Product Requirements Document (PRD)

Project Name: FaraLite Access Control Software

Version: 1.1 (Updated based on Mockup Discussions)

Document Status: Draft

Prepared by: [User]

Last Updated: 2025-05-15

---

Table of Contents

1. Introduction and Goals

2. Target Audience

3. Platform, Technology Stack, and Language Support

3.1. Platform

3.2. Technology Stack

3.3. Language Support

4. Features and Functional Requirements

4.1. Device Connectivity

4.2. Main Dashboard – In/Out Logs View

4.3. User Management Page (Cardholders)

4.4. Camera Settings Page

4.5. Hardware Settings Page

4.6. System Settings Page

4.7. Logs Synchronization and Offline Support

4.8. Access Permissions Logic

4.9. Reports Page

5. Database Design

6. Setup and Distribution

7. Non-Functional Requirements

7.1. Performance

7.2. Reliability

7.3. Security

7.4. Usability

7.5. Localization

7.6. Maintainability

8. Future Features (Optional Enhancements for Versions > 1.1)

9. Limitations (for Version 1.1)

10. Document Revision History

---

1. Introduction and Goals

The primary goal is to create FaraLite, a simple, standalone, Windows-compatible access control software application. FaraLite will interface with hardware access control devices via WebSocket using a predefined JSON protocol. Key characteristics include:

No requirement for external services or heavy dependencies.

Installation via a lightweight setup executable (.exe).

Integration with IP cameras for live video feeds.

Management of user access based on card numbers.

Support for Administrator and Operator user roles within the software itself.

---

2. Target Audience

Residential and commercial property managers

Security personnel

IT staff for small-to-medium-sized buildings

---

3. Platform, Technology Stack, and Language Support

3.1. Platform:

Operating System: Windows 10 and above (64-bit recommended).

3.2. Technology Stack:

Communication: WebSocket client, RTSP client (for IP cameras).

Database: Embedded local SQLite (no separate installation required).

Preferred Development Language: Python (to facilitate potential future extension to a web application).

3.3. Language Support:

Supported Languages: Persian and English (switchable within the application).

Default Language: Auto-detected from Windows locale settings upon first run.

Language Switching: User-selectable via a dropdown menu in the System Settings page.

Localization Mechanism: UI strings must be maintained in external localization files (e.g., .resx, JSON, or similar format).

---

4. Features and Functional Requirements

4.1. Device Connectivity

4.1.1. Protocol Adherence: Must use the predefined WebSocket protocol (as per provided JSON specification).

4.1.2. Device Search:

Provide a button to scan the Local Area Network (LAN) for supported devices.

Limit: Support discovery of a maximum of 2 devices.

Display: Show search results in a list format, including IP addresses, MAC addresses, and device names (if available).

4.1.3. Port Assignment (Hardware Settings Page):

Allow users to assign IN/OUT roles to ports on one or both connected devices.

Example Configurations:

Single Device: Allow tagging two distinct ports on the device as IN and OUT.

Dual Devices: Allow tagging one device as IN and the other as OUT.

4.1.4. Connection Status Indicator:

Display real-time connection status on the main window.

Visual Cue: Use a color indicator (e.g., Green for Online, Red for Offline).

4.2. Main Dashboard – In/Out Logs View

4.2.1. Default View: This page shall be the default landing page after application startup and successful login.

4.2.2. Log Display:

Order: Show log entries sorted in descending order (most recent first).

Columns: Timestamp, User Name, User ID, Card Number, Device Name/Identifier, Direction (In/Out), Status (Accepted/Rejected).

Optional Column: Photo Thumbnail of the user.

4.2.3. Live Camera Feeds:

Display two small, configurable live video feeds (e.g., for designated indoor and outdoor cameras).

4.3. User Management Page (Cardholders)

4.3.1. CRUD Operations: Support Add, Edit, and Delete operations for normal users (cardholders).

4.3.2. Auto-Refresh: The user list should automatically update after any Add, Edit, or Delete operation.

4.3.3. User Fields:

First Name

Last Name

User ID (must be a unique serial number)

Unit Number

Car Plate Number

Card Number (8-digit)

Permission Level (details in section 4.8)

Photo:

Option 1: Auto-capture from a connected webcam.

