

Individual Thesis

Automating Specific Operations Risk Assessment for UAS Operations

Evangelia Krigkou

UNIVERSITY OF CYPRUS



COMPUTER SCIENCE DEPARTMENT

May 2026

UNIVERSITY OF CYPRUS
COMPUTER SCIENCE DEPARTMENT

Automating Specific Operations Risk Assessment for UAS Operations
Evangelia Krigkou

Supervising Professor
Dr. Panayiotis Kolios

The Individual Thesis was submitted in partial fulfilment of the requirements for the degree of Computer Science of the Department of Computer Science of the University of Cyprus.

May 2026

Acknowledgements

I would like to express my sincere gratitude to my supervisor, Dr. Panayiotis Kolios, for his guidance and valuable feedback throughout the development of this thesis.

I would also like to thank my colleagues at KIOS, Frantzeska Michael, and Yiannis Grigoriou, for their help and support over the past two years, particularly in understanding and applying the SORA methodology.

Finally, I am deeply grateful to my family and friends for their continuous encouragement, patience, and support throughout my studies and the completion of this work.

Abstract

Unmanned Aircraft Systems are increasingly used in a wide range of operations. As these operations become more complex, regulatory safety assessment becomes necessary, particularly for operations in the Specific category. The Specific Operations Risk Assessment methodology provides a structured assessment process, but manual completion can be difficult because applicants must combine risk parameters, calculations, operational-area information, mitigations, and supporting documents.

This thesis presents the design and implementation of a web-based platform for preparing, generating, submitting, and reviewing SORA applications. The system supports both SORA 2.0 and SORA 2.5, with SORA 2.5 as the primary current workflow. It provides guided application completion, draft saving, validation feedback, map-based area definition, automated calculations, document generation, signed document upload, and review-status tracking.

The system includes an interactive map for defining the operational area and presenting supporting spatial information. It also calculates and stores key outputs, including UAS physical values, GRC, final GRC, SAIL, and containment-related values where applicable. These outputs are reused in completed views and generated documents.

The platform supports Civil Aviation Authority staff with a review dashboard that allows submitted applications and documents to be inspected, accepted, or rejected, with reasons recorded. Review outcomes are visible to applicants, and email notifications are sent when documents are accepted or rejected. The prototype uses controlled local user management, where authorised CAA staff create and manage user accounts rather than relying on public self-registration.

The evaluation shows that the platform improves structure, consistency, traceability, and usability by connecting application data, calculations, map information, documents, submissions, and review decisions within a single workflow.

List of Abbreviations

AJAX	Asynchronous JavaScript and XML
AMC	Acceptable Means of Compliance
ARC	Air Risk Class
BVLOS	Beyond Visual Line of Sight
CAA	Civil Aviation Authority
CONOPS	Concept of Operations
CSRF	Cross-Site Request Forgery
DCA	Department of Civil Aviation
EASA	European Union Aviation Safety Agency
ERP	Emergency Response Plan
EVLOS	Extended Visual Line of Sight
GeoJSON	Geographic JavaScript Object Notation
GIS	Geographic Information System
GM	Guidance Material
GRC	Ground Risk Class
iGRC	Intrinsic Ground Risk Class
JARUS	Joint Authorities for Rulemaking on Unmanned Systems
MVT	Model-View-Template
OSO	Operational Safety Objective
SAIL	Specific Assurance and Integrity Level
SORA	Specific Operations Risk Assessment
TMPR	Tactical Mitigation Performance Requirement
UA	Unmanned Aircraft
UAS	Unmanned Aircraft System
UAV	Unmanned Aerial Vehicle
UUID	Universally Unique Identifier
VLOS	Visual Line of Sight

Contents

CHAPTER 1. INTRODUCTION	1
1.1 PROBLEM STATEMENT	2
1.2 AIM AND OBJECTIVES	3
1.3 THESIS CONTRIBUTION	3
1.4 SCOPE OF THE THESIS	4
1.5 THESIS STRUCTURE.....	4
CHAPTER 2. BACKGROUND AND RELATED WORK	6
2.1 UNMANNED AIRCRAFT SYSTEMS.....	6
2.2 UAS OPERATION CATEGORIES.....	7
2.3 THE SORA METHODOLOGY.....	8
2.4 EXISTING DIGITAL APPROACHES TO RISK ASSESSMENT.....	13
2.5 LIMITATIONS OF THE MANUAL SORA PROCESS	13
CHAPTER 3. REQUIREMENTS ANALYSIS	15
3.1 SYSTEM OVERVIEW	15
3.2 SYSTEM STAKEHOLDERS	19
3.3 FUNCTIONAL REQUIREMENTS	21
3.4 NON-FUNCTIONAL REQUIREMENTS	23
3.5 USE CASE ANALYSIS	25
3.6 SYSTEM WORKFLOW	34
CHAPTER 4. SYSTEM DESIGN	38
4.1 DESIGN OVERVIEW	38
4.2 SYSTEM ARCHITECTURE	39
4.3 USER INTERFACE DESIGN.....	43
4.4 DATABASE DESIGN	46
4.5 BACKEND DESIGN	54
4.6 SECURITY AND PRIVACY DESIGN.....	56
CHAPTER 5. IMPLEMENTATION	59
5.1 DEVELOPMENT ENVIRONMENT AND TECHNOLOGIES.....	59
5.2 DATABASE MODELS IMPLEMENTATION	62
5.3 BACKEND IMPLEMENTATION	63
5.4 APPLICANT ENTRY POINT AND APPLICATION MANAGEMENT DASHBOARD.....	67
5.5 SHARED APPLICATION COMPONENTS.....	70
5.6 SORA 2.0 IMPLEMENTATION	77
5.7 SORA 2.5 IMPLEMENTATION	81

5.8 VALIDATION AND COMPLETION FEEDBACK.....	93
5.9 DOCUMENT GENERATION AND SUBMISSION WORKFLOW	97
5.10 ADMINISTRATIVE DASHBOARD AND REVIEW WORKFLOW	101
5.11 INTERNATIONALISATION IMPLEMENTATION	104
CHAPTER 6.SYSTEM EVALUATION AND CONTRIBUTION	105
6.1 CHALLENGES OF THE TRADITIONAL SORA PROCESS	105
6.2 SYSTEM CONTRIBUTION	107
6.3 COMPARISON WITH THE MANUAL APPROACH.....	108
6.4 USABILITY AND PERFORMANCE DISCUSSION	109
CHAPTER 7. CONCLUSION	112
7.1 CONCLUSION	112
7.2 SYSTEM LIMITATIONS.....	113
7.3 FUTURE WORK.....	115
REFERENCES	117

Chapter 1

Introduction

1.1 Problem Statement	2
1.2 Aim and Objectives	3
1.3 Thesis Contribution	3
1.4 Scope of the Thesis	4
1.5 Thesis Structure	4

Unmanned Aircraft Systems are increasingly used in civil, commercial, research, and public-service operations. Their use includes mapping, inspection, emergency response, environmental monitoring, agriculture, and other activities where aerial data collection or remote operation is useful. As these operations become more common and complex, safety assessments are necessary, particularly when drones operate near people, infrastructure, controlled airspace, or restricted areas.

Within the European regulatory framework, UAS operations are classified by risk level [1]. Operations in the Specific category require an operational risk assessment before authorisation can be granted. The Specific Operations Risk Assessment methodology provides a structured process for assessing these operations, considering ground and air risks, mitigations, containment, operational safety objectives, and supporting evidence.

Although the SORA methodology grants a systematic approach, completing it manually can be difficult. Applicants must understand technical concepts, correctly apply calculation tables, define the operational area, prepare supporting documentation, and ensure the submitted material is complete and consistent. This thesis addresses this

challenge by designing and implementing a web-based platform that supports operators and CAA staff throughout the SORA application process.

1.1 Problem Statement

The manual preparation of a SORA application is complex and time-consuming. Applicants must work with several interconnected technical parameters, meaning that an incorrect or incomplete earlier step can affect later parts of the assessment.

The process also requires applicants to navigate among guidance materials, calculation tables, document templates, and mapping tools. This creates a fragmented workflow in which information may need to be copied or recalculated several times. As a result, manual completion could lead to calculation errors, missing fields, inconsistent documentation, or incorrect application of SORA steps.

The operational area adds further difficulty because it affects several assessment outputs. Applicants must identify the intended area of operation, consider population density, potential interactions with restricted, reserved or authorisation-required zones, and apply this information to the risk assessment. When these tasks are handled separately, the relationship between the map and the calculated risk values may be unclear.

There are also difficulties for CAA staff. When applications are prepared manually through separate documents, reviewers may need to spend additional time verifying that the assessment steps were applied correctly, that calculated values are consistent with the provided operational data, and that the submitted documents contain all required information. Incomplete or inconsistent submissions can therefore slow down the review process and may require further communication between the authority and the applicant.

Therefore, there is a need for a guided digital system to support the preparation, generation, submission, and review of SORA applications. Such a system can help to organise the workflow, reduce repeated manual work, improve consistency, and support applicants in completing the required assessment material, as well as reviewers in evaluating consistent and correctly completed documents.

1.2 Aim and Objectives

The aim of this thesis is to design and develop a web-based system that supports and streamlines the SORA application process for UAS operations and CAA reviewers.

The objectives of the thesis are:

- 1 To study the SORA 2.0 and SORA 2.5 methodologies and identify the main requirements that affect system design.
- 2 To define the functional and non-functional requirements of a digital SORA application platform.
- 3 To design the system architecture, database structure, user interface, backend logic, and review workflow.
- 4 To implement a functional web-based system that supports applicants and CAA staff.
- 5 To support guided application completion through form sections, draft saving, validation, map-based operational area definition, automated calculations, document generation, and signed document upload.
- 6 To support the review workflow through submitted-document access, review-status management, rejection-reason recording, applicant feedback, and email notification of review outcomes.
- 7 To evaluate the developed system in terms of its contribution to usability, consistency, traceability, and reduction of manual complexity.

1.3 Thesis Contribution

This thesis presents a web-based platform to support the SORA application process. The platform supports both SORA 2.0 and SORA 2.5, with SORA 2.5 treated as the primary current workflow and SORA 2.0 retained as a legacy and comparative workflow.

The main contribution is transforming a manual and fragmented process into a guided digital workflow. The developed system organises application data, performs selected calculations, integrates map-based operational area definition, generates application documents, and supports the review of submitted material.

The thesis also demonstrates how software engineering can support a regulatory aviation process. The system does not replace authorised regulatory review, but it provides a structured environment that improves consistency, traceability, and management of SORA application data.

1.4 Scope of the Thesis

The scope of the thesis includes the design and implementation of a prototype web-based platform for SORA application preparation and review. The system includes applicant-facing and staff-facing functionality.

The implemented scope includes user authentication, CAA-managed local user account creation, role-based access, SORA 2.0 and SORA 2.5 application creation, draft saving, guided form completion, map-based operational area definition, reserved-area conflict checking, risk-related calculations, application completion, document generation, signed document upload, review-status tracking, email notification support, and administrative review. The system also supports English and Greek interface content.

