Pylae – Gate & Reception Control (v1 Specification)

1. Project Overview

Pylae is a Windows desktop application for **visitor / member management and access logging** at gates and reception desks.

Key goals:

- Run as a **single WinForms .NET 10 app** usable by both **Admins** and **Users (gate/reception)**.
- Use **encrypted SQLite** locally for all data.
- Support **multiple offices/sites**, including **built-in HTTP-based sync** between app instances on a LAN:
  - The HTTP sync feature is **implemented in v1**, but **disabled by default**.
  - It is **switchable per instance** via an admin setting (`NetworkEnabled`).
- Produce **badges** for members with:
  - Printed **MemberNumber** (digits).
  - A **QR code** that encodes the same MemberNumber.
- Log **entries/exits** with long-term visit history (5+ years) while keeping the app and sync operations performant.
- Be **Greek-first** in UI and exports, but structurally English in DB and APIs.
- Show a **localizable subtitle** (default: “Gate & Reception Control”) in key views (e.g. login, main window, About dialog).
- Theming:
  - Pylae must follow the user’s Windows light/dark theme automatically.
  - Default color mode: system theme (light/dark, including high contrast when enabled).
  - The app must not hardcode light-theme-only colors; use system colors or theme-aware styles.

-Custom controls must opt into theming where appropriate.

---

2. Roles, Permissions & Account Management

2.1 Roles

- **Admin**
  - Full CRUD on:
    - Members
    - Users
    - Offices
    - MemberTypes
  - Manage Settings (SiteCode, network, logging, idle timeout, language, badge validity, etc.).
  - Import/export encrypted DBs and photos (backup, sync, copy).
  - View and export Visits and AuditLog, from local or remote instances.
  - Reset other users' passwords and QuickCodes.
- **User** (gate/reception)
  - Perform **check-in / check-out** operations.
  - Search Members and Visits.
  - View member details and photos.
  - Add / edit **Notes** on Visits.
  - Change their **own** password and QuickCode (except for special shared account).
  - Cannot create or edit Members, Users, Offices, MemberTypes, or Settings.

2.2 Special Accounts

- **Protected System Admin Account**
  - One built-in admin account is **protected** and:
    - Cannot be deleted.
    - Cannot be deactivated (`IsActive` cannot be set to 0).
    - Cannot have its `Role` changed from `admin` .

    - This guarantees there is always at least one admin.
- **At-Least-One-Admin Rule**
    - Any operation that would result in **zero active admin accounts** is **blocked**:
        - Deleting or deactivating the last admin.
- **Shared Gate Account**
    - Optional `User` role account flagged as `IsShared = 1` :
        - Intended for shared use at busy gates.
        - Its password and QuickCode are **only manageable by Admins**.
        - Regular users cannot change its credentials.

2.3 Authentication & Login

- Every user logs in with:
    - **Username + Password** (always allowed).
    - For `Role = user` and `IsShared = 0` , an optional **6-digit QuickCode** can be enabled:
        - Allows fast unlock at a shared workstation.
- For **Admins**:
    - Only full password login is allowed (no QuickCode login).

2.4 Quick Login Management

- **Admins can:**
    - Set, reset, or clear QuickCodes for any `user` -role account.
    - Set/reset QuickCode and password for the **shared gate account**.
- **Normal users can:**
    - Change their own password.
    - Set/change/clear their own QuickCode (if allowed by policy).
- QuickCodes are:
    - Exactly 6 digits.
    - Stored hashed (not plaintext) along with normal passwords.

2.5 Forgot Password / Credential Recovery

- There is **no email-based or internet-based** password reset (target is LAN/standalone).
- If a user forgets their password:

    - They contact an **Admin**, who can:
        - Reset the password to a temporary value.
        - Optionally force “must change on next login” behavior (implementation detail).
- If a user forgets their QuickCode:
    - They log in with password.
    - Either:
        - Self-reset QuickCode (if allowed).
        - Or ask Admin to reset it.

