Product Requirements Document (PRD)

1) Product Overview

Product name (working): My Grocery List

One-liner: A simple, beautiful, offline-first grocery list app for South Indian households, built with Flutter, Dart, and SQLite, featuring quick add/edit, CSV import/export, and persistent state.

Primary users: South Indian families (multi-member households) managing weekly staples (rice, dals, spices, oils, vegetables, milk, batter) with occasional bulk buys and festival-specific items.

Platforms: Chrome(Via Web), Android (primary), iOS (secondary). Offline-first.

Key goals

- Fast capture and update of items with correct units.
- Bulk select (Select All), bulk delete, and quick edit.
- CSV import/export for sharing and backups.
- Persistent state saved to SQLite, auto-restored on app open.
- Clear, friendly UI with regional cues and local language support.

---

2) Personas & Use Cases

Persona A: Lakshmi (age 35) – manages weekly cooking; needs quick access to staples (idli rice, toor dal, coconut, jaggery). Wants to mark "Needed" for the weekend market.

Persona B: Arjun (age 24) – hostel returnee; quickly imports a CSV from mom; wants bulk delete of unwanted items and editing quantities.

Persona C: Ravi (age 42) – buys in bulk monthly; requires unit accuracy (kg, L, packets, dozen) and clear history via export.

Core use cases

1. Add items with unit + value and mark as Needed.
2. Multi-select via Needed checkbox → Delete or Edit.
3. Import CSV from WhatsApp/Downloads; merge with or replace current list.
4. Export current list to CSV and share.
5. Persist last saved state and restore on next launch.
6. Quick "Select All" to toggle Needed for the entire list.
7. Read in-app user manual via Top-right "i".

---

3) Scope & Features

3.1 Must-have (MVP)

- Item table with: **Sl No. (auto)**, **Item name**, **Qty (value + unit)**, **Needed (Y/N)**.
- Select All for Needed.
- Add / Edit / Delete (single & bulk via Needed selection).
- Save button persists to SQLite; last saved state shown by default.
- CSV Import (from local file) & CSV Export (to local/share sheet).
- In-app manual (icon "i").

3.2 Nice-to-have (Post-MVP)

- Preset catalog (South Indian staples) for one-tap add.
- Multi-language (English + Kannada/Tamil/Telugu/Malayalam).
- Smart suggestions (recent/frequent items).
- Shopping mode (filter by Needed only).
- Undo (snackbar) for delete.

---

4) Data Model & Storage (SQLite)

4.1 Entities

items

- id INTEGER PRIMARY KEY AUTOINCREMENT
- name TEXT NOT NULL
- qty_value REAL NULL (supports decimals)
- qty_unit TEXT NULL (enumerated string; see Units below)
- needed INTEGER NOT NULL DEFAULT 0 (0 = No, 1 = Yes)
- position INTEGER NOT NULL (for Sl No. ordering starting at 1; re-indexed on changes)
- created_at TEXT NOT NULL (ISO 8601)
- updated_at TEXT NOT NULL (ISO 8601)

app_meta (key-value for simple flags)

- key TEXT PRIMARY KEY
- value TEXT

4.2 Units of Measurement (Qty → dropdown)

Solids: kg, g, mg, piece, pieces, packet, packets, box, boxes, tin, bag, bags, bunch, bundle, dozen, tray.

Semi-solids/Powders: kg, g, mg, packet, box, jar, pouch, sachet.

Liquids: L, ml, bottle, bottles, can, tetra-pack.

Spices/Small measures (optional quick picks): tbsp, tsp, cup, pinch (stored as text; conversion not required).

Store qty_unit as string from a curated list. Validation ensures the dropdown only allows supported entries.

4.3 Constraints & Rules

- name trimmed, 1–60 chars; unique by (name, qty_unit) **optional** (configurable) to reduce duplicates.
- position defines Sl No.; must be contiguous starting at 1.
- On Delete, re-index position.
- On Import with **Replace**: truncate items then insert; **Merge**: upsert by name (and optionally qty_unit).

4.4 Migrations

- v1: create items, app_meta.
- v1.1: add position and backfill positions by current order.

---

5) CSV Specification

Filename: annadaata_grocery_YYYYMMDD_HHMM.csv

Columns (ordered):

1. Sl No (ignored on import; recomputed)
2. Item (string)
3. Qty Value (number or blank)
4. Qty Unit (string from supported units or free text; validated)
5. Needed (Y/N)

Export rules

- Export all current rows sorted by position.
- Needed serialized as Y or N.

Import rules

- Source: local file picker (Downloads/Documents). CSV with header row required.
- Validation: header must match; invalid units → map to closest or flag row for user choice.
- Options after parsing preview: **Merge** (default) or **Replace**.
- Error handling: show line number, reason; allow skip invalid rows.

Sample CSV

Sl No,Item,Qty Value,Qty Unit,Needed

1,Idli Rice,10,kg,Y

2,Toor Dal,2,kg,N

3,Coconut Oil,1,L,Y

4,Sambar Powder,200,g,Y

5,Curd,500,ml,N

---

6) User Flows

6.1 Add Item

Top bar → **Add** → Bottom sheet / modal form: Item, Qty Value, Qty Unit (dropdown), Needed (checkbox) → **Save**.
On save: insert row at end with position = last + 1. Snackbar: "Item added. Don't forget to Save to keep changes."