The thesis does not include fully automated regulatory decision-making. Final acceptance or rejection remains the responsibility of authorised Civil Aviation Authority staff. The system also does not guarantee operational safety solely through its use, because the assessment depends on correct applicant input, valid methodology implementation, accurate supporting data, and official review.

The developed platform is evaluated as a thesis prototype. Further validation, security hardening, production deployment, and real-user testing would be required before operational use at scale.

1.5 Thesis Structure

This thesis is organised into seven chapters.

Chapter 1 introduces the research problem, aim, objectives, contribution, scope, and structure of the thesis.

Chapter 2 presents the background and related work. It introduces UAS operations, operation categories, the SORA methodology, the distinction between SORA 2.0 and SORA 2.5, and the limitations of the manual process.

Chapter 3 presents the requirements analysis, including stakeholders, functional requirements, non-functional requirements, use cases, and system workflow.

Chapter 4 describes the system design, including the architecture, user interface, database, and backend designs, as well as security and privacy considerations.

Chapter 5 explains the implementation of the developed system, including the technologies used, database models, backend processing, application workflows, validation, document generation, administrative review, and internationalisation.

Chapter 6 evaluates the system's contribution by comparing the developed workflow with the manual approach and discussing usability and performance aspects.

Chapter 7 concludes the thesis by summarising the work, identifying limitations, and outlining future work.

Chapter 2

Background and Related Work

2.1 Unmanned Aircraft Systems	6
2.2 UAS Operation Categories	7
2.3 The SORA Methodology	8
2.4 Existing Digital Approaches to Risk Assessment	13
2.5 Limitations of the Manual SORA Process	13

This chapter provides the theoretical and regulatory background for the thesis. It introduces Unmanned Aircraft Systems, explains the European UAS operation categories, presents the Specific Operations Risk Assessment methodology, and compares SORA 2.0 with SORA 2.5. It also discusses existing digital approaches to UAS risk assessment and identifies the limitations of the manual SORA process.

2.1 Unmanned Aircraft Systems

An Unmanned Aircraft System is not only the aircraft itself, but the complete system required to operate it. Under the European UAS regulatory framework, a UAS includes the unmanned aircraft and its control and monitoring unit [1]. This distinction is important because the safety of a drone operation depends not only on the airborne vehicle but also on the remote-control equipment, communication link, operational procedures, and personnel involved.

In common usage, the term “drone” often refers to an unmanned aircraft. The term “UAV” usually refers to the unmanned aerial vehicle itself, while “UAS” refers to the wider system. A remote pilot is the person responsible for conducting the flight, while the

UAS operator is the natural or legal person operating or intending to operate one or more UAS. These distinctions are relevant because regulatory obligations may apply to the operator, the remote pilot, or both, depending on the category and type of operation.

UAS are used in a wide range of civil and commercial applications. These include aerial photography, mapping, infrastructure inspection, agriculture, emergency response, environmental monitoring, research, and traffic monitoring. Recent research also highlights the role of UAS in critical infrastructure monitoring, search and rescue, power and telecommunication inspection, and traffic-related data collection [6].

The growth of UAS operations creates new safety concerns. Drones may operate near people, buildings, roads, infrastructure, controlled airspace, airports, or other aircraft. Loss of control, navigation errors, communication failures, or incorrect operational planning may create risks to third parties on the ground or in the air. For this reason, UAS regulation is risk-based: operations with lower risk are subject to lighter requirements, while operations with higher risk require more detailed assessment and authorisation.

2.2 UAS Operation Categories

European UAS regulation classifies operations into three categories: Open, Specific, and Certified [1]. These categories are based on the level of operational risk. The Open category covers lower-risk operations and is generally subject to predefined operational limits. The Specific category covers operations that do not meet the Open category conditions and therefore require an operational authorisation, a declaration under a standard scenario, or another acceptable regulatory route. The Certified category applies to operations with the highest level of risk, such as those involving the transport of people or certain assemblies of people.

The category of an operation depends on several factors, including the characteristics of the aircraft, the operating environment, the proximity to uninvolved persons, the type of airspace, the altitude of operation, the nature of the mission, and the mitigations applied. EASA's categorisation reflects the principle that regulatory requirements should be proportionate to the risk created by the operation.

The Specific category is the main regulatory context for this thesis. When an operation cannot be conducted under the Open category and is not covered by a standard scenario or predefined risk assessment, the operator needs to perform an operational risk assessment. The SORA methodology provides an acceptable approach for conducting this assessment under Article 11 of Commission Implementing Regulation (EU)2019/947 [1].

2.3 The SORA Methodology

The Specific Operations Risk Assessment methodology is a structured process for assessing UAS operations in the Specific category [1]. Its purpose is to evaluate whether a proposed UAS operation can be conducted with an acceptable level of safety. The methodology supports both the UAS operator and the competent authority by providing a common process for identifying risks, applying mitigations, determining safety objectives, and preparing the evidence required for authorisation.

The SORA methodology focuses mainly on aviation safety risks, particularly the risk of fatal injury to third parties on the ground and in the air [4]. EASA workshop material explains that SORA addresses air safety by protecting uninvolved persons through mitigations and safety objectives derived from the assessment. The same material also clarifies that risks such as privacy, security, environmental, and industrial infrastructure risks are not the focus of SORA and may need to be addressed through other applicable requirements.

SORA is based on several key outputs. These include the initial GRC & ARC, the application of mitigations, the determination of the final GRC, the residual ARC, the SAIL, the TMPR, the OSOs, and the final safety evidence submitted to the competent authority. The methodology, therefore, serves as both a risk classification process and a documentation process.

In Cyprus, the Department of Civil Aviation explains that UAS operators must conduct SORA when the intended operation cannot be adapted to Open category rules, a predefined risk assessment, or a standard scenario [3]. The Cyprus DCA also states that

Article 11 of Commission Implementing Regulation (EU) 2019/947 concerns the Specific category risk assessment procedure.

2.3.1 Overview of the SORA Process

The SORA process begins with a pre-application evaluation. Before starting the full assessment, the operator must determine whether the operation falls under the Open category, is covered by a standard scenario or predefined risk assessment, falls under the Certified category, or is subject to a specific no-go decision from the competent authority. If none of these conditions applies, the SORA process can proceed [3].

The first main step is the description of the proposed operation. This includes the operational purpose, UAS characteristics, operating environment, flight geography, operational volume, adjacent areas, airspace information, and supporting procedures. This description provides the basis for the remaining assessment steps.

The process then evaluates ground risk and air risk. Ground risk concerns the potential effect of the operation on people on the ground, while air risk concerns the possibility of conflict with other airspace users. Mitigations may be applied to reduce these risks, and the resulting values are used to determine the SAIL.

The SAIL then determines the level of safety objectives and evidence required to support the operation. In SORA 2.5, containment and adjacent-area considerations are treated more explicitly before the final safety portfolio is completed.

The process is sequential but interdependent. Earlier outputs affect later steps, and changes to the operation may require parts of the assessment to be recalculated or revised.

2.3.2 SORA 2.0

SORA 2.0 is the earlier version of the methodology supported in this thesis. It was adopted by EASA as part of the AMC and GM material connected to Article 11 of Regulation (EU) 2019/947. The EASA workshop material describes the SORA 2.0 structure as comprising the main body and annexes covering ConOps, GRC mitigations, ARC strategic and tactical mitigations, and OSOs [4].

In SORA 2.0, the first main step is the Concept of Operations. The ConOps describes the intended operation, the UAS used, the operating location, and the procedures followed by the operator. The Cyprus DCA states that the ConOps is the foundation for the remaining SORA activities and should therefore be accurate and detailed [2].

A defining characteristic of SORA 2.0 is that the intrinsic GRC is determined mainly from the operational scenario and the UA's maximum characteristic dimension. The scenario considers whether the operation is VLOS or BVLOS and whether it takes place over a controlled ground area, sparsely populated area, populated area, or assembly of people.

After the intrinsic GRC is identified, SORA 2.0 applies ground risk mitigations. These include strategic ground-risk mitigations, mitigation of ground-impact effects, and emergency response planning. If the claimed mitigations meet the required robustness level, correction factors are applied to obtain the final GRC.

The air risk process follows the ARC structure. The initial ARC may be reduced through strategic mitigations, and any remaining air risk is addressed through tactical mitigation where applicable. The final GRC and residual ARC are then used to determine SAIL and the corresponding OSOs.

2.3.3 SORA 2.5

SORA 2.5 is the updated methodology considered in this thesis [3]. The Cyprus DCA states that EASA Executive Decision 2025/018/R incorporated SORA 2.5 into the AMC and GM under Commission Implementing Regulation (EU) 2019/947, and that applicants for operational authorisations in Cyprus shall use SORA 2.5 from 01/03/2026 onwards [3].

SORA 2.5 retains the general risk-based purpose of SORA while changing several technical and structural elements. The EASA workshop material identifies important updates, including a more quantitative approach to ground risk, improved containment,

clarification of ground risk mitigations, a functional-test-based approach in Annex E, and a new Annex F for the quantitative ground risk model [4].

The most important change concerns ground risk. Instead of relying mainly on qualitative operational scenarios, SORA 2.5 uses more explicit parameters, including maximum UA characteristic dimension, maximum speed, controlled-ground-area status, and population density. The Cyprus DCA SORA 2.5 guidance states that the applicant must identify the flight geography, calculate the contingency volume, calculate the initial ground risk buffer, and identify the highest population density within the iGRC footprint [3].

SORA 2.5 also gives containment a clearer role. Containment considers whether the UA can be kept within the intended operational and adjacent areas and is linked to factors such as SAIL, adjacent population density, assemblies of people, and the characteristics of the UA.

Another structural change is the organisation of the safety case. The Cyprus DCA guidance describes a natural break point after Step 9, separating the process into one phase focused on deriving safety requirements and another phase focused on demonstrating compliance with those requirements [3].

2.3.4 Comparison Between SORA 2.0 and SORA 2.5

SORA 2.0 and SORA 2.5 share the same general purpose: both assess the safety of UAS operations in the Specific category by considering ground risk, air risk, mitigations, SAIL, OSOs, and safety evidence. However, SORA 2.5 modifies the structure and several technical details of the methodology.

Aspect	SORA 2.0	SORA 2.5
First step	Concept of Operations.	Documentation of the proposed operation.
Process structure	Ten-step SORA process.	Ten-step process with clearer two-phase organisation.

Ground risk approach	Mainly based on operational scenario and UA characteristic dimension.	Uses more explicit parameters, including population density, UA dimension, speed, and iGRC footprint.
VLOS treatment in ground risk	VLOS/BVLOS is part of the intrinsic GRC table.	VLOS is treated differently and is removed from the Step 2 ground risk classification logic.
Mitigation structure	Includes M1, M2, and M3 ground risk mitigations.	Revises mitigation structure; emergency response planning is no longer treated in the same way as SORA 2.0 ground risk mitigation.
Air risk process	Initial ARC, strategic mitigations, residual ARC, and TMPR.	Largely similar, with additional clarification and adjacent-airspace considerations.
Containment	Addressed later and less explicitly.	Treated as a clearer step before OSO determination.
Safety evidence	OSOs are identified from SAIL.	OSOs and evidence are integrated more clearly into the Comprehensive Safety Portfolio.

