Functional Specification Document (FSD)

Project: ESP32-S3 Filament Manager UI

1. Purpose

This document defines the functional requirements, architecture, and operational behavior of the ESP32-S3 Filament Manager application. The system provides a touchscreen-based user interface for managing 3D printer filament profiles, including selection, visualization, and future expansion for RFID and network-based integration.

The FSD serves as a stable reference for implementation, debugging, and future enhancements.

---

2. System Overview

2.1 Hardware Platform

- **Controller:** ESP32-S3
- **Display:** Waveshare ESP32-S3-LCD-4.3C (RGB panel + touch)
- **Storage:** Internal Flash + LittleFS
- **Memory:** Internal RAM + External PSRAM
- **Peripherals (current/future):**
- Touch controller
- PN532 NFC/RFID
- Wi-Fi

2.2 Software Stack

- **Framework:** Arduino (ESP32 core)
- **Graphics:** LVGL 8.x
- **Display Driver:** LovyanGFX
- **Filesystem:** LittleFS
- **Data Format:** JSON
- **JSON Parsing:** ArduinoJson v7 (PSRAM-backed allocator)

---

3. High-Level Architecture

3.1 Module Breakdown

Module	Responsibility
`lvgl_display`	Display init, LVGL tick, splash screen

Module	Responsibility
`ui_manager`	Screen transitions, event routing
`screen_library`	Filament library grid UI
`screen_filament_select`	Active filament selection
`screen_about`	System info / credits
`filament_db`	Load, parse, cache filament database
`network_manager`	Wi-Fi connectivity
`rfid_driver`	NFC tag reading (future)
`system_state`	Global state machine
`config_manager`	Persistent configuration

---

4. Application Startup Flow

1. Boot ESP32-S3
2. Initialize Serial logging
3. Initialize display + LVGL
4. Show splash screen
5. Initialize subsystems:
6. Network
7. RFID
8. Filament database
9. Display subsystem status on splash screen
10. Transition to main UI
11. Initialize UI screens

---

5. Filament Database

5.1 Storage

- Location: `/material_database.json`
- Filesystem: LittleFS
- Size: ~180 KB

5.2 JSON Structure (Expected)

```
{
  "result": {
    "list": [
      {
        "base": {
          "id": "...",
          "brand": "...",
          "name": "...",
          "meterialType": "...",
          "colors": ["#RRGGBB"]
        },
        "kvParam": {
          "nozzle_temperature": 200,
          "hot_plate_temp": 60
        }
      }
    ]
  }
}
```

5.3 FilamentDB Responsibilities

- Mount LittleFS
- Load JSON into PSRAM-backed `JsonDocument`
- Parse and validate structure
- Populate in-memory cache (`std::vector<FilamentProfile>`)
- Expose read-only accessors

5.4 Memory Strategy

- JSON document allocated in PSRAM via custom allocator
- Parsed data copied into compact internal structures
- JSON document discarded after load

---

6. Filament Library Screen

6.1 Purpose

Displays all available filament profiles in a grid layout optimized for a 4.3" touchscreen.

6.2 Layout

- Title bar (top)
- Back button (top-left)
- Scrollable grid (bottom)

6.3 Grid Rules

- Fixed column count: **4 columns**
- Rows computed dynamically:

```
rows = ceil(filamentCount / 4)
```

- Max rows capped to prevent excessive memory usage

6.4 Grid Cell Contents

Each filament cell displays: - Color swatch - Brand (top) - Filament name (bottom)

6.5 Behavior

- Grid auto-resizes based on filament count
- Touching a cell selects filament (future behavior hook)

---

7. UI Manager

7.1 Responsibilities

- Screen creation lifecycle
- Screen transitions
- Centralized LVGL event handling

7.2 Navigation

- Splash → Main
- Main → Library
- Library → Select / Back

---

8. Error Handling & Diagnostics

8.1 Filesystem Errors

- Mount failure → splash error
- Missing JSON → splash error

8.2 JSON Errors

- Parse failure → abort load
- Missing keys → logged warning, safe defaults

8.3 Memory Safety

- Heap usage logged during DB load
- PSRAM allocator used for large JSON
- Avoid long-lived JSON references

---

9. Constraints

- Internal RAM is limited (~320 KB)
- PSRAM availability varies by board
- LVGL object creation must occur after LVGL init
- No blocking operations in `loop()`

---

10. Future Extensions

- RFID filament auto-detection
- Filament usage tracking
- Network sync with printer/host
- Material filtering and search
- Per-filament tuning UI

---

11. Non-Goals (Current Phase)

- Printer motion control
- G-code generation
- Real-time printer telemetry

---

12. Status

Document Status: Draft (v1.0)

This FSD reflects the current architecture and implementation state and is intended to stabilize development going forward.