

System Requirements – Luxury Watch Winder Cabinet

System Requirements – Luxury Watch Winder Cabinet

Purpose

This document lists the system-level functional and non-functional requirements derived from customer features for the AI-powered luxury watch winder cabinet. Each requirement is traceable to the corresponding customer feature.

Requirement Table

ID	Description	Type	Domain	Derived From
SR01	The Android app shall launch in fullscreen kiosk mode at device boot.	F	Android App	CF01
SR02	The app shall prevent access to the Android system UI.	F	Android App	CF01
SR03	The Raspberry Pi shall handle fingerprint authentication and notify the UI via USB.	F	Raspberry Pi / Android App	CF02
SR04	Authentication shall occur before	F	Android App	CF02

	accessing any core feature.			
SR05	The user shall be able to view a scrollable watch library.	F	Android App	CF03
SR06	Watch data shall be stored in a local encrypted Room database.	NF	Android App	CF03
SR07	The app shall allow adding watches by selecting from a synced DB.	F	Android App	CF04
SR08	The master watch DB shall be fetched securely from the backend.	F	Backend	CF04, CF11
SR09	The app shall allow adding watches using voice prompts.	F	Android App	CF05
SR10	The voice assistant shall use an online LLM API (e.g., OpenAI/Gemini).	F	Android App	CF05, CF13
SR11	Watch placement shall be confirmed via a touch or voice interaction.	F	Android App	CF06

SR12	The app shall prompt the user to place the watch and confirm manually.	F	Android App	CF06
SR13	The app shall allow requesting a specific watch via touch or voice.	F	Android App	CF07
SR14	The app shall export the user library in JSON format to the Raspberry Pi.	F	Android App	CF08, CF12
SR15	The cabinet shall serve the requested watch only after valid state change.	F	Raspberry Pi	CF08, CF10
SR16	The app shall display the current system/cabinet state.	F	Android App	CF09
SR17	The UI shall send state change requests over USB using JSON format.	F	Android App	CF10
SR18	The Raspberry Pi shall be the single authority to	F	Raspberry Pi	CF10

	approve state changes.			
SR19	The Pi shall respond with the current state and a status indicator.	F	Raspberry Pi	CF10, CF15
SR20	All API communication (watch DB, LLM) shall use HTTPS and token auth.	NF	Security	CF11, CF13
SR21	The app shall encrypt local user library data and API keys.	NF	Security	CF06, CF11
SR22	The app shall alert the user if any system error occurs.	F	Android App	CF14
SR23	The Raspberry Pi shall report system errors and state via USB.	F	Raspberry Pi	CF14
SR24	The system shall store last known state to support boot recovery.	F	Raspberry Pi	CF15
SR25	The app shall resume from last known UI state on reboot.	F	Android App	CF15

Traceability Matrix

Each requirement in this table is derived directly from a corresponding Customer Feature (CF), which is mapped to one or more Use Cases (UC) in the UI Design Spec.