6.2 Edit Item(s)

Select by toggling **Needed** checkbox (acts as selection as per spec). Tap **Edit** icon (contextual) → If multiple selected, open bulk editor allowing: toggle Needed for all, or edit common fields; otherwise open single editor with full fields → **Apply** → pending state. **Save** persists.

6.3 Delete Item(s)

Select by **Needed** checkboxes → Tap **Delete** → Confirm dialog (show count). On confirm, remove from list (pending) and re-index position. **Save** persists. Optional undo snackbar (post-MVP).

6.4 Select All

Top bar checkbox → toggles Needed for every visible row. If some are selected, tapping sets all selected; tapping again clears all.

6.5 Import CSV

Top bar **Import** → File picker → Parse → Preview screen (table diff):

- Detected rows, invalid rows → list with reasons.
- Choice: **Merge** vs **Replace**.
- **Continue** → Apply changes to in-memory list; **Save** to persist.

6.6 Export CSV

Top bar **Export** → Generate CSV from current list → Share sheet.

6.7 Save

Top bar **Save** → Upsert all in-memory changes to SQLite. Persist position. Update timestamps. Snackbar: "Saved. Last saved: 12 Aug 2025, 7:20 pm."

6.8 App Relaunch

On launch, load from SQLite ordered by position. Display last saved timestamp (read from app_meta:last_saved_at).

6.9 Help (User Manual)

Top-right **i** → Scrollable guide with screenshots/illustrations, FAQ, CSV template download link, contact/feedback.

---

7) UX / UI Design

7.1 Visual Style

- **Theme:** Light by default; dark mode supported.
- **Palette:**
    - Primary: Teal/Green (fresh produce vibe)
    - Accents: Amber (spices), Deep Orange (festival)
    - Surface: Warm neutral whites.
- **Typography:** Inter or SF Pro; Large titles for page, medium weights for rows.
- **Iconography:** Material Symbols (outlined) for Add, Edit, Delete, Import, Export, Save, Help.
- **Micro-interactions:** Subtle elevation on cards, checkbox animation, pull-to-refresh (no network call, just re-order/refresh).

7.2 Layout (Main Screen)

- **Top App Bar:** Title, icons: Add • Import • Export • Save • Help (i). Right-aligned **Select All** checkbox.
- **List Body:** Reorderable ListView (optional), each row:
    - **Sl No.** (left, muted)
    - **Item** (primary label)
    - **Qty** (value + unit chip)
    - **Needed** checkbox (also selection)
    - **Inline Edit** icon (opens editor)
    - **Inline Delete** icon
- **FAB (optional):** Quick Add.

7.3 Forms

- Item: text; suggestions dropdown from recent/presets.
- Qty Value: numeric keyboard with decimals.

- Qty Unit: searchable dropdown with grouped units (Solids / Liquids / Small measures).
- Needed: checkbox.

7.4 Accessibility

- Minimum tap target 44x44.
- High-contrast mode support.
- Screen reader labels (semantics for buttons/checkboxes).
- Motion reduced when OS reduce-motion is on.

7.5 Localization (Post-MVP)

- Strings in ARB; support English + Kannada/Tamil/Telugu/Malayalam.
- Localized unit labels while storing canonical English tokens in DB.

---

8) Technical Design

8.1 Architecture

- **Pattern:** MVVM or BLoC (Riverpod recommended for simplicity and testability).
- **Layers:**
    - Data: sqflite, path_provider; repositories for items & meta.
    - Domain: models (freezed + json_serializable optional).
    - Presentation: Riverpod providers + widgets.

Key packages

- sqflite (SQLite)
- path_provider (db path)
- file_picker (CSV import)
- csv (parse/generate)
- share_plus (export/share)
- intl (date formatting)
- flutter_localizations + easy_localization (optional)
- riverpod or flutter_riverpod

8.2 SQLite Schema (DDL)

```
CREATE TABLE IF NOT EXISTS items (
  id INTEGER PRIMARY KEY AUTOINCREMENT,
  name TEXT NOT NULL,
```

```
  qty_value REAL,
  qty_unit TEXT,
  needed INTEGER NOT NULL DEFAULT 0,
  position INTEGER NOT NULL,
  created_at TEXT NOT NULL,
  updated_at TEXT NOT NULL
);

CREATE UNIQUE INDEX IF NOT EXISTS idx_items_position ON items(position);
CREATE INDEX IF NOT EXISTS idx_items_needed ON items(needed);

CREATE TABLE IF NOT EXISTS app_meta (
  key TEXT PRIMARY KEY,
  value TEXT
);
```

8.3 Repository Contracts (Dart)

```
abstract class ItemsRepo {
  Future<List<Item>> fetchAll();
  Future<void> saveAll(List<Item> items); // bulk upsert in a tx
  Future<void> clear();
}
```

Save algorithm (bulk):

1. Begin transaction.
2. Delete rows not present by id (or rebuild table for Replace).
3. Upsert each item with new position.
4. Update app_meta:last_saved_at.
5. Commit.