The JARUS summary of changes shows that SORA 2.5 updates the main process structure, changes Step 1 from ConOps description to documentation of the proposed operation, revises the ground risk process, introduces a clearer two-phase organisation, and places containment before OSO identification [5].

The most significant difference is the ground risk approach. SORA 2.0 relies mainly on the operational scenario and the UA characteristic dimension, whereas SORA 2.5 uses a more explicit combination of the UA dimension, speed, population density, and iGRC footprint [5]. The air risk process has changed less substantially, although SORA 2.5 provides additional clarification on adjacent airspace and strategic mitigation considerations.

For this thesis, SORA 2.5 is treated as the primary current workflow because it is the version required for Cyprus operational authorisation risk assessments from 01/03/2026 onwards [3]. SORA 2.0 remains relevant as a supported legacy and comparative workflow because it explains the earlier structure and clarifies the transition between the two methodologies.

2.4 Existing Digital Approaches to Risk Assessment

Digital approaches to risk assessment are used to reduce manual effort, improve consistency, and support traceability. In the context of UAS operations, such approaches may include workflow-based platforms, decision-support systems, automated validation tools, document-generation systems, and map-based spatial analysis tools.

Previous research has explored the application and partial automation of SORA. For example, existing work has examined approaches for using ConOps data to support automatic determination of GRC, ARC, SAIL, and OSOs. Related work has also considered web-based questionnaire tools to assess UAS compliance with JARUS SORA 2.0 guidelines and to generate evaluation reports [6].

This approach does not remove the need for regulatory review. However, it shows that software tools can support the SORA process by organising information, reducing repeated manual work, and improving consistency between assessment inputs and outputs.

2.5 Limitations of the Manual SORA Process

The manual execution of SORA presents several limitations. First, the methodology requires the applicant to understand numerous specialised concepts, including operational volume, flight geography, contingency volume, ground risk buffer, GRC, ARC, SAIL, TMPR, OSOs, and containment. These concepts are linked, and misunderstanding one step may affect the correctness of later outputs.

Second, the process depends heavily on disconnected sources of information. The applicant may need to consult official guidance, identify the correct table, perform calculations, use map tools, prepare an operation manual, complete templates, and submit evidence to the competent authority. Each transfer of information between sources creates the possibility of omission or inconsistency.

Third, the map-related part of SORA is difficult to perform manually. The applicant must define the operational area and determine the operational volume, contingency volume, ground risk buffer, adjacent area, and relevant airspace. In SORA 2.5, population density within the iGRC footprint becomes especially important. If map data, population density, restricted areas, and risk tables are handled separately, the relationship between the operational area and risk outputs becomes less direct.

Fourth, the calculation process can be error-prone. Manual calculation of terminal velocity, kinetic energy, operational volume dimensions, GRC, ARC, SAIL, and containment-related outputs requires accurate inputs and correct interpretation of tables. Errors may arise from wrong assumptions, incorrect units, outdated templates, or misapplied mitigation rules.

Fifth, document preparation can introduce inconsistencies. The same application data may need to appear in forms, supporting documents, safety evidence, and final submission material. If these documents are prepared manually, values may be copied incorrectly or updated in one place but not another.

Finally, the transition between SORA 2.0 and SORA 2.5 adds complexity. Applicants must understand which version applies, how the ground risk calculation has changed, which documentation is required, and how containment and safety evidence are treated.

These limitations justify the development of a structured digital system that supports guided completion, calculation consistency, map integration, document generation, and administrative review.

Chapter 3

Requirements Analysis

3.1 System Overview	15
3.2 System Stakeholders	19
3.3 Functional Requirements	21
3.4 Non-Functional Requirements	23
3.5 Use Case Analysis	25
3.6 System Workflow	34

This chapter presents the requirements analysis of the developed SORA application platform. It begins with a system overview that describes the platform's purpose, users, core capabilities, and scope. It then identifies the main stakeholders involved in the preparation, submission, review, administration, and maintenance of SORA applications. The chapter also defines the system's functional and non-functional requirements, covering both the services the platform must provide and the quality attributes it must meet. In addition, persona-based analysis, typical user tasks, and use case descriptions are presented to clarify the needs and interactions of the main user groups. Finally, the chapter explains the overall system workflow, including the applicant and staff review workflows, as well as the lifecycle states of applications and submitted documents. The requirements are based on the SORA application process, the needs of UAS applicants and CAA staff, and the intended Cypriot operational and regulatory context.

3.1 System Overview

The developed system is a web-based application that supports the digital execution of the SORA process for UAS operations. The system provides a structured environment for applicants to prepare, complete, generate, and submit SORA-standard application documents containing auto-filled information, while CAA staff can review and validate

the submitted applications. It, thus, acts as an intermediary digital workflow between UAS operators and the authority responsible for assessing operational risk documentation.

The application supports both SORA 2.0 and SORA 2.5 methodologies, with SORA 2.5 as the primary workflow. This distinction is important because the two versions require different data structures, assessment steps, and document outputs. The system treats SORA 2.0 and SORA 2.5 as distinct application flows within the same platform, with separate creation and editing processes for each version. The user interface also reflects this separation, allowing users to choose between SORA 2.0 and SORA 2.5 when creating a new application.

3.1.1 Purpose of the System

The primary purpose of the system is to assist UAS applicants in preparing risk assessment applications in a guided and standardised manner. Instead of relying solely on manually completed documents, the system structures the SORA process into digital form sections, automatically performs selected calculations, stores application data, and supports document generation. This reduces repetition, improves consistency, and helps ensure that the information required for SORA evaluation is collected systematically.

The system is also intended to support CAA staff by centralising the review of submitted application material. Staff users can access submitted applications, view the associated application data and uploaded documents, and update the review outcome. The administrative interface includes status filtering, sorting, application viewing, document viewing, ZIP downloading for SORA 2.5 outputs, and status management for submitted documents.

3.1.2 System Users

The system is designed around two primary user groups. The first group consists of UAS applicants or operators. These users create SORA applications, complete the required assessment sections, define the operational area, generate application documents, upload signed documents, and monitor the status of their applications. The “My Applications” page demonstrates this user-facing workflow by separating draft applications from

completed applications and by providing actions such as viewing, generating, uploading, and deleting applications.

The second group consists of CAA staff or administrative users. These users access the administrative dashboard, review submitted applications and supporting documents and accept or reject submissions. When rejecting a document, the system requires a rejection reason to support traceability and provide feedback to the applicant.

CAA staff users also perform local user-management tasks. Public self-registration is not provided. Instead, authorised CAA staff access User Management via the Django administration interface to create local users and assign the appropriate role.

3.1.3 Core System Capabilities

The system provides several core capabilities that correspond to the main stages of the SORA application lifecycle. First, it supports user authentication and role-based redirection. Authenticated users are directed either to the applicant home area or to the administrative dashboard, depending on whether they have staff or administrative privileges.

Second, the system supports the creation and management of SORA applications. It includes separate models for application data, including pre-evaluation information, drone specifications, risk assessment data, air risk assessment data, strategic mitigations, map polygons, generated documents, and submitted documents. Applications can exist in different states, including draft, completed, and expired, which allows the platform to manage the application lifecycle over time.

Third, the system provides a guided form-based workflow for applicants. The SORA 2.0 application interface is organised into five main tabs: pre-evaluation, drone specifications, ground risk, and air risk and TMPR. The SORA 2.5 interface is more extensive and is organised into seven visible tabs: pre-evaluation, drone specifications, ground risk, air risk, TMPR and SAIL, OSOs & containment, and operation manual. This tab-based structure helps users complete the assessment progressively and reflects the step like progression of the SORA process regulated by EASA.

Fourth, the system includes an interactive map component for defining the operational area. Users can draw a polygon on the map, undo points, clear the map, and visualise geographic information relevant to the operation. The map module loads population density and no-fly zone data, calculates population density for the selected operational area, and checks whether the operational area or its buffers intersect restricted or reserved zones, among other checks.

Fifth, the system performs automated calculations related to the SORA assessment process. These calculations support the determination of risk-related values such as the Ground Risk Class, Air Risk Class, SAIL, and, where applicable, containment-related outputs. The calculated values are used to assist the applicant during form completion and to support the generation of structured application documents.

Finally, the system supports document generation, upload, and review. Applicants can generate DOCX documents, upload signed PDF documents, and, for SORA 2.5, upload ZIP files containing additional generated documents. Submitted documents are stored with a review status of pending, accepted, or rejected, and a rejection reason can be recorded when necessary.

3.1.4 System Boundary and Scope

The system focuses on supporting the preparation, submission, and review of SORA-related application material. It does not replace the legal authority of the CAA, nor does it automatically approve UAS operations. Instead, it provides a structured digital mechanism for collecting applicant data, applying calculation logic, generating documentation, and supporting administrative review.

The scope of the system therefore includes applicant registration and authentication, SORA 2.0 and SORA 2.5 application creation, draft saving, form completion, map-based operational area definition, risk-related calculations, document generation, signed document upload, and staff review. The system provides multilingual support in English and Greek, reflecting the requirements of its intended Cypriot operational and regulatory environment.

Overall, the system functions as a structured digital workflow for SORA application management. It supports applicants by guiding them through the assessment process and supports Civil Aviation Authority staff by providing a centralised review interface. In this way, the system contributes to greater consistency, traceability, and efficiency in preparing and reviewing UAS operational risk assessment applications.

3.2 System Stakeholders

Stakeholders are the individuals or organisations that interact with the system and utilize its outputs. In the context of the developed SORA application platform, the main stakeholders include applicants, CAA staff, the CAA as an organisation, and the staff-side maintainer function. The maintainer function is not a separate external user group in the prototype; it is performed by authorised CAA staff with access to User Management.

3.2.1 Applicant / UAS Operator

The applicant or UAS operator, is the primary external user of the system. This stakeholder uses the platform to prepare and manage SORA applications for UAS operations. The applicant is responsible for entering the required operational information, completing the relevant assessment steps, defining the operational area, and generating the necessary application documents.

The applicant's main interactions with the system include logging in to the system, creating SORA 2.0 or SORA 2.5 applications, saving applications as drafts, completing the required form sections, generating application documents, uploading signed PDF documents, and monitoring the status of submitted applications. The system supports this stakeholder by providing a structured workflow that reduces manual document preparation and helps ensure that the required assessment information is completed in a consistent manner.

3.2.2 Civil Aviation Authority Staff / Reviewer

CAA staff, or reviewers, are responsible for examining submitted applications and supporting documents. This stakeholder group uses the system to access submitted

application data, review uploaded documents, and determine whether a submission should be accepted or rejected.