2.6 Idle Lock

- `IdleTimeoutMinutes` (Settings):
    - `0` → disabled.
    - `>0` → after N minutes of inactivity, app auto-locks.
- Unlock behavior:
    - Show “Session locked” dialog:
        - Re-enter credentials for current user:
            - Password or QuickCode (for user role with QuickCode).
            - Admin must use password.
        - Or “Switch user” → go back to full login form.

---

3. Architecture

3.1 Platform

- **.NET 10** (target).
- **WinForms** with **CommunityToolkit.Mvvm**:
    - Forms are views.
    - ViewModels hold state & logic.
    - Minimal logic in form code-behind.

3.2 Databases

Two separate encrypted SQLite databases per installation/site:

- `master.db`
  - Members, Users, Offices, MemberTypes, Settings, AuditLog.
- `visits.db`
  - Visits only (append-only log).

3.3 Data Location (Per Site)

Default per-machine layout:

- Binaries:
  - `C:\Program Files\Pylae\`
- Data (per site):
  - `C:\ProgramData\Pylae\{siteCode}\`
    - `Data\master.db`
    - `Data\visits.db`
    - `Photos\*.jpg`
    - `Logs\*.log`
    - `Config\` (optional extra files)

`siteCode` is lower-case ASCII (e.g. `hq` , `gate_a`).

3.4 Encryption

- Use **Microsoft.Data.Sqlite.Core** + **SQLitePCLRaw.bundle_e_sqlite3mc** (SQLite3 Multiple Ciphers).
- Connection strings always include a `Password` .
- An organization-level **DB encryption password**:
  - Set on first run.
  - Must be used consistently on all instances sharing data.
- Implementation must **not** store the password in plaintext.

---

4. Networking & Sync

4.1 Network Enablement

Per instance:

- `NetworkEnabled` (0/1)
- `NetworkPort` (int, e.g. 8080)
- `NetworkApiKey` (string)
- `SiteCode` , `SiteDisplayName`

Behavior:

- `NetworkEnabled = 0` (default):
  - No HTTP server is started.
- `NetworkEnabled = 1` :
  - Start embedded HTTP server (Kestrel/minimal API) on `NetworkPort` .

4.2 HTTP API (LAN-only, v1)

Core endpoints (conceptual):

- `GET /api/sync/info`
  - Returns site metadata and basic stats.
- `GET /api/sync/visits/full`
  - Streams full encrypted `visits.db` .
- `GET /api/sync/visits?from=YYYY-MM-DD&to=YYYY-MM-DD`
  - Returns visits for the given date range (JSON or small DB).
- `POST /api/sync/master`
  - Accepts a package with:
    - Encrypted `master.db` .
    - Optional photo bundle (zip).
  - Replaces/merges into local `master.db` and `Photos/` .

Authentication:

- All HTTP calls require header `X-Api-Key: {NetworkApiKey}` .
- Intended for **LAN** only, not internet exposure by default.

4.3 Admin-Driven Connections

- Any Admin can:
  - Connect from one instance to another reachable instance.
  - Use “Remote Sites” UI to configure:
    - Host, Port, ApiKey.
  - Pull visits or push master data and photos.
  - Optionally inspect remote Settings (for diagnostics).

---

5. Localization

5.1 Rules

- **Structure (DB/API)**: English.
  - Table/column names.
  - Codes like `entry` , `exit` , `admin` , `user` .
- **Presentation**: localized (Greek for v1).
  - Form labels, messages, menus.
  - Badge labels.
  - Excel column headers.
  - Office names and MemberType display names.
  - App subtitle.

5.2 Language Settings

- `PrimaryLanguage` (Settings), e.g. `'el-GR'` .
- On startup:
  - Set `CurrentCulture` and `CurrentUICulture` .
- UI text comes from resx files.

Subtitle localization:

- Resource key, e.g. `App_Subtitle` :
  - `Strings.resx` (default): `"Gate & Reception Control"` .

  - `Strings.el-GR.resx` : `"Έλεγχος Πυλών & Υποδοχής"` .

Forms should never hardcode the subtitle string; they should always read it from resources.

5.3 Site Names

- `SiteCode` – ASCII, lower-case.
- `SiteDisplayName` – localized.

