android.hardware.telephony" android:required="false" />

<uses-permission android:name="android.permission.SEND\_SMS" />

* Requested **at runtime** in SmsNotificationsActivity.
* App **continues to function** when denied (rubric requirement).

**How to Demo (Rubric Checklist)**

* **Login:** Sign up → relaunch → log in (valid creds pass, invalid show toast).
* **CRUD:** Add item (with notes) → shows in grid; Edit (qty + notes) → persists; Delete → disappears; relaunch → still there.
* **SMS:** Open SMS screen → **Deny** (send disabled, no crash) → **Allow** (send test, success toast/feedback).

**Testing & Debugging Tips**

* **Database Inspector:** View → App Inspection → Database Inspector → verify items.notes after Save/Update.
* **Common “R” issues:** ensure only import com.prismatica.iotInterface.R;, rebuild if ids look red, and match ids to the layout passed to setContentView() / adapter’s inflated row layout.
* **Safe parsing:** quantity uses a safe int parse to avoid crashes on empty input.

**Known Limitations**

* No dedicated **user detail** or **device data/history** screens yet (grid + dialog only).
* SMS is a manual test screen; no background scheduling.
* Local SQLite only; no sync/export.

**Future Work**

* User detail & device data views
* Thresholds per item + auto-SMS suggestions
* CSV export/import; backup/restore
* Theming & accessibility polish

**Acknowledgments**

Built for **CS360 Project Three**: focuses on functional app launch readiness (auth, local persistence, runtime permissions) with clean code practices.