The reviewer's main interactions with the system include accessing the administrative dashboard, viewing completed applications, inspecting submitted documents, updating the review status, and providing a rejection reason when an application or document is rejected. This role supports the regulatory review process by ensuring that submitted material is assessed in a traceable and organised manner.

3.2.3 Civil Aviation Authority

The CAA is an organisational stakeholder rather than an individual system user. It benefits from the system because the platform supports a more standardised and traceable application review process. By centralising submitted applications and review outcomes, the system contributes to improved consistency, efficiency, and accountability in the handling of SORA-related documentation.

The authority's interest is primarily related to the reliability of the workflow, the completeness of submitted information, and the ability of staff to review applications using a consistent digital process. The system does not replace the authority's legal decision-making role, rather, it supports the administrative and technical handling of application material.

In the prototype, authorised CAA staff also perform user-management tasks. These staff users can access the Civil Aviation Authority Dashboard for application review and can also access User Management through Django administration to create and manage local user accounts. This means that the reviewer and maintainer functions are performed by the same staff-side role.

3.2.4 Developer Superuser

The developer superuser is a technical account used for system setup and configuration. This account is not part of the ordinary applicant or CAA review workflow. It is reserved for development-level configuration and is protected from modification by non-superuser administrative users.

3.2.5 System Maintainer

The system maintainer function is performed by authorised CAA staff. The maintainer is responsible for managing local user accounts through the User Management area, which is provided through Django administration. This includes creating users, updating user details, and assigning users to the appropriate role, such as applicant or staff.

Public self-registration is not implemented in the prototype. Instead, access is controlled by CAA staff through the creation of local accounts. This approach reflects the platform's controlled-access nature and prevents unverified public users from creating accounts directly. Future production versions will extend this approach through CY Login or GovCY Digital Services Factory authentication.

3.3 Functional Requirements

Functional requirements define the services, actions, and behaviours that the system must provide to its users. In this system, the functional requirements are derived from the SORA application workflow, including applicant registration, application creation, risk assessment completion, operational area definition, document generation, document submission, and administrative review.

Although SORA 2.5 is the current primary workflow for new Cyprus applications, SORA 2.0 is retained in the requirements because the implemented prototype supports both methodologies.

ID	Requirement Description
FR1	The system shall allow authorised CAA staff to create and manage local user accounts through User Management.
FR2	The system shall support user authentication (login/logout).
FR3	The system shall provide role-based access to applicant functions, review functions, and user-management functions.
FR4	The system shall allow applicants to create SORA 2.0 applications.
FR5	The system shall allow applicants to create SORA 2.5 applications.

FR6	The system shall allow applicants to save applications as drafts.
FR7	The system shall support completion of pre-evaluation data.
FR8	The system shall allow applicants to enter UAS and drone specification data.
FR9	The system shall allow drone specification fields to be populated from stored drone model data.
FR10	The system shall support ground risk assessment input.
FR11	The system shall support air risk assessment input.
FR12	The system shall support the entry of mitigation-related information.
FR13	The system shall support interactive map-based definition of the operational area.
FR14	The system shall allow users to revise the operational area, including undoing map points and clearing the map.
FR15	The system shall display population-density information.
FR16	The system shall display restricted, reserved or no-fly areas on the map to support safe operation planning.
FR17	The system shall warn users when the defined operational area intersects with restricted, reserved or no-fly zones.
FR18	The system shall display reserved operational areas from previously accepted applications for the selected operation date.
FR19	The system shall prevent applicants from defining an operational area that overlaps restricted zones or reserved operational areas.
FR20	The system shall calculate risk-related values required by the SORA process, including GRC, ARC, and SAIL.
FR21	The system shall calculate additional SORA 2.5 outputs where applicable, including final GRC and containment-related values.
FR22	The system shall allow completion of the application.
FR23	The system shall allow applicants to delete their own applications.
FR24	The system shall determine application expiry based on the planned operation date.
FR25	The system shall allow document generation in DOCX format on each application.
FR26	The system shall allow applicants to upload signed PDF documents.

FR27	The system shall allow SORA 2.5 applicants to upload additional generated document packages where required.
FR28	The system shall allow staff users to access submitted applications.
FR29	The system shall allow staff users to review submitted application data and uploaded documents.
FR30	The system shall allow staff users to accept or reject submitted documents.
FR31	The system shall require a rejection reason when a submitted document is rejected.
FR32	The system shall display review statuses to applicants, including pending, accepted, and rejected statuses.
FR33	The system shall send email notifications to applicants upon document submission and review-status changes.
FR34	The system shall maintain an audit log of key user actions.
FR35	The system shall provide multilingual support in English and Greek.

3.4 Non-Functional Requirements

Non-functional requirements define the quality attributes, constraints, and operational characteristics of the system. While functional requirements describe what the system must do, non-functional requirements describe how the system should perform, behave, and be maintained. For the developed SORA application platform, these requirements are particularly important because the system handles regulatory application data, performs risk-related calculations, and supports interaction between applicants and Civil Aviation Authority staff.

ID	Category	Requirement Description
NFR1	Usability	The system shall provide a guided and structured interface for completing the SORA process.
NFR2	Usability	The system shall organise the application process into clear sections or steps to support progressive completion.

NFR3	Accuracy	The system shall perform risk-related calculations according to the defined SORA methodologies, tables, and calculation rules implemented in the platform.
NFR4	Accuracy	The system shall minimise the risk of user input errors through validation of required fields and relevant form constraints.
NFR5	Reliability	The system shall preserve application data when applications are saved as drafts.
NFR6	Reliability	The system shall maintain submitted application data and uploaded documents unless they are explicitly deleted by an authorised user.
NFR7	Security	The system shall ensure that applicants can only access their own applications and submitted documents.
NFR8	Security	The system shall restrict staff and administrative functions to authorised users only.
NFR9	Privacy	The system shall protect sensitive user, application, and document data from unauthorised access.
NFR10	Privacy	Public-facing application identifiers shall not expose internal database identifiers or database structure.
NFR11	Maintainability	The system shall allow updates to calculation logic, document templates, localisation content, and supporting datasets when required.
NFR12	Maintainability	The system shall be structured in a way that supports future corrections, extensions, and methodology updates.
NFR13	Performance	The system shall respond to common user actions, form interactions, and map interactions within acceptable time limits during normal use.
NFR14	Compatibility	The system shall operate on modern web browsers.
NFR15	Compatibility	Generated documents shall be compatible with standard document-processing software.
NFR16	Localisation	The system shall provide interface content in both English and Greek.

NFR17	Localisation	The system shall maintain consistent terminology across the English and Greek interfaces.
NFR18	Traceability	The system shall record key user actions and review outcomes to support accountability and auditability.
NFR19	Traceability	The system shall preserve review statuses and rejection reasons so that applicants and staff can track the outcome of submitted documents.
NFR20	Scalability	The system architecture shall allow future expansion, including additional templates, assessment logic, datasets, or user roles if required.

3.5 Use Case Analysis

This section analyses the main ways in which users interact with the SORA application platform. The use case analysis is based on the stakeholders identified in Section 3.2 and the functional requirements defined in Section 3.3. It presents representative user personas, typical user tasks, and detailed use case descriptions in order to clarify the practical needs of applicants, Civil Aviation Authority reviewers, and system administrators. This analysis helps connect the system requirements with the workflows that the platform must support during application creation, completion, submission, and review.

3.5.1 User Personas

Persona 1 - Applicant

Field	Description
Persona	Andreas Georgiou
User group	Applicant / UAS Operator
Background	Andreas is a UAS operator preparing an application for a drone operation in Cyprus. He needs to complete the SORA process and provide the necessary operational and risk assessment information to the Civil Aviation Authority.

Goals	To prepare a complete SORA application, define the operational area, calculate required risk values, generate the application document, and submit the signed documentation for review.
Needs	A guided workflow, clear form sections, automatic calculations, map-based operational area definition, document generation, and status tracking.
Challenges	Manual SORA preparation can be time-consuming, repetitive, and prone to missing information or calculation errors.

Persona 2 - CAA Reviewer

Field	Description
Persona	Maria Christodoulou
User group	Civil Aviation Authority Staff / Reviewer
Background	Maria is responsible for reviewing submitted SORA applications and supporting documents. She needs access to application data, uploaded documents, and review status controls.
Goals	To review submitted application material, assess whether documentation is complete, accept valid submissions, and reject incomplete or incorrect submissions with justification.
Needs	A centralised dashboard, access to submitted documents, clear application information, review-status controls, and the ability to record rejection reasons.
Challenges	Submitted material may be incomplete, inconsistent, or difficult to trace when applications are prepared manually.

Persona 3 - System Administrator

Field	Description
Persona	System Administrator
User group	Administrator / Maintainer

Background	The administrator is responsible for maintaining the correct operation of the platform and ensuring that users, templates, calculation logic, and system content remain up to date.
Goals	To ensure that the system remains reliable, secure, maintainable, and aligned with operational or regulatory changes.
Needs	Access control, maintainable system configuration, editable templates, updateable calculation logic, and localisation support.
Challenges	SORA requirements, templates, or supporting datasets may change over time, requiring the system to remain adaptable.

3.5.2 Typical User Tasks

Task 1 - Creating a new SORA application

The applicant creates a new SORA application by selecting the appropriate SORA version, either SORA 2.0 or SORA 2.5. The system creates an application workflow with the relevant form sections and assessment steps for the selected version.

Field	Description
User group	Applicant / UAS Operator.
Frequency	Performed whenever a new UAS operation requires a SORA application.
Required conditions	The user must be authenticated and must know the SORA process.
Outcome	A new draft application is created and becomes available for completion.

Task 2 - Completing the SORA assessment

The applicant completes the required SORA sections, including pre-evaluation data, UAS specifications, ground risk assessment, air risk assessment, mitigation information, and version-specific fields as well as define the intended operational area. The system supports the process through structured forms and automatic validation.

Field	Description
User group	Applicant / UAS Operator.
Frequency	Performed for every application.
Required conditions	The applicant must have the required operational information, UAS characteristics, risk assessment details, and an operation manual file where applicable.
Outcome	The application contains the necessary data required for assessment and document generation.

Task 3 - Generating and submitting application documents

After completing the application, the applicant generates the required application documents. The generated document is signed externally and then uploaded back into the system as a signed PDF. For SORA 2.5, additional document packages may also be uploaded where required.

Field	Description
User group	Applicant / UAS Operator.
Frequency	Performed once the application has been completed.
Required conditions	The application must contain the required information, and the applicant must have completed the necessary document check process.
Outcome	The application documents are submitted and become available for CAA review.

Task 4 - Reviewing submitted applications

CAA staff access the administrative dashboard to view submitted applications and their associated documents. The reviewer examines the application data and uploaded files to determine whether the submission is acceptable.

Field	Description
User group	CAA Staff / Reviewer.