---

6. Data Model – master.db

6.1 Members

Represents visitors/members who receive badges and can appear in visits.

- `Id` – `TEXT` (GUID, PK).
- `MemberNumber` – `INTEGER` , NOT NULL.
  - Reusable numeric badge ID (printed and encoded in QR).
  - Uniqueness enforced via app logic among active members.
- `FirstName` – `TEXT` , NOT NULL.
- `LastName` – `TEXT` , NOT NULL.
- `BusinessRank` – `TEXT` (rank/grade/position).
- `OfficeId` – `INTEGER` (FK → Offices.Id).
- `IsPermanentStaff` – `INTEGER` (0/1).
- `MemberTypeId` – `INTEGER` (FK → MemberTypes.Id).
- `PersonalIdNumber` – `TEXT` (government ID).
- `BusinessIdNumber` – `TEXT` (internal ID, distinct from MemberNumber).
- `PhotoFileName` – `TEXT` (e.g. `{Id}.jpg`).
- `BadgeIssueDate` – `TEXT` (`YYYY-MM-DD`).
- `BadgeExpiryDate` – `TEXT` (`YYYY-MM-DD` , nullable; computed from Settings/BadgeValidityMonths when issuing).
- `DateOfBirth` – `TEXT` , optional.
- `Phone` – `TEXT` (not on badge).

- `Email` – `TEXT` (not on badge).
- `Notes` – `TEXT` (not on badge).
- `IsActive` – `INTEGER` (0/1).
- `CreatedAtUtc` – `TEXT` .
- `UpdatedAtUtc` – `TEXT` .

Behavior on deletion:

- Member row deleted or marked `IsActive = 0` .
- Photo file removed from `Photos/` .
- Visits remain untouched (using snapshot data).

6.2 MemberTypes

Categories like STAFF, VISITOR, THIRD_PARTY_SUPPLIER.

- `Id` – `INTEGER` PK AUTOINCREMENT.
- `Code` – `TEXT` , UNIQUE (e.g. `staff` , `visitor`).
- `DisplayName` – `TEXT` (localized).
- `Description` – `TEXT` .
- `IsActive` – `INTEGER` .
- `DisplayOrder` – `INTEGER` .
- `CreatedAtUtc` – `TEXT` .
- `UpdatedAtUtc` – `TEXT` .

6.3 Offices

Issuing offices.

- `Id` – `INTEGER` PK AUTOINCREMENT.
- `Code` – `TEXT` , UNIQUE, NOT NULL.
- `Name` – `TEXT` , NOT NULL (localized).
- `Phone` – `TEXT` .
- `HeadFullName` – `TEXT` .
- `HeadBusinessTitle` – `TEXT` .
- `HeadBusinessRank` – `TEXT` .

- `Notes` – `TEXT` (printed on badge).
- `IsActive` – `INTEGER` .
- `DisplayOrder` – `INTEGER` .
- `CreatedAtUtc` – `TEXT` .
- `UpdatedAtUtc` – `TEXT` .

6.4 Users

Application accounts, including protected admin and shared gate account.

- `Id` – `INTEGER` PK AUTOINCREMENT.
- `Username` – `TEXT` , UNIQUE, NOT NULL.
- `FirstName` – `TEXT` , NOT NULL.
- `LastName` – `TEXT` , NOT NULL.
- `PasswordHash` – `TEXT` .
- `PasswordSalt` – `TEXT` (or combined).
- `Role` – `TEXT` (`'admin'` / `'user'`).
- `QuickCodeHash` – `TEXT` , nullable (hashed 6-digit code).
- `IsShared` – `INTEGER` (0/1).
  - `1` marks the shared gate account; only admins can change its credentials.
- `IsSystem` – `INTEGER` (0/1).
  - `1` marks the protected built-in admin:
    - Cannot be deleted or deactivated.
    - Role cannot be changed from `admin` .
- `IsActive` – `INTEGER` (0/1).
- `CreatedAtUtc` – `TEXT` .
- `LastLoginAtUtc` – `TEXT` or `NULL` .

