**Bourbon Tracker – User Watchlist Implementation: Solution Blueprint**

**Core Tables & Relationships**

**1. user\_watchlist**

| **Column** | **Type** | **Notes** |
| --- | --- | --- |
| watch\_id | PK | Auto-increment |
| user\_id | Integer | FK → users table (authentication DB) |
| plu | Integer | Product identifier (PLU, unique per user) |
| custom\_name | Text | Optional, user-customized label for the item |
| notify\_email | Boolean | Notify this user by email for this PLU? |
| notify\_text | Boolean | Notify this user by text for this PLU? |
| active | Boolean | Is this watch currently enabled for this user? |
| added\_on | DateTime | When user started watching this item |

* Composite index: (user\_id, plu, active) for fast lookups.

**2. product\_change\_log**

| **Column** | **Type** | **Notes** |
| --- | --- | --- |
| change\_id | PK | Auto-increment |
| plu | Integer | Product affected |
| store\_id | Integer | Store affected (nullable for warehouse/global changes) |
| old\_qty | Integer | Previous inventory value |
| new\_qty | Integer | New inventory value |
| change\_type | Text | (e.g., “increase”, “decrease”, “restocked”) |
| check\_time | DateTime | When change was detected |

**3. user\_last\_notified *(optional, for precise notification tracking)***

| **Column** | **Type** | **Notes** |
| --- | --- | --- |
| user\_id | Integer | FK → users table |
| plu | Integer | FK → PLU |
| last\_notified | DateTime | When the user was last notified for this product |

**Supporting Tables (existing)**

* alcohol: Holds all state products (may have gaps)
* bourbons: Local list of bourbons for tracking (may have PLUs not in alcohol)
* current\_inventory/inventory\_history: Inventory data per store
* users (in authentication DB): User profile, status, contact info

**Data Flow / Process Overview**

1. **Watchlist Management**
   * Users add/remove/opt-out/in products on their list (user\_watchlist), by searching or entering a PLU (must be 5-digit integer).
   * Default monitored products are added to all new users’ watchlists by default (active=true), but can be toggled off.
2. **Inventory Update**
   * Scheduled script (every 15min) processes all changes, updating current\_inventory and appending detected changes to product\_change\_log.
3. **Notification Worker**
   * After each inventory update:
     + For each user:
       - Query their active watchlist PLUs.
       - Check for new entries in product\_change\_log since their user\_last\_notified timestamp (or since last script run).
       - Batch notifications per user, honoring their preferences (email/text).
       - Update user\_last\_notified as appropriate.
4. **Admin Features**
   * If multiple users add the same PLU (esp. one not in defaults), notify admin to consider for default tracking.
   * Admin UI can show popularity of “custom” tracked products.
5. **Cleanup & Integrity**
   * When a user is deleted or suspended, remove or inactivate all their watchlist rows (CASCADE or logic).
   * Ensure plu is always stored and matched as a canonical integer.
   * All notification/processing logic is keyed off PLU, never just name.

**Sample SQL Table Definitions**

sql

CopyEdit

CREATE TABLE user\_watchlist (

watch\_id INTEGER PRIMARY KEY AUTOINCREMENT,

user\_id INTEGER NOT NULL,

plu INTEGER NOT NULL,

custom\_name TEXT,

notify\_email BOOLEAN DEFAULT 1,

notify\_text BOOLEAN DEFAULT 0,

active BOOLEAN DEFAULT 1,

added\_on DATETIME DEFAULT CURRENT\_TIMESTAMP,

UNIQUE(user\_id, plu)

);

CREATE INDEX idx\_user\_watchlist\_user\_plu\_active

ON user\_watchlist (user\_id, plu, active);

CREATE TABLE product\_change\_log (

change\_id INTEGER PRIMARY KEY AUTOINCREMENT,

plu INTEGER NOT NULL,

store\_id INTEGER,

old\_qty INTEGER,

new\_qty INTEGER,

change\_type TEXT,

check\_time DATETIME NOT NULL

);

CREATE TABLE user\_last\_notified (

user\_id INTEGER NOT NULL,

plu INTEGER NOT NULL,

last\_notified DATETIME,

PRIMARY KEY (user\_id, plu)

);

**Best Practices / Lessons Encoded**

* Only one watchlist table, user\_id for multi-tenancy.
* Use PLU (5-digit int) as canonical key.
* Allow entries not in alcohol table, but validate format.
* Notification batching is per user after main change scan.
* Admin gets notified when multiple users track a new (custom) PLU.
* Bulk watchlist operations and category toggling for user convenience.
* Clean up watchlist data on user deletion.