Frequency	Performed whenever applications are submitted for review.
Required conditions	The reviewer must be authenticated as an authorised staff user, and at least one application must have been submitted.
Outcome	The reviewer can decide on the submitted application material.

Task 5 - Accepting or rejecting a submission

After reviewing a submitted document, the reviewer updates its status to either accepted or rejected. If the submission is rejected, the system requires a reason for the rejection so the applicant can understand the issue.

Field	Description
User group	CAA Staff / Reviewer.
Frequency	Performed for each submitted document after review.
Required conditions	The reviewer must have access to the submitted document and must complete the review decision.
Outcome	The document review status is updated, and the applicant can view the outcome.

3.5.3 Use Case Descriptions

The following use cases describe the main interactions between the system actors and the SORA application platform. Each use case identifies the responsible actor, the objective of the interaction, the main flow of events, and the expected outcome.

UC1 - Create Local User Account

Field	Description
Actor	CAA Staff / Maintainer
Objective	To create a local user account for an applicant or staff user.
Main flow of events	The authorised CAA staff member opens User Management through Django administration, creates a new user account, assigns the appropriate role, and saves the account.

Outcome	The user has a local account and can log into the platform according to the assigned role.
---------	--

UC2 - Log In

Field	Description
Actor	Applicant / UAS Operator, CAA Staff / Reviewer, System Administrator
Objective	To access the system using valid credentials.
Main flow of events	The user enters a username and password. The system validates the credentials and redirects the user according to their role. Applicants are directed to the applicant area, while staff or administrative users are directed to the administrative dashboard.
Outcome	The user gains access to the appropriate system area based on their role.

UC3 - Create New SORA Application

Field	Description
Actor	Applicant / UAS Operator
Objective	To create a new SORA application for a planned UAS operation.
Main flow of events	The applicant selects the option to create a new application, chooses the relevant SORA version, and the system opens the corresponding application workflow for SORA 2.0 or SORA 2.5.
Outcome	A new application is created and becomes available for completion.

UC4 - Save Draft Application

Field	Description
Actor	Applicant / UAS Operator
Objective	To save incomplete application data for later continuation.

Main flow of events	The applicant enters partial application information and selects the save draft option. The system stores the entered data without requiring full completion of the application.
Outcome	The application is saved as a draft and can be edited later by the applicant.

UC5 - Complete SORA Assessment

Field	Description
Actor	Applicant / UAS Operator
Objective	To complete the required assessment information for the selected SORA version.
Main flow of events	The applicant completes the relevant form sections, including pre-evaluation data, UAS specifications, ground risk assessment, air risk assessment, mitigation information, and version-specific assessment fields. The system validates required information and performs relevant calculations.
Outcome	The application contains the required assessment data and can proceed to completion and document generation.

UC6 - Define Operational Area

Field	Description
Actor	Applicant / UAS Operator
Objective	To define the geographical area in which the UAS operation is planned to take place.
Main flow of events	The applicant uses the interactive map to define the operational area. The system displays relevant geographical information, including population-density information and restricted, reserved or no-fly areas, to support safe operation planning.

Outcome	The operational area is stored as part of the application and can be used in the risk assessment process.
---------	---

UC7 - Generate Application Document

Field	Description
Actor	Applicant / UAS Operator
Objective	To generate the required application documentation from the completed application data.
Main flow of events	The applicant completes the application and selects the document generation function. The system uses the stored application data to populate the relevant document template and generate the application document.
Outcome	A DOCX application document is generated for the applicant.

UC8 - Upload Signed Document

Field	Description
Actor	Applicant / UAS Operator
Objective	To submit the signed application document for review.
Main flow of events	The applicant signs the generated document externally, returns to the system, selects the relevant application, and uploads the signed PDF document.
Outcome	The signed document is stored in the system and becomes available for review by CAA staff.

UC9 - View Application Status

Field	Description
Actor	Applicant / UAS Operator

Objective	To monitor the progress and review outcome of submitted documents.
Main flow of events	The applicant opens the application list and views the status associated with each completed application or submitted document.
Outcome	The applicant can see whether the document is pending review, accepted, rejected, or expired.

UC10 - Delete Application

Field	Description
Actor	Applicant / UAS Operator
Objective	To remove an application that is no longer required.
Main flow of events	The applicant selects an application from their application list and chooses the delete option. The system confirms the action and removes the application from the user's records.
Outcome	The selected application is deleted.

UC11 - Review Submitted Application

Field	Description
Actor	CAA Staff / Reviewer
Objective	To inspect submitted application material before making a review decision.
Main flow of events	The reviewer accesses the administrative dashboard, opens a submitted application, views the application data, and inspects the uploaded documents.
Outcome	The reviewer has sufficient information to decide whether the submission should be accepted or rejected.

UC12 - Accept Submission

Field	Description
Actor	CAA Staff / Reviewer
Objective	To mark a submitted document or application material as accepted.
Main flow of events	The reviewer examines the submitted material and selects the accepted status. The system records the review decision and updates the document status.
Outcome	The submitted document is marked as accepted and the applicant can view the updated status.

UC13 - Reject Submission

Field	Description
Actor	CAA Staff / Reviewer
Objective	To reject incomplete, incorrect, or unsuitable submitted material.
Main flow of events	The reviewer examines the submitted material, selects the rejected status, enters a rejection reason, and confirms the decision.
Outcome	The submitted document is marked as rejected, and the rejection reason is stored for applicant feedback and traceability.

3.6 System Workflow

The system workflow describes the sequence of actions through which a SORA application is created, completed, submitted, and reviewed. The workflow is divided into the applicant workflow, the Civil Aviation Authority staff workflow, and the lifecycle states used by the system to represent the progress of applications and submitted documents.

3.6.1 Applicant Workflow

The applicant workflow begins when a user registers or logs into the system. After authentication, the applicant can create a new SORA application by selecting the appropriate methodology, either SORA 2.0 or SORA 2.5. Once the application is created, the system treats it as a draft, allowing the applicant to complete the required information progressively and return to the application later if needed.

The applicant then completes the required sections of the application. These include pre-evaluation information, UAS and drone specifications, operational area definition through the map interface, ground risk assessment input, air risk assessment input, and any additional version-specific information required by the selected SORA methodology. During this process, the system performs automated calculations to support the assessment, including risk-related values such as GRC, ARC, SAIL, and, where applicable, SORA 2.5-specific outputs.

After the required sections have been completed, the applicant can complete the application and generate the relevant application document. The generated document is then signed externally by the applicant. Once signed, the applicant uploads the signed PDF document back into the system. At this stage, the uploaded document becomes available for review by Civil Aviation Authority staff, and the applicant can monitor the review status through the application interface.

In summary, the applicant workflow consists of the following steps:

1. The user logs into the system.
2. The user creates a new SORA 2.0 or SORA 2.5 application.
3. The system creates the application as a draft.
4. The user completes the required application sections.
5. The system performs the relevant automated calculations.
6. The user completes the application.
7. The user generates the application document.
8. The user signs the document externally.
9. The user uploads the signed PDF document.
10. The submitted document becomes available for review.
11. The user monitors the review status.

3.6.2 Staff Workflow

The staff workflow begins when an authorised Civil Aviation Authority staff member logs into the system. If the user has staff or administrative privileges, the system provides access to the administrative dashboard. Through this dashboard, staff users can view submitted applications, inspect application data, and access uploaded documents.

After reviewing the submitted material, the staff user updates the review status. A submission may be accepted if the documentation is considered valid and complete. Alternatively, it may be rejected if the submitted material is incomplete, incorrect, or unsuitable. In the case of rejection, the system requires a rejection reason, ensuring that the applicant receives feedback and that the review decision remains traceable.

The applicant is informed of the outcome through the visible review status in the system and, where applicable, through system-generated notifications. This supports transparency between the applicant and the reviewing authority.

CAA staff may also access User Management through Django administration. This staff-side function allows authorised users to create and manage local accounts separately from the application review workflow.

In summary, the staff workflow consists of the following steps:

1. The staff user logs into the system.
2. The system grants access to the administrative dashboard.
3. The staff user accesses submitted applications.
4. The staff user reviews application data and uploaded documents.
5. The staff user accepts or rejects the submitted document.
6. If the document is rejected, the staff user provides a rejection reason.
7. The system updates the review status.
8. The applicant can view the outcome.

3.6.3 System States

The system uses lifecycle states to represent the progress of both applications and submitted documents. These states help distinguish between applications that are still

being prepared, applications that have been completed, and submitted documents that are awaiting or have received a review decision.

Application Lifecycle States

State	Description
Draft	The application has been created but has not yet been completed. The applicant may continue editing and saving progress.
Completed	The application has been completed by the applicant and can be used for document generation and submission.
Expired	The application is no longer valid in relation to the planned operation date. Draft applications may expire when the two-week submission window has passed, while completed applications may expire after the planned operation date.

Submitted Document Lifecycle States

State	Description
Pending Review	The applicant has uploaded a signed document, and the document is awaiting review by CAA staff.
Accepted	The submitted document has been reviewed and accepted by the staff user.
Rejected	The submitted document has been reviewed and rejected. A rejection reason is recorded to explain the decision.

Chapter 4

System Design

4.1 Design Overview	38
4.2 System architecture	39
4.3 User Interface Design	43
4.4 Database Design	46
4.5 Backend Design	54
4.6 Security and Privacy Design	56

4.1 Design Overview

The system design translates the requirements identified in Chapter 3 into a structured technical solution. While Chapter 3 defines what the SORA application platform must support, this chapter explains how the platform is organised to provide those functions. The design focuses on the architecture, user interface, database structure, backend responsibilities, and security mechanisms required to support the applicant and the Civil Aviation Authority workflows.

The platform is designed around a separation of responsibilities between the presentation layer, the application layer, and the data layer. The presentation layer provides the browser-based interface, including the applicant dashboard, application forms, interactive map, upload interface, completed application views, and administrative dashboard. The application layer coordinates request handling, validation, calculation logic, document generation, file management, and review processing. The data layer stores users, applications, assessment records, operational area data, submitted documents, review outcomes, and audit records.

A central design decision is the separation between SORA 2.0 and SORA 2.5 workflows. Since SORA 2.5 is treated as the primary current workflow, the design places particular emphasis on SORA 2.5-specific requirements, such as population-density-based ground risk assessment, containment handling, uploading the operation manual, and generating multiple document outputs. SORA 2.0 is retained as a supported legacy and comparative workflow, allowing the platform to preserve the earlier methodology while clearly distinguishing it from the current SORA 2.5 process.

The design also integrates an interactive map as a core component of the application workflow. This is necessary because SORA assessment depends not only on textual and technical information, but also on the geographical context of the intended UAS operation.

Therefore, the design combines guided form-based data collection, map-based spatial input, automated calculation support, document generation, file upload, and administrative review into a single structured platform. This design supports applicants during application preparation and submission, as well as CAA staff during review and decision-making.

4.2 System architecture

The system follows a three-tier web architecture, implemented using the Django Model-View-Template (MVT) pattern. This architecture separates user interaction, application processing, and data storage into distinct layers. Such separation improves maintainability, supports clearer system organisation, and allows each part of the platform to be developed and modified independently.