8.4 CSV Import Algorithm

1. Parse CSV → list of records.
2. Validate header; coerce Needed to {Y,N}.
3. Normalize units to canonical set.

4. Show preview + invalid rows.
5. Apply Merge/Replace to in-memory list.

8.5 Performance

- Lists of up to 2,000 items must scroll at 60fps.
- All DB ops within 50–150 ms for typical lists (<300 items).

8.6 Error Handling

- File I/O: show readable errors (permissions, malformed CSV).
- DB: rollback on failure; show retry.
- Validation: highlight offending fields with inline messages.

---

9) Acceptance Criteria (per Feature)

Add Item

- Given valid inputs, item appears at end with correct Sl No.
- On Save, DB contains new row; relaunch shows it.

Edit Item

- Selecting one/many via Needed then Edit opens editor.
- Changes reflected in list; Save persists.

Delete

- Selecting one/many then Delete removes them; Sl Nos re-index.
- Save persists deletions.

Select All

- Tapping Select All toggles Needed for all; visual state consistent.

Import CSV

- Valid CSV imports; preview shows row count and invalids.
- Replace clears previous items; Merge keeps others.
- After Save, relaunch shows imported data.

Export CSV

- Produces CSV with correct header and Y/N values, ordered by Sl No.
- Share sheet opens; file present in temporary location.

Persistence

- On relaunch, last saved list loads in < 300 ms on typical device.

Help

- Manual opens with sections and links.

---

10) Preset Catalog (Optional Seed Data)

- **Rice:** Sona Masoori, Idli Rice, Ponni Raw Rice, Basmati.
- **Dals/Legumes:** Toor, Urad (split/whole), Moong, Chana.
- **Flours/Batter:** Idli/Dosa Batter, Ragi Flour, Rice Flour, Besan.
- **Oils/Fats:** Groundnut Oil, Coconut Oil, Gingelly (Sesame) Oil, Ghee.
- **Spices/Mixes:** Sambar Powder, Rasam Powder, Garam Masala, Red Chilli, Turmeric, Mustard, Jeera, Pepper.
- **Condiments:** Tamarind, Jaggery, Salt.
- **Dairy:** Milk, Curd, Paneer, Butter.
- **Vegetables:** Drumstick, Brinjal, Tomato, Onion, Potato, Green Chilli, Curry Leaves, Coriander.
- **Others:** Coconut, Banana Leaf (festival), Incense, Camphor.

---

11) Security & Privacy

- No network required; data stored locally in app sandbox.
- File access permission prompted only during import/export.
- No PII beyond grocery entries.

---

12) Analytics (Optional)

- Local counters for adds/edits/deletes (no tracking). Exposed in Help for user curiosity.

---

13) Testing Strategy

- Unit tests: repository, CSV parser, unit normalizer, position re-indexer.
- Widget tests: list row interactions, add/edit/delete flows.
- Integration tests: import/export end-to-end.

Key edge cases

- Duplicate names with different units.
- Blank Qty Value with unit → allowed.
- Invalid Needed value during import.

- Non-UTF-8 CSV → show encoding error.
- Very long item names → ellipsis + wrap in editor.

---

14) Release Plan

- **v1.0 (MVP):** Core CRUD, CSV import/export, persistence, manual.
- **v1.1:** Presets, Undo, Shopping mode filter.
- **v1.2:** Localization + Dark mode polish.

---

15) In-App Manual Outline (Help → "i")

1. Getting Started
2. Adding Items
3. Editing & Deleting (bulk)
4. Using Select All
5. Importing from CSV (Merge vs Replace)
6. Exporting to CSV & Sharing
7. Units Guide
8. Troubleshooting & FAQs
9. Privacy
10. Contact / Feedback

---

16) Appendix

16.1 Example Item JSON (in-memory)


```
{
  "id": 12,
  "name": "Coconut Oil",
  "qty_value": 1.0,
  "qty_unit": "L",
  "needed": true,
  "position": 7,
  "created_at": "2025-08-12T13:45:22Z",
  "updated_at": "2025-08-12T13:46:03Z"
```

```
}
```

16.2 Pseudocode — Reindex After Delete

```
for (int i = 0; i < items.length; i++) {
  items[i] = items[i].copyWith(position: i + 1);
}
```

16.3 Pseudocode — Export CSV

```
final rows = [
  ['Sl No', 'Item', 'Qty Value', 'Qty Unit', 'Needed'],
  ...items.map((it) => [
    it.position,
    it.name,
    it.qtyValue?.toStringAsFixed(it.qtyValue == it.qtyValue?.roundToDouble() ? 0 : 2) ?? '',
    it.qtyUnit ?? '',
    it.needed ? 'Y' : 'N',
  ])
];
```

16.4 UI Empty States

- No items yet → “Your pantry is empty. Tap Add to begin.”
- Import parsed 0 valid rows → “No valid items found. Check the header format.”

End of PRD