Account management rules:

- Attempts to delete or deactivate a `IsSystem = 1` user are forbidden.
- Attempts to delete/deactivate any admin that would leave **zero active admins** are forbidden.

6.5 Settings

Key/value pairs.

- `Key` – `TEXT` PK.
- `Value` – `TEXT` .
- `UpdatedAtUtc` – `TEXT` .

Key groups:

- Identity & Locale
  - `SiteCode`
  - `SiteDisplayName`
  - `PrimaryLanguage`
- Network
  - `NetworkEnabled` (0/1)
  - `NetworkPort`
  - `NetworkApiKey`
- Security & Session
  - `IdleTimeoutMinutes`
- Logging
  - `LogLevel`
  - `LogFileMaxSizeMB`
  - `LogRetentionDays`
  - `HealthLoggingEnabled`
- Badge Validity
  - `BadgeValidityMonths` :
    - `-1` → badge expiry **disabled**.
    - `>0` → number of months from `BadgeIssueDate` for expiry.
  - `BadgeExpiryWarningDays` :
    - Number of days before expiry to start showing warnings (e.g. `30`).
- Organization Identity
  - `OrgBusinessTitle` (for badge).
  - `OrgBusinessTel` (for badge).

6.6 AuditLog

Structured audit trail.

- `Id` – `INTEGER` PK AUTOINCREMENT.
- `TimestampUtc` – `TEXT`.
- `SiteCode` – `TEXT`.
- `UserId` – `INTEGER` (nullable).
- `Username` – `TEXT`.
- `ActionType` – `TEXT`.
- `TargetType` – `TEXT`.
- `TargetId` – `TEXT`.
- `DetailsJson` – `TEXT`.

Covers:

- Logins/logouts.
- Member/Office/MemberType/User CRUD.
- Settings changes (including network, badge validity).
- DB import/export operations.
- HTTP sync operations.
- Visit notes creation/updates.

---

7. Data Model – visits.db

7.1 Visits

Immutable log with snapshot of important member info.

- `Id` – `INTEGER` PK AUTOINCREMENT.
- `VisitGuid` – `TEXT` (optional).
- `MemberId` – `TEXT` (GUID).
- **Member snapshots:**
  - `MemberNumber` – `INTEGER`.
  - `MemberFirstName` – `TEXT`.

  - `MemberLastName` – `TEXT`.
  - `MemberBusinessRank` – `TEXT`.
  - `MemberOfficeName` – `TEXT`.
  - `MemberIsPermanentStaff` – `INTEGER` (0/1).
  - `MemberTypeCode` – `TEXT`.
  - `MemberTypeName` – `TEXT`.
  - `MemberPersonalIdNumber` – `TEXT`.
  - `MemberBusinessIdNumber` – `TEXT`.
- **Visit data:**
  - `Direction` – `TEXT` (`'entry'` / `'exit'`).
  - `TimestampUtc` – `TEXT`.
  - `TimestampLocal` – `TEXT`.
  - `Method` – `TEXT` (`'scan'` / `'manual'`).
  - `SiteCode` – `TEXT`.
  - `UserId` – `INTEGER`.
  - `Username` – `TEXT`.
  - `UserDisplayName` – `TEXT`.
  - `WorkstationId` – `TEXT`.
  - `Notes` – `TEXT`.

Indexes:

- `idx_Visits_Timestamp`.
- `idx_Visits_MemberNumber_Timestamp`.
- `idx_Visits_SiteCode_Timestamp`.

---

8. Check-In / Check-Out Flow & Expiry Warnings

8.1 Gate Screen & Mode Selection

- Mode toggle: **Entry** / **Exit** (always explicit).
- Input field for `MemberNumber` (scanner or manual).

- Optional field for `Notes` (e.g. “Meeting with …”).

8.2 Flow

1. User selects mode (Entry/Exit).
2. Enter/scan MemberNumber.
3. System finds active `Member` with that number.
4. If not found:
   - Show error; no visit recorded.
5. If found:
   - Load Member, MemberType, Office.
   - Check badge expiry (see below).
   - Insert `Visits` record with snapshot + visit data.
   - Show member card with latest info and photo.
   - Show any expiry warnings if applicable.