The architecture is well-suited to the SORA application platform because the system requires structured user interfaces, backend processing, database interaction, risk-calculation logic, document generation, file handling, and administrative review functionality. By separating these responsibilities, the system can support both applicant-facing and staff-facing workflows within the same web application.

4.2.1 Architectural Pattern

The system is based on Django's MVT pattern. In this pattern, the model layer represents the system's data structure and manages database interactions. The view layer processes HTTP requests, applies validation and business logic, and determines the response returned to the user. The template layer renders dynamic HTML pages that present system data through the user interface.

Although the system follows the Django MVT pattern, it can also be understood as a three-tier web architecture. The client tier is responsible for user interaction through the browser. The application tier contains the Django backend, including views, forms, utility logic, calculations, document generation, and review processing. The data tier stores application records, user data, uploaded documents, generated documents, operational area data, and review outcomes.

This architectural pattern supports the requirements defined in Chapter 3 by allowing the system to manage complex application workflows, perform automated calculations, preserve user data, and provide separate access paths for applicants and Civil Aviation Authority staff.

4.2.2 Main Architectural Layers

The system architecture can be divided into three main layers: the presentation layer, the application layer, and the data layer.

Layer	Components	Responsibility
Presentation Layer	Browser, HTML templates, CSS, JavaScript, interactive map	Provides the user interface and handles user interaction.
Application Layer	Django views, forms, validation logic, calculation utilities, document generation logic, review workflow logic	Processes requests, validates data, performs calculations, generates documents, and controls application behaviour.

Data Layer	Database models, file storage, uploaded documents, generated documents	Stores and retrieves system data, application records, documents, map data, and review statuses.
-------------------	--	--

The presentation layer is accessed through a web browser. It includes the applicant interface, the application form pages, the interactive map, the document upload interface, and the administrative dashboard. JavaScript is used to support dynamic behaviour, such as form interaction, asynchronous requests, map updates, and client-side feedback.

The application layer is implemented through the Django backend. It receives user requests, processes form submissions, validates application data, performs SORA-related calculations, manages uploaded files, generates documents, and handles review decisions. This layer is central to the system because it connects the user interface with stored application data and system outputs.

The data layer stores all persistent information required by the platform. This includes user accounts, SORA 2.0 and SORA 2.5 application data, pre-evaluation data, UAS specifications, risk assessment data, operational area information, submitted documents, generated documents, review statuses, rejection reasons, and audit records.

4.2.3 Request-Response Lifecycle

The system operates through a standard web request-response lifecycle. A user action begins in the browser, such as opening an application form, saving a draft, completing an application, uploading a document, or reviewing a submission. The browser sends an HTTP request to the Django backend, where the appropriate view processes the request.

If the request contains form data, the backend validates the submitted information and either stores it in the database or returns validation feedback to the user. If the request requires calculation, the backend applies the relevant SORA calculation logic and returns the computed result. If the request involves document generation, the backend retrieves stored application data, applies it to the relevant document template, and produces the required output file.

After processing the request, the backend returns a response. This response may be a rendered HTML page, a redirect, a JSON response used by asynchronous JavaScript functionality, or a downloadable file such as a generated document. This lifecycle allows the system to support both traditional page-based interaction and dynamic updates within the same architecture.

4.2.4 Supporting Technologies

Several supporting technologies are integrated into the system architecture to provide the required functionality.

JavaScript is used on the client side to support dynamic interaction within the forms and application pages. It enables interface behaviour that does not require a full page reload, such as tab handling, validation feedback, upload interactions, and map-related updates.

Asynchronous requests, using AJAX or the Fetch API, are used where the frontend must communicate with the backend without reloading the page. This is particularly useful for calculation updates, document upload behaviour, and map-related processing.

Leaflet is used for geospatial visualisation. It supports the interactive map component, allowing the user to define the operational area and view spatial information relevant to the intended UAS operation.

The document generation functionality uses a template-driven approach. Application data stored in the system is inserted into predefined document templates, allowing the platform to generate structured DOCX application documents. This reduces manual repetition and improves consistency between the completed application data and the generated documentation.

Selenium is used where automated rendering or capture of dynamic map content is required. This supports the inclusion of visual map-related outputs in generated documentation when dynamic browser-rendered content must be captured.

Together, these technologies support the system’s main design goals: structured data entry, dynamic user interaction, map-based operational area definition, automated calculation support, document generation, and administrative review.

4.2.5 Architecture Diagram

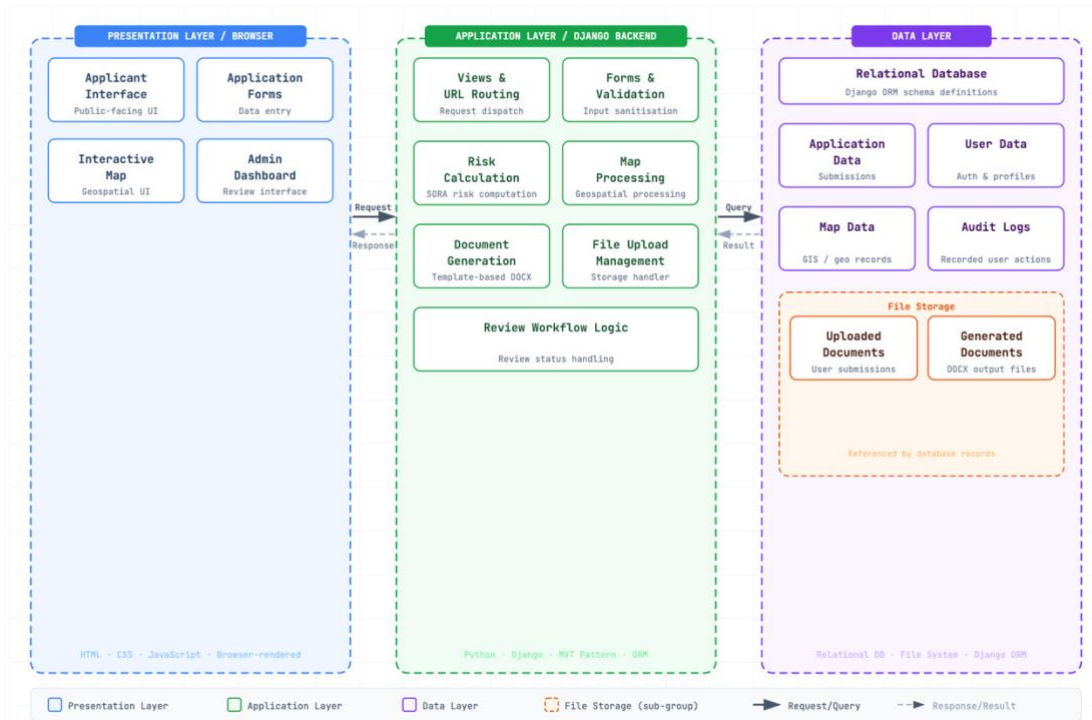


Figure 4.1: High-level system architecture of the SORA application platform.

4.3 User Interface Design

The user interface is designed to guide users through the SORA application process in a structured and sequential manner. Since the SORA workflow requires collecting various categories of information, the interface follows a guided, form-based design. This allows users to complete the process incrementally rather than interacting with the system as a single, unstructured form.

The interface design supports both applicant-facing and staff-facing workflows. Applicants require a clean, guided environment to prepare and submit SORA applications, while Civil Aviation Authority staff require a review-oriented interface to access submitted applications, view documents, and update review decisions.

4.3.1 Interface Design Objectives

The primary objective of the user interface design is to reduce the complexity of the SORA application process. For this reason, the interface is designed to organise the application process into logical sections and to present only the relevant information at each stage.

The main design objectives are to support usability, reduce cognitive load, improve clarity, and guide users through the correct sequence of actions. The interface also aims to reduce the likelihood of incomplete submissions by grouping related fields together and supporting validation of required information. In this way, the interface contributes directly to the functional and non-functional requirements defined in Chapter 3.

4.3.2 Form-Based Workflow Design

The system uses a form-based interaction model divided into sections or tabs. This structure allows the applicant to complete the application step by step. Each section corresponds to a specific part of the SORA workflow.

This tab-based design supports progressive completion. Users are not required to complete the entire application at once. Instead, the interface breaks the workflow into smaller, more manageable parts, achieving readability and supporting data entry accuracy.

The SORA 2.0 and 2.5 workflows are presented separately due to their different detail and assessment requirements. The interface, therefore, adapts to the selected version while keeping a consistent layout and style.

4.3.3 Map-Based Interface Design

A key part of the interface is the integration of an interactive map for defining the operational area. The application form is designed around a split layout, where the form input area and the map area are presented as related components within the same workflow.

The map interface allows the applicant to visually define the intended operational area and consider relevant geographical information during the application process. This is important because the SORA process depends not only on textual and technical data, but also on the spatial context of the proposed UAS operation. The map-based design, therefore, improves the usability and accuracy of operational area definition.

The map component also supports the presentation of supporting geographic information, including population density and no-fly or restricted areas. This helps applicants evaluate the suitability of the selected area and supports the risk assessment process.

4.3.4 Main User Interface Pages

The system interface consists of several main pages, each supporting a different part of the workflow. The login page provides access to the system and directs users to the appropriate area depending on their role. The home or landing page provides entry into the applicant workflow. The “My Applications” page allows applicants to view their draft and completed applications, generate documents, upload signed files, monitor review statuses, and delete applications when required.

The application form pages support the completion of SORA 2.0 and SORA 2.5 applications. These pages contain the structured form sections and the map-based operational area interface. The read-only application detail view allows completed applications to be viewed without editing, which supports both applicant review and administrative inspection. The file upload interface allows applicants to submit signed PDF documents and, where applicable, additional SORA 2.5 document packages.

The administrative dashboard is designed for Civil Aviation Authority staff. It provides access to submitted applications and supporting documents, allowing reviewers to inspect submissions and update review statuses. The dashboard supports filtering, sorting, viewing application materials, and recording rejection reasons when a submission is rejected.

4.3.5 Visual Consistency and Localisation

The interface design uses a consistent visual structure across the platform. Common layout elements, navigation areas, buttons, tables, forms, and feedback messages are reused throughout the system to provide a predictable user experience.

The system uses Bootstrap alongside the gov.cy design system to support a structured, government-oriented visual style. This is appropriate for a platform intended to support a Cypriot regulatory application process, as it gives the interface a formal and familiar public-service appearance. The use of consistent design components also improves readability and helps maintain a professional interface across applicant and administrative pages.

The interface supports both English and Greek, aligning with the system's Cypriot operational and regulatory context. This bilingual approach enhances accessibility and usability for its target user groups.

The user interface design supports the platform's main goals by combining guided form completion, continuous access to spatial information, clear application management, and structured administrative review.

