💾 Software Requirements & Specifications – Luxury Watch Winder Cabinet

---

🎯 Purpose

This document outlines the software-level functional and non-functional requirements and specifications for the Luxury Watch Winder Cabinet, based on the system requirements (SRs) derived previously.

📘 Software Scope

This spec applies to:

- **Android Tablet Application** (UI, Voice Assistant, DB, USB Comm)
- **Raspberry Pi Software** (State Machine, Authentication, USB Protocol)
- Communication via **USB Serial using JSON**

📱 Android App Software Requirements

ID	Requirement	Type	Source SR
SWR01	App shall launch automatically on boot and hide Android system UI.	F	SR01, SR02
SWR02	App shall lock user access until Raspberry Pi sends authentication OK.	F	SR03, SR04
SWR03	App shall use Room database to store encrypted user watch library.	NF	SR06, SR21

SWR04	App shall fetch and sync the master watch DB from the backend securely.	F	SR08, SR20
SWR05	App shall allow user to select a watch from the master DB.	F	SR07
SWR06	App shall allow user to trigger add, delete, or edit of watch entries.	F	SR07
SWR07	App shall show a confirmation UI for watch loading with TTS/voice prompt.	F	SR11, SR12
SWR08	App shall send user library JSON to Raspberry Pi via USB upon update.	F	SR14
SWR09	App shall display the current state received from Raspberry Pi.	F	SR16
SWR10	App shall send a state change request (JSON) over USB when user initiates.	F	SR17
SWR11	App shall handle incoming messages for state status and errors.	F	SR19, SR23
SWR12	App shall provide voice command capability via LLM API (e.g., OpenAI).	F	SR10

SWR13	App shall resume from last known screen/state after boot.	F	SR25
SWR14	App shall encrypt all API keys and sensitive data using Android Keystore.	NF	SR21

🧠 Raspberry Pi Software Requirements

ID	Requirement	Type	Source SR
SWR15	Pi shall handle fingerprint authentication and notify app of success/fail.	F	SR03
SWR16	Pi shall maintain and control the master state machine of the system.	F	SR18
SWR17	Pi shall validate incoming state change requests and reject invalid ones.	F	SR18
SWR18	Pi shall respond to valid state requests with current state + status field.	F	SR19
SWR19	Pi shall accept updated user watch library in JSON format via USB.	F	SR14

SWR20	Pi shall log system errors and send alerts to Android App via USB.	F	SR23
SWR21	Pi shall persist the current state and resume it on reboot.	F	SR24

🔁 USB Communication Protocol (Spec)

Format (JSON Envelope):

```
{
  "command": "state_change_request",
  "payload": {
    "requested_state": "DELIVER_SLOT_2",
    "timestamp": "2025-07-18T15:33:00Z"
  }
}

```

Example Commands:

- `state_change_request`
- `user_library_update`
- `authentication_status`
- `current_state_response`
- `error_alert`

📈 Performance & Security Requirements

ID	Requirement	Type	Domain
SWR22	All API calls (LLM, DB) must use HTTPS with valid TLS cert.	NF	Android App
SWR23	All USB JSON messages must be acknowledged by	NF	Android + Pi

	receiver within 250 ms.		
SWR24	App must fail gracefully if USB communication is lost.	NF	Android App
SWR25	Pi must prevent any unauthorized state changes or spoofed messages.	NF	Raspberry Pi

---