8.3 Badge Expiry Warnings

If badge validity is enabled (`BadgeValidityMonths > 0`) and `BadgeIssueDate` is set:

- Compute `BadgeExpiryDate` if not already present:
  - `IssueDate + BadgeValidityMonths` .
- On each check-in/check-out:
  - If `BadgeExpiryDate` is in the **past**:
    - Show a **strong warning** (e.g. red banner):
      - “Badge expired on {date}”.
    - Still allow the visit to be logged by default (v1 behavior).
    - Log this event in AuditLog (e.g. `ActionType = 'UseExpiredBadge'`).
  - If `BadgeExpiryDate` is within `BadgeExpiryWarningDays` :
    - Show a **soft warning** (e.g. yellow banner):
      - “Badge will expire on {date}`”.

---

9. Badge Design

9.1 Elements

- **Fixed text**:
  - Titles/instructions/tips (localized from resx).
- **Org/office data** (from Settings + Offices):
  - `OrgBusinessTitle` , `OrgBusinessTel` .
  - `Offices.Name` .
  - `HeadFullName` , `HeadBusinessTitle` , `HeadBusinessRank` .
  - `Offices.Notes` .
- **Member data** (from Members & MemberTypes & Offices):
  - `MemberNumber` (digits).
  - QR code encoding `MemberNumber` .
  - Name (FirstName + LastName).
  - BusinessRank.
  - Office name.
  - Permanent/Temp status.
  - Member type (MemberTypeName).
  - PersonalIdNumber.
  - BusinessIdNumber.
  - Photo.
- **Dynamic dates**:
  - `BadgeIssueDate` .
  - `BadgeExpiryDate` (if validity enabled).

9.2 Rendering

- Use an open source PDF library (e.g. QuestPDF with compatible license).
- Fixed layout in v1 (no badge editor).
- Localized labels via resx.

10. Logging & Auditing

10.1 App/System & Health Logs

- Library: **Serilog** with rolling file sink.
- Location: `C:\ProgramData\Pylae\{siteCode}\Logs\` .
- Config from Settings:
  - `LogLevel` , `LogFileMaxSizeMB` , `LogRetentionDays` , `HealthLoggingEnabled` .
- Automatic cleanup:
  - Delete log files older than `LogRetentionDays` on startup/interval.

10.2 AuditLog

- Records sensitive operations (see §6.6).
- Optionally mirrored to log files with `AUDIT:` prefix.
- Admin UI for filtering and export to Excel/JSON.

---

11. Exports & Backups

11.1 Human-Readable Exports (Admin)

- Excel via ClosedXML.
- JSON for archiving/integration.
- Domains:
  - Members.
  - Visits.
  - AuditLog (optional).

11.2 Encrypted DB Backup

- Zip that includes:
  - Encrypted `master.db` and `visits.db` .

  - Photos (optional).
- Admin-triggered via UI.

11.3 HTTP Sync

- Uses HTTP API (LAN only).
- Moves encrypted DBs or JSON.
- Controlled by `NetworkEnabled` toggle.

---

12. Scalability & Retention

- Target: up to ~1,000 visits/day/site.
- Visits over 5 years: ~1.8M rows; `visits.db` ~0.5–0.7 GB.
- With indexing, check-ins/check-outs and searches remain performant.
- Future option: `VisitRetentionYears` with archival/purge; not required in v1.

---

13. Libraries (Open Source Only)

- UI/MVVM: `CommunityToolkit.Mvvm` .
- DB + encryption: `Microsoft.Data.Sqlite.Core` , `SQLitePCLRaw.bundle_e_sqlite3mc` .
- Exports: `ClosedXML` , QuestPDF (license-appropriate version).
- QR/Barcode: `QRCoder` , optional `ZXing.Net` .
- Logging: `Serilog` + file sink.

---

14. Non-Goals for v1

- No badge layout designer.
- No per-user language selection.

- No external internet service; HTTP APIs are for LAN.
- No webcam scanning (assume keyboard-wedge scanners).
- No automatic entry/exit toggling based on last state; mode is explicitly chosen.