4.4 Database Design

The database design provides the persistent data structure required to support the SORA application platform. It stores user information, SORA application data, drone specifications, risk assessment inputs and outputs, map-based operational area data, submitted documents, review statuses, and audit records. The database therefore supports both the applicant workflow and the Civil Aviation Authority review workflow.

The design follows an application-centred structure. Each SORA application is represented by a central application entity, with related information stored in separate models linked to that application. This improves maintainability and allows each part of the SORA process to be handled independently.

4.4.1 Database Design Objectives

The main objective of the database design is to organise the information required by the SORA workflow in a structured and maintainable way. The database must enable application creation, draft saving, completion, deletion, expiry tracking, document generation, upload, staff review, and audit logging.

Its design should meet several key objectives. Firstly, it needs to retain application data, allowing applicants to revisit and complete incomplete drafts. Secondly, it should store data necessary for risk assessment and document creation. Thirdly, it must accommodate both SORA 2.0 and SORA 2.5 versions without forcing all fields into a single, generic application form. Fourth, it should facilitate traceability by capturing statuses of submitted documents, rejection reasons, and audit log entries. Lastly, the structure should support public-facing application IDs without revealing internal database IDs.

4.4.2 Core Application Entities

The core entities of the database are the User, Application, and Application25 models. The User model represents authenticated users of the system, including applicants and staff users. Each user may create multiple SORA applications.

The Application model represents a SORA 2.0 application. It stores the application owner, URL token, application status, creation and update timestamps, and links to the related SORA 2.0 assessment data. These linked data entities include pre-evaluation data, drone specifications, ground risk assessment data, air risk assessment data, TMPR data, map data, and strategic mitigation data.

The Application25 model represents a SORA 2.5 application. It follows the same general application-centred design but includes links to the assessment components required by the SORA 2.5 workflow. These include pre-evaluation data, drone specifications, SORA 2.5 ground risk assessment data, SORA 2.5 air risk assessment data, TMPR data, containment data, map data, strategic mitigation data, population density, and the operation manual PDF.

Both application models include a status field that supports the lifecycle states of the system: draft, completed, and expired. These states allow the platform to distinguish between applications that are still being prepared, applications that have been completed, and applications that are no longer valid in relation to the planned operation date.

The application models also include non-sequential public-facing identifiers. This design prevents users from relying on or exposing internal database primary keys when viewing or managing applications. The use of URL tokens and hashed identifiers supports a clearer separation between internal database records and externally visible application references.

4.4.3 SORA 2.0 Data Entities

The SORA 2.0 application data is divided into several related models. The PreEvaluation model stores the initial application information, including the application title, operation date and time, organisation, applicant contact details, operational description, and pre-evaluation responses. This model provides the basic contextual information required before the risk assessment process proceeds.

The DroneSpecification model stores the technical characteristics of the UAS. These include flight height, maximum take-off mass, maximum velocity, maximum wind speed, cross-sectional area, characteristic dimensions, parachute information, deployment time, terminal velocity, and kinetic energy. These values are important because they are used in the risk calculation process and in document generation.

The RiskAssessment model stores ground risk-related data for SORA 2.0. It includes the operational scenario, ground risk class, mitigation selections, validation levels, mitigation results, and the final ground risk class after mitigations. This allows the system to preserve both the original ground risk classification and the result after mitigation measures have been applied.

The ArcAssessment model stores the air risk assessment data for SORA 2.0. This includes the initial ARC, final ARC, and final SAIL. Separating air risk data from ground risk data

improves the organisation of the database and reflects the distinction between ground risk and air risk within the SORA process.

The TmprAssessment model stores Tactical Mitigation Performance Requirement information. This includes the operation type and the applicant's declarations for the detect, decide, command, execute, feedback loop, and assurance elements. Including this model in the SORA 2.0 application structure allows TMPR-related information to be stored separately from the air risk assessment values while remaining linked to the same application.

The StrategicMitigation model stores mitigation fields and mitigation rationales. Since an application may contain multiple strategic mitigation entries, these records are connected to Application through a many-to-many relationship. This avoids storing a fixed number of mitigation fields directly inside the main Application model and allows the number of mitigation records to vary between applications.

4.4.4 SORA 2.5 Data Entities

The SORA 2.5 workflow requires additional database entities because the methodology contains more detailed assessment steps and outputs. The Application25 model therefore links to several SORA 2.5-specific assessment models while also reusing shared entities where appropriate.

The RiskAssessment25 model stores the SORA 2.5 ground risk mitigation selections. It includes fields for M1A sheltering, M1B operational restrictions, M1C ground observation, and M2 reduction of impact effects. It also stores the calculated ground risk class and the ground risk class after mitigations. This model separates SORA 2.5 ground risk data from the SORA 2.0 ground risk model because the mitigation structure differs between the two methodologies.

The ArcAssessment25 model stores the air risk assessment values for SORA 2.5. These include the initial ARC, adjacent airspace ARC, final ARC, and final SAIL. This separation allows SORA 2.5-specific air risk information to be stored independently from the SORA 2.0 air risk model.

The TmprAssessment model is reused in the SORA 2.5 workflow to store Tactical Mitigation Performance Requirement information. It includes the operation type and the detect, decide, command, execute, feedback loop, and assurance declarations. Reusing this model avoids duplicating the same TMPR data structure across the two SORA versions.

The Containment25 model stores containment-related information specific to SORA 2.5. It includes the outdoor assemblies category and the calculated containment level. Separating containment information into its own model improves clarity because containment is a distinct SORA 2.5 assessment output.

The Application25 model also stores SORA 2.5-specific application-level data, including the confirmation place, population density, map polygon reference, and operation manual PDF. This design keeps SORA 2.5-specific outputs and documents associated with the SORA 2.5 application record while preserving separation between the main application entity and its detailed assessment components.

4.4.5 Supporting Entities

Several supporting entities are used across the system. The DroneModel model stores predefined drone model data, including the model name, maximum take-off mass, cross-sectional area, maximum velocity, maximum wind speed, and characteristic dimensions. This supports the drone specification auto-fill functionality.

DroneModel is represented as a supporting lookup entity rather than as a directly connected relationship in the ER diagrams. It is used to populate drone specification fields during form completion, but no direct foreign-key relationship is stored between DroneModel and DroneSpecification. The selected drone specification values are stored inside DroneSpecification. This ensures that later changes to the drone model database do not affect previously completed applications.

The MapPolygon model stores the operational area as GeoJSON. This allows the system to preserve the user-defined operational area and use it for map visualisation, population-

density analysis, restricted-area checks, buffer calculations, and generated documentation.

The SubmittedDocument model stores uploaded documents associated with applications. It can be linked either to a SORA 2.0 application or to a SORA 2.5 application. It stores the uploaded file, document type, upload timestamp, uploading user, review status, and rejection reason. The review status supports the document lifecycle states of pending, accepted, and rejected.

The AuditLog model records key user actions. It stores the user, action, timestamp, and optional details. This supports traceability and accountability by allowing important system events to be recorded, such as login, logout, application deletion, document submission, and review-related actions.

4.4.6 Relationships and Design Rationale

The database design is based on a central application relationship. A user may have multiple SORA 2.0 and SORA 2.5 applications. Each application is then linked to the data entities required by its version of the SORA workflow.

For SORA 2.0, an Application is linked to pre-evaluation data, drone specifications, ground risk assessment data, air risk assessment data, TMPR data, strategic mitigations, map data, submitted documents, and audit-related user actions. Most of these relationships are one-to-one because each application has one main set of pre-evaluation data, one drone specification record, one ground risk assessment record, one air risk assessment record, one TMPR record, and one map polygon. Strategic mitigations are represented through a many-to-many relationship because an application can contain multiple mitigation entries.

For SORA 2.5, an Application25 is linked to pre-evaluation data, drone specifications, SORA 2.5 ground risk assessment data, SORA 2.5 air risk assessment data, TMPR data, containment data, strategic mitigations, map data, operation manual files, submitted documents, and audit-related user actions. The design separates containment information

into a dedicated Containment25 model, while TMPR information is stored through the shared TmprAssessment model.

The separation between Application and Application25 is a deliberate design decision. Although the two models share common application-level fields and some shared supporting entities, the SORA 2.5 workflow requires additional assessment structures and outputs, including SORA 2.5-specific risk assessment and containment data. Keeping the two application models separate makes the version-specific workflows clearer and avoids forcing all fields into one large generic application model.

The design also separates application data from submitted review documents. Application data stores the information entered by the applicant and used for calculation and document generation, while SubmittedDocument stores files uploaded for review. This distinction supports the review workflow because submitted documents require their own review status, upload timestamp, document type, and rejection reason.

Overall, the database design supports maintainability, traceability, and explicit separation between SORA 2.0 and SORA 2.5 workflows. It allows the system to store complex application data while keeping the main parts of the SORA process organised into related models.

4.4.7 Entity-Relationship Diagram

The database design is represented through two entity-relationship diagrams. Figure 4.2 presents the SORA 2.0 database structure, while Figure 4.3 presents the SORA 2.5 database structure. The diagrams are separated for readability because the full database schema contains many related entities and would be difficult to interpret as a single figure.

Figure 4.2 shows Application as the central entity for the SORA 2.0 workflow. It is linked to the user, pre-evaluation data, drone specifications, ground risk assessment data, air risk assessment data, TMPR data, map data, strategic mitigation records, submitted documents, and audit logging.

Figure 4.3 shows Application25 as the central entity for the SORA 2.5 workflow. It is linked to the user, pre-evaluation data, drone specifications, SORA 2.5 ground risk assessment data, SORA 2.5 air risk assessment data, TMPR data, containment data, map data, strategic mitigation records, submitted documents, and audit logging.

Both diagrams also include DroneModel as a supporting lookup entity. DroneModel is not connected to DroneSpecification through a foreign-key relationship because the selected drone values are copied into the DroneSpecification record during form completion. This design preserves the historical accuracy of completed applications even if the predefined drone model data is later updated.



