**NextGen Supply System – Configuration Settings Module**

**Introduction**

The Configuration Settings Module of the NextGen Supply System allows administrators, technicians, and operators to securely log in and configure device and machine settings, ensuring smooth and reliable system operation.

It integrates user authentication, device setup, RFID mapping, and machine assignments into a structured workflow.

**This module is designed for:**

Admins & Operators – securely log in and manage configurations.

Technicians – configure device communication and hardware parameters.

System Integrators – ensure correct mapping of RFID readers and machines to controllers.

**Pages & Workflows**

**1. Login Page**

**Inputs**

* Company Code
* Username & Password (or Face Recognition)
* PS. In real app the face recognition is not included in configuration settings
* It only use default user and password

**Process**

* Authenticate user using JWT tokens.
* Detect Device ID automatically or allow manual entry.

**Output**

* Redirects to configuration pages.

**2. Device Settings Page**

**Fields**

* Company Code (auto-filled from login)
* Device ID
* COM Port
* Baud Rate

**Purpose**

* Establishes serial communication between the NextGen system and physical hardware.

**Validations**

* Ensures correct COM Port and Baud Rate to prevent failures.

**3. RFID Settings Page**

**Purpose**Configure and register RFID reader/writer devices.

**Parameters**

* RFID Reader ID
* Port Mapping
* Communication Protocol

**Features**

* Assign multiple RFID readers to devices.
* Support for various protocols and configurations.

**4. Machine Settings Page**

**Purpose**  
Configure machine parameters and assign them to controllers.

**Table View (per row)**

* Machine Type
* Machine Tag
* Machine Name
* Controller ID (dropdown)

**Features**

* Scrollable table for large datasets.
* Dynamic expansion to support future machine types.

**5. Data Flow**

* Company Code → passed from Login → Device Settings.
* Device ID + COM Port + Baud Rate → passed to RFID & Machine Settings.
* Machine Configurations

**6. Security**

* JWT Authentication → required for all actions.
* Role-Based Restrictions → only admins can modify configs.
* Face Recognition Login → optional via Python face\_recognition + FastAPI service.

**7. Deployment & Integration**

* Backend → ASP.NET Core 8.0 Web API + JWT Authentication
* Frontend → Flutter multi-page workflow for configuration
* Face Embedder Service → Python + FastAPI (face\_recognition, NumPy, Pillow)
* Database → SQL Server with Entity Framework Core (EF Core) migrations

**Development Environment**

* IDE: Visual Studio 2022 (C# backend), Visual Studio Code / Android Studio (Flutter frontend), PyCharm/VSCode (Python microservice)

**Languages:**

* C# (ASP.NET Core, EF Core)
* Dart (Flutter)
* Python (FastAPI, face recognition)
* SQL (SQL Server)

**8. Example User Workflow**

1. User logs in with Company Code + Credentials (or via Face Recognition).
2. Configures Device Settings (Device ID, COM Port, Baud Rate).
3. Sets up RFID Settings (assign readers to devices).
4. Completes Machine Settings (define machine type, tag, controller mapping).
5. Saves configuration → system ready for operation.