System Requirements & Specifications (SRS)

Real Estate &
Plot Report Generation

Prepared for:	**Dr. AbdulRahman AbdulHak**
Prepared by:	**Nour Habra**
Date:	**2025/07/15**

Table of Contents

Introduction

1.1 Purpose

The purpose of this document is to define the System Requirements and Specifications (SRS) for a web application that generates property assessment reports based on uploaded geospatial KML data. These reports are designed to be submitted to banks and loaning institutions and follow specific standards such as those of the Bank of Cyprus.

This document is intended to be used by developers, stakeholders, designers, and testers involved in the design and implementation of the system.

1.2 Scope

The application enables users to upload real estate parcel data in KML format, populate and manage report templates, visualize property boundaries on maps, edit and embed images, and generate PDF reports with dynamic content and structured appendices.

There are four types of user roles (Admin, Starter, Checker, Submitter), each contributing to a multi-step report generation and verification workflow. The system supports templated document creation, multi-company support, and dynamic visualizations based on KML data.

The development will be delivered in two phases:

- Phase 1 (4 weeks): Core report generation workflow, KML parsing, photo editing and embedding, and the complete flow from template selection to report export.
- Phase 2: Advanced functionality including role-based checklists, document appendix management, and super-admin multi-tenancy controls.

1.3 Stakeholders

1. Admin users (within each company): Define templates and control available structure.
2. Report Creators (Starters): Initiate and fill reports with data and imagery.
3. Checkers: Validate reports and provide feedback.
4. Submitters: Finalize and submit reports.
5. Bank Staff: End recipients of the generated reports (external stakeholder).
6. Super Admin: Manage companies and high-level access.

Overall Description

2.1 System Perspective

This system is a web-based, multi-tenant application designed for generating real estate reports based on KML geospatial data. It functions as a workflow platform for compiling, verifying, and submitting standardized documents for banking and financing purposes.

The platform integrates:

- A document templating system managed by admins.
- A user-role-driven lifecycle from report creation to submission.
- KML parsing and static map rendering via external mapping APIs.
- An image editing utility for report visuals.

The backend will use:

- PostgreSQL to manage structured data like users, reports, and companies.
- MongoDB to store flexible, admin-defined document template structures.

2.2 User Roles and Permissions

Role	Capabilities
Super Admin	Manage companies and view all data across tenants.
Admin	Upload and configure report templates (mark dynamic/static fields, image sections); manage visibility of templates.
Starter	Select templates, upload KML, fill dynamic fields, upload/edit photos, and submit reports for checking.
Checker	Review submitted reports, leave comments, change status to "Checked" or "Needs Modification".
Submitter	Final review and submission of reports; add appendices and export to PDF.

Each user belongs to a specific company, and companies operate in isolated data spaces using tenant identifiers (via foreign keys).

2.3 System Features Overview

Key modules include:

- Template Management: Uploading and configuring document templates with placeholders and image regions.
- Report Lifecycle: Report creation, editing, status changes across a structured workflow.
- KML Visualization: Parsing uploaded KML files and rendering maps with property boundaries.
- Photo Editor: Cropping, drawing, and pixelating images before embedding them into reports.

- Checklist System: Role-based checklists to verify content quality before progressing reports.
- Appendix Management: Final report can include uploaded PDFs, images, and notes as annexed content.

2.4 Assumptions and Dependencies

- Each company's users and data are separated via logical constraints (no schema isolation).
- Mapping overlays will be powered by a third-party service (e.g., Google Maps Static API).
- KML uploads will follow standard schema containing coordinates and zone metadata.
- PDF/Word templates must be preformatted and consistent with placeholder logic.
- All images and documents are assumed to be internally hosted and served via CDN or blob storage.

System Features and Functional Requirements

3.1 Phase 1 – Core Features

3.1.1 Template Management (Admin)

- Upload Word or PDF templates.
- Define placeholders (static text, dynamic text, image zones).
- Mark templates as active/inactive.

3.1.2 KML Upload and Geospatial Extraction

- Upload a KML file to extract:
- Plot number, coordinates, area, and zone.
- Use this data to auto-fill dynamic fields in the report.

3.1.3 Static Map Rendering

- Generate static maps (satellite, terrain, road, hybrid) using extracted KML boundaries.
- Overlay boundaries on map views via an external map service.

3.1.4 Photo Editor

- Users can crop, draw, and pixelate images before inserting them into designated report sections.
- Ensure edited images maintain quality and fit within layout constraints.

3.1.5 PDF Report Export

- Compile dynamic text, maps, and images into a finalized report.
- Export the report as a PDF, matching the admin's template layout.

3.2 Phase 2 – Extended Features

3.2.1 Full Report Lifecycle Management

Support statuses:

- In Progress
- Submitted for Checking
- Checked
- Needs Modification
- Completed.

Status transitions controlled by user roles.

3.2.2 Role-Based Access Control

- Restrict actions based on user type (Admin, Starter, Checker, Submitter).
- Each role sees only the features relevant to them.

3.2.3 Company and User Management

- Each company has its own users and templates.
- Admins manage users within their company.
- Super Admin can manage all companies and users.

3.2.4 Role-Based Checklists

- Each role (Starter, Checker, Submitter) has its own checklist to ensure report quality.
- Checklists guide users before progressing the report to the next stage.

