

DMS Scope Document (Windows Desktop Version)

Project Title: Data Entry Desktop Application for Windows

Development Framework:

- Frontend & Backend: Flutter (Dart)
 - Database: SQLite (Local storage)
-

1. Purpose

To develop a Windows-based desktop application for efficient data entry and management, with secure local data storage using SQLite. The system will allow manual backups and will function entirely offline without cloud hosting.

2. Objectives

1. Offline Data Storage:
 - Store all data locally in an SQLite database.
 2. Fixed Templates for Certificates:
 - Enable structured data entry for birth & death records (no PDFs).
 3. Excel/CSV-Based Reports for Operators:
 - Track the number of words typed by each operator and export reports to Excel/CSV.
 4. Enhanced Usability Features:
 - QR code generation for records.
 - Image zoom functionality for better clarity.
 5. Security & Role-Based Access:
 - Local role-based authentication to restrict unauthorized actions.
 6. Manual Backup & Restore:
 - Users will be responsible for manually backing up and restoring the SQLite database.
-

3. Key Deliverables

3.1 Admin Features

- User Management: Add, edit, and delete users; assign roles.
- Project Management: Create/manage projects and assign users.
- Data Entry Control: Lock/unlock data entries.
- System Logs: Track user actions.
- Reporting: Generate client and operator productivity reports.

3.2 Client Features

- Data Entry Management: Add, search, update, and delete records.
- Data Control: Lock/unlock records.
- File Management: Select a predefined folder path for storing images.
- Reporting: Generate reports by project and date.

3.3 Data Entry Features

- Data Management: Input records with image zoom.
- Export Options: Export entries to Excel or CSV.
- Reporting: Filter reports by project and date.

3.4 System Features

- QR Code Generator for quick access.
 - Image Zoom Tool for better readability.
 - Offline Mode for full functionality without internet.
 - Security Measures: Role-based authentication and local encryption.
 - Manual Backup & Restore for database safety.
-

4. Technical Specifications

- Frontend & Backend: Flutter (Dart)
 - Database: SQLite
 - Security: Role-based authentication, local encryption
-

5. Project Timeline

The development and testing of this application will take approximately 10-12 weeks.

6. Quality Assurance (QA)

- Functional Testing: Verify all features work correctly.
 - Usability Testing: Ensure a smooth user experience.
 - Performance Testing: Optimize app speed and efficiency.
 - Security Testing: Validate encryption and access control.
-

7. Success Metrics

- Fast & efficient data retrieval from SQLite.
 - Positive user feedback on offline usability.
 - Secure handling of sensitive client data with zero breaches.
-

8. Risk Management

- Data Loss Risks: Users must manually back up the database.
 - Security Concerns: Implement role-based access control (RBAC).
 - Windows-Specific Issues: Optimize performance for Windows OS.
-

9. Stakeholder Roles

- Project Manager: Oversees development and ensures deadlines.
- Flutter Developers: Implement UI, database, and business logic.
- QA Team: Conduct testing and ensure quality standards.
- Clients: Provide feedback during testing and deployment.