Figure 4.2: Entity-relationship diagram for the SORA 2.0 application workflow

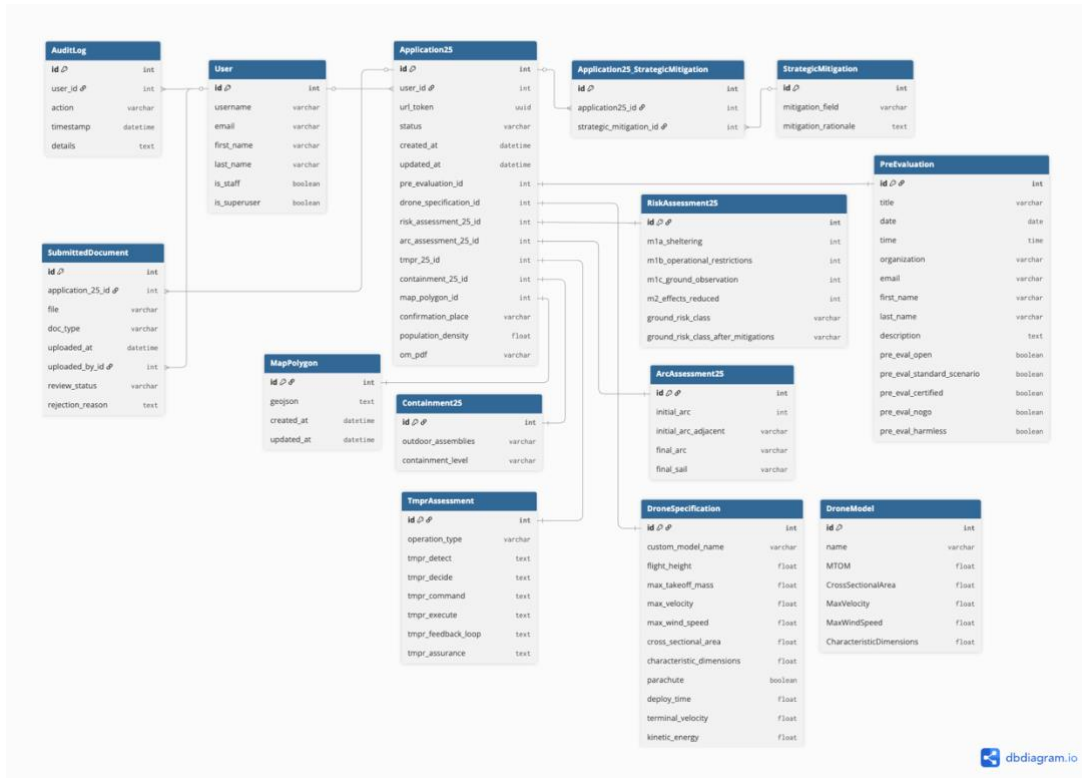


Figure 4.3: Entity-relationship diagram for the SORA 2.5 application workflow.

4.5 Backend Design

The backend implements the application and business logic layer of the SORA application platform. It is built using Django views, forms, models, and utility functions. Views handle HTTP requests, while forms validate and structure user input before it is stored. Utility functions handle calculations, document generation, and supporting operations that should remain separate from the presentation layer. This separation improves maintainability by allowing different logic to be modified independently.

Each SORA version has its own set of views for creating, editing, viewing, deleting, and generating documents. This reflects the design decision already discussed in the database section, where SORA 2.0 and SORA 2.5 are represented by separate application models.

The backend supports two different application states: draft and completed. Draft saving allows applicants to store incomplete application data and return to it later, while

completion applies stricter validation before the application is considered ready for document generation and submission.

The backend design also reflects the separation between SORA 2.0 and SORA 2.5. SORA 2.0 is retained as a supported legacy and comparative workflow, while SORA 2.5 is treated as the primary current workflow. Both versions share common backend responsibilities, including application creation, draft saving, drone specification handling, map polygon storage, document generation, and signed document upload. However, SORA 2.5 requires additional backend support for population-density-based ground risk assessment, containment outputs, operation manual handling, and multiple document outputs.

Risk calculation logic is designed as reusable backend utility logic rather than as template-level logic. This ensures that calculated values remain consistent across form display, application completion, stored records, generated documents, and map-related outputs. In the SORA 2.5 workflow, this is especially important because ground risk, SAIL, and containment depend on several connected inputs, including population density, characteristic dimension, maximum speed, mitigations, final ARC, adjacent area data, and outdoor assembly information.

The map-related backend design supports the storage and later reuse of the operational area. The applicant defines the operational area through the frontend map, while the backend stores the resulting polygon as GeoJSON. This allows the same spatial data to be reused for application viewing, risk-related processing, buffer representation, and generated documentation. The detailed implementation of the interactive map and its validation behaviour is discussed in Chapter 5.

The document and file-management design separates generated documents from submitted review documents. Generated documents are produced from completed application data, while submitted documents are uploaded by the applicant after external review and signing. For SORA 2.5, the operation manual is handled separately because it forms part of the application content, whereas uploaded signed documents belong to the review workflow.

The administrative review workflow is designed around submitted documents. Staff users can access submitted material, inspect the completed application data and uploaded files, and update the review status. If a submission is rejected, the rejection reason is stored so that the applicant can view the feedback. This supports traceability and keeps review decisions connected to the submitted material.

4.6 Security and Privacy Design

The system's security and privacy design focuses on protecting users' private data, as the platform manages all user information. Therefore, access to system resources must be regulated according to user roles and ownership.

4.6.1 Authentication and Role-Based Access Control

The system requires users to authenticate before accessing the main application platform. Authentication separates public access from protected system functionality. After logging in, users are directed to the appropriate section of the system based on their role. Regular applicants are directed to the applicant-facing workflow, while staff or administrative users are directed to the administrative dashboard.

Role-based access control distinguishes between applicants and staff users. Applicants can create, edit, view, delete, generate, and submit their own SORA applications. Staff users have access to the review interface, where they can inspect submitted applications and update document review statuses. This separation ensures that administrative functions are not exposed to ordinary applicant accounts.

The prototype uses controlled local account management rather than public self-registration. User accounts are created by authorised CAA staff through User Management. Applicants therefore access the platform using accounts created by the authority-side user rather than registering independently.

4.6.2 Application-Level Access Restrictions

Application-level access control is a crucial component of the system design. Each SORA application is linked to its creator. When a user tries to perform any action on an application, the backend verifies that the application belongs to the authenticated user.

For staff users, access is intentionally broader because the review workflow requires them to view submitted applications and documents. However, this access is limited to users with staff or administrative privileges.

4.6.3 Document and File Protection

Uploaded documents are associated with the relevant application and the uploading user. This design ensures that submitted files are not stored as unrelated files but are linked to the corresponding SORA application and review workflow. The system distinguishes between general submitted documents, SORA 2.5 document packages, and operation manual files.

The file management design also supports tracking review status. Each submitted document can be marked as pending, accepted, or rejected. If a document is rejected, the reason for rejection is stored with the document. This preserves review decisions and links them to the relevant uploaded file.

The system also limits document upload behaviour to align with the intended workflow. Applicants upload signed PDFs for review, while SORA 2.5 can also include additional document packages where required. This supports a controlled file-submission process and reduces ambiguity among generated documents, uploaded signed documents, and reviewable submission files.

4.6.4 Public Identifiers and Audit Logging

The system uses non-sequential public-facing identifiers to reduce exposure of internal database identifiers. This is important because internal primary keys are predictable and should not be relied upon as public references in user-facing workflows. By using hashed identifiers and URL tokens, the system separates internal database records from the identifiers shown to users.

Audit logging is included to support traceability and accountability. Key actions, such as login, logout, application deletion, document submission, and review-related actions, can be recorded with the responsible user, timestamp, and action details. This is important in a regulatory application context because administrative and applicant actions should be traceable where necessary.

Audit logs do not replace access control, but they provide an additional layer of accountability. They allow important system events to be reviewed if there is a need to investigate user activity or review workflow decisions.

4.6.5 Limitations and Future Improvements

The implemented security and privacy mechanisms provide a foundation for controlled access and traceability. However, additional measures would be required for a production deployment.

Future improvements could also include malware scanning for uploaded files, file size limits, stronger audit reporting, and more advanced monitoring of administrative actions. These measures would further strengthen the platform if it were deployed in a live regulatory environment.

The prototype does not integrate with CY Login or the GovCY Digital Services Factory authentication infrastructure. In a production deployment, local account management should be replaced or extended with official identity-provider integration.

In conclusion, the security and privacy design supports the system's main protection goals: safeguarding applicant data, protecting operational information, restricting access to authorised users, securely preserving uploaded documents, and maintaining traceability of important system actions.

Chapter 5

Implementation

5.1 Development Environment and Technologies	59
5.2 Database Models Implementation	62
5.3 Backend Implementation	63
5.4 Applicant Entry Point and Application Management Dashboard	67
5.5 Shared Application Components	70
5.6 SORA 2.0 Implementation	77
5.7 SORA 2.5 Implementation	81
5.8 Validation and Completion Feedback	93
5.9 Document Generation and Submission Workflow	97
5.10 Administrative Dashboard and Review Workflow	101
5.11 Internationalisation Implementation	104

This chapter details how the system design from Chapter 4 was implemented. It covers the technologies used, database models, backend logic, user interface, SORA workflows, submission process, and bilingual support. The focus is on the practical development of the web application built with the Django framework. The implementation follows the layered architecture described in Chapter 4, maintaining a clear separation between presentation, application logic, and data management.

5.1 Development Environment and Technologies

The implementation follows the layered design described in Chapter 4, where the presentation layer is handled through HTML templates, CSS, JavaScript, and browser-based interaction; the application layer is implemented using Django views, forms, and utility functions; and the data layer is represented through Django models, database storage, and file storage.

The main development language used for the backend was Python. Python was suitable for this system because it is directly supported by Django and provides access to libraries for document generation, geospatial processing, and automation tasks. Django was used as the backend framework because it provides built-in support for URL routing, views, templates, forms, models, authentication, file handling, and internationalisation.

During development, SQLite was used as the database. SQLite is appropriate for development and prototyping environments because it is lightweight, easy to configure, and integrates directly with Django. It allowed the database models to be developed and tested without requiring a separate database server. For a production deployment, however, a more robust database configuration would be required, as discussed in the security and future-improvement sections.

The frontend was implemented using HTML, CSS, and JavaScript. Django templates were used to render dynamic pages, while JavaScript was used to support client-side interaction, asynchronous updates, validation feedback, map interaction, upload behaviour, and administrative status updates. The base template includes Bootstrap, Bootstrap Icons, Font Awesome, the gov.cy design system, and the project stylesheet, forming the interface's common visual and layout foundation.

The interface styling combines Bootstrap with the gov.cy design system. This combination was selected because Bootstrap provides reusable layout and interface components, while the gov.cy design system gives the platform a visual style suitable for a Cypriot public-service context. All templates use gov.cy interface components, including the gov.cy header, service name, form styling, and footer structure.

Leaflet.js was used to implement the interactive map component. The map interface allows applicants to define the operational area visually and supports the display of geospatial information relevant to the SORA process. The map template imports Leaflet and related geospatial libraries, including Turf.js, and passes backend URLs, static GeoJSON paths, CSRF data, existing polygon data, and completed application data into the JavaScript map module. The JavaScript map module then manages drawn points,

markers, polygon data, population-density data, restricted-zone information, and map validation feedback.

The document generation functionality was implemented using python-docx. The system uses DOCX templates containing placeholders, and the backend replaces those placeholders with data stored in the application models. The utility module includes a DocxGenerator class responsible for generating DOCX files from application data and inserting calculated values, applicant information, risk assessment information, OSO data, and map-related content.

Selenium and ChromeDriver were used to capture dynamic map content for generated documents. Since the map is rendered in the browser using JavaScript, it cannot be inserted into a DOCX document directly from static backend data alone. The backend, therefore, uses Selenium in headless mode to open the application detail page, wait for map overlays to render, capture the map container, and encode the screenshot for later insertion into the generated document.