Option 2: Upload from an image file.

4.3.4. Card Assignment: Functionality to assign tags/cards to the user.

4.3.5. Device Synchronization:

Download user data from connected hardware device(s).

Upload user data to connected hardware device(s).

4.4. Camera Settings Page

4.4.1. IP Camera Search:

Provide a button to scan the LAN for available IP cameras (supporting RTSP).

4.4.2. Discovered Cameras List:

Display a list of found cameras, including their IP address and RTSP URL (if discoverable).

4.4.3. Camera Assignment:

Provide dropdown menus to assign each discovered/configured camera as "Indoor," "Outdoor," or "None" for display on the main dashboard.

4.5. Hardware Settings Page

4.5.1. Manual Device Configuration: Allow manual entry of device IP addresses (as an alternative if device search fails).

4.5.2. Port Tagging (Manual Config): Allow tagging of port numbers as IN/OUT for manually configured devices.

4.5.3. Configuration Persistence: Save and persist device configurations locally (e.g., in JSON format or SQLite database).

4.5.4. Time Synchronization: Provide a button to synchronize the connected hardware devices' clocks with the system clock of the computer running FaraLite.

4.6. System Settings Page

4.6.1. Software User Account Management (Admin/Operator):

Functionality to Add, Edit, and Delete Administrator and Operator accounts for the FaraLite software.

Account Fields: Username, Password, Role (Administrator/Operator).

4.6.2. Language Configuration:

Provide a dropdown menu to switch the application interface language between Persian and English (as detailed in 3.3).

4.6.3. Optional Password Protection for Settings:

Provide a checkbox to enable/disable password protection for accessing critical settings pages.

If enabled, provide fields to set and confirm the password.

4.6.4. Bulk User Data Management:

"Upload All Users to Devices" button.

"Download All Users from Devices" button.

4.7. Logs Synchronization and Offline Support

4.7.1. Local Log Storage: The software must store all access logs pulled from devices in the local SQLite database.

4.7.2. Offline Log Queuing and Synchronization:

If the connection to a hardware device is lost, the software should recognize that the device may buffer logs.

Upon reconnection, the software must automatically fetch any queued logs from the device.

4.7.3. Timestamp Handling: Ensure accurate timestamping, with an option to align device logs with the system clock if discrepancies arise.

4.8. Access Permissions Logic

The following permission levels can be assigned to users (cardholders):

4.8.1. "Open": User is granted access at any time through controlled hardware.

4.8.2. "Close": User is explicitly denied access through controlled hardware.

4.8.3. "Restricted": Access is granted based on predefined working hours or time-based schedules (details of schedule definition TBD, initially may be a single site-wide schedule).

4.9. Reports Page

4.9.1. Log Search & Filtering Capabilities:

Search Fields: User Name, User ID, Car Plate Number, Card Number, Date Range, Time Range.

Filter Options: Entry Type (In/Out), Device, Status (Accepted/Rejected), User, Card Number.

Controls: "Apply Filters" and "Clear Filters" buttons.

4.9.2. Report Display: Log results matching search/filter criteria to be displayed in a table format.

4.9.3. Report Generation Options:

Checkbox to include/exclude User Photo Thumbnails in the report.

Button to "Generate Report" (export to CSV format).

Button to "Export to Excel" (export to .xlsx format).

---

5. Database Design

A local, file-based SQLite database will be used.

Schema to include tables for:

Cardholder Users (Name, User ID, Card No., Permissions, Photo path, etc.)

Access Logs (Timestamp, User Info, Device, Direction, Status, etc.)

Device Configurations (IP, MAC, Name, IN/OUT port assignments)

IP Camera Configurations (IP, RTSP URL, Assigned Role)

Software User Accounts (Admin/Operator credentials)

(Detailed schema to be defined during the technical design and implementation phase.)

---

6. Setup and Distribution

Distribution Method: A simple Windows installer package (e.g., .exe or .msi).

Dependencies: No external database or third-party service installations required post-setup.

Included Components:

Embedded SQLite engine