3.2.5 Appendix Management

- Submitters can attach files (PDFs, images) at the end of the report.
- These are added as annexed pages in the final exported PDF.

3.2.6 Super Admin Dashboard

- View and manage all companies and users.
- Monitor report volume and system usage.

3.2.7 Audit Logs and Enhanced Permissions

- Track actions (edits, submissions, approvals).
- Allow fine-grained control over template visibility and report access.

3.2.8 UX Improvements and Polishing

Final refinements to improve navigation, clarity, and responsiveness of the system UI.

Data Requirements

4.1 Structured Data (Stored in PostgreSQL)

The application will store business-critical information in a relational database. This includes:

- Companies: Information about each company using the system.
- Users: Employees working in each company, along with their roles (e.g., Admin, Starter, Checker, Submitter).
- Reports: All reports generated using the system, including their statuses and history of changes.
- KML Information: Key data extracted from uploaded KML files such as plot number, coordinates, area, and zone classification.
- Images: Photos uploaded by users and any edited versions used within the reports.
- Appendices: Additional documents (PDFs, images, etc.) attached to the final report as annexes.

Each piece of data is linked appropriately — for example, each report is linked to the company it belongs to, and each image is linked to the report where it was used.

4.2 Semi-Structured Data (Stored in MongoDB)

This is used to manage flexible templates that admins create.
For each uploaded document (Word or PDF), the admin can define which parts are:

- Static text that never changes.
- Dynamic fields that need to be filled in later by users.
- Image sections where users will place relevant photos.

Since templates may vary a lot between companies or projects, a flexible format is used here to store them, allowing different layouts and placeholder setups without strict formatting rules.

4.3 File Storage

The system will handle and store a wide variety of user-uploaded and system-generated files, such as:

- KML files (used to extract map and plot data)
- Images (photos of the plot, with or without edits)
- Final PDF reports
- Appended documents (e.g., scanned papers, legal documents, or additional images)

These files will be stored and referenced securely within the application, allowing users to access and include them in reports as needed.

4.4 Audit and History Tracking

To maintain transparency and accountability throughout the report workflow, the system will log key user actions and changes, such as:

- Who created or edited a report, and when.
- When a report status was changed (e.g., submitted, checked, completed).
- Comments and feedback added during review stages.

This information will be accessible to authorized users and used for internal review, traceability, and compliance with professional reporting standards.

External Interfaces

This section outlines the key external systems or services that the application interacts with to deliver its functionality.

5.1 Mapping and Visualization API

The application will generate static map images based on geospatial data (KML files) uploaded by users. This is used to visualize plot boundaries within the report (as seen in the sample PDF).

A third-party static mapping service (e.g., Google Maps Static API, or an equivalent) will be used to:

- Render different map views (satellite, road, terrain, hybrid).
- Overlay the property boundary extracted from the KML file.
- Ensure consistent, high-quality map visuals in the final report.

The specific API service may be chosen and integrated during implementation.

5.2 Document Viewing and Export

The system must support:
Uploading Word/PDF templates.

- Viewing these templates within the application for defining placeholders.
- Generating final reports in PDF format, including all dynamic data, images, maps, and appended content.

This may require integration with:

- A PDF generation library (e.g., pdf-lib, Puppeteer, or similar).
- A Word/PDF reader to extract text positions for placeholder mapping.

5.3 File Storage and Retrieval

The system interacts with internal or cloud-based storage to handle:

- Upload and retrieval of KMLs, images, and PDFs.
- Secure access to files based on user roles and company context.
- Storage structure that maintains clear relationships between files and their associated reports.

Depending on deployment environment, this could be local storage or cloud storage like Amazon S3, Google Cloud Storage, etc.

5.4 Authentication and Permissions (Internal)

Although not "external" in a technical sense, the application must interface with an authentication provider (e.g., session-based or JWT-based identity system) to:

- Control user access (RBAC) to templates, reports, and actions.
- Enforce company-level data boundaries.
- Apply role-specific permissions throughout the application.

Non-Functional Requirements

6.1 Security

- Role-based access control to restrict user actions.
- Data isolation per company using foreign keys.
- Secure file upload, storage, and retrieval.
- Authentication for all users.

6.2 Scalability

- System should support many concurrent users across multiple companies.
- Templates and reports must remain performant even with growing volume.

6.3 Usability

- Simple and intuitive UI per user role.
- Minimal training required for report creators and reviewers.

Development Phases

7.1 Phase 1 – Core Features

This phase (4 weeks) focuses on delivering the essential features required for report generation, review, and submission.

Included Features:

- Admin template upload and placeholder marking
- KML upload and auto-extraction of geospatial data
- Static map rendering using external API
- Photo editor (crop, draw, pixelate)
- PDF export of the final report

Goal: Enable a fully functioning end-to-end flow for producing, reviewing, and exporting reports based on KML data and document templates.

7.2 Phase 2 – Extended Features

This phase, whose duration is to be agreed upon later, will introduce advanced workflow, management, and customization capabilities.

Included Features:

- Role-based access control
- Company and user account management
- Full report lifecycle (In Progress → Completed)
- Role-based checklists to ensure report quality at each stage
- Appendix management (attach PDFs/images at end of reports)
- Super Admin dashboard for company and user oversight
- Enhanced permissions and audit views
- UX improvements and polishing

Goal: Strengthen control, traceability, and flexibility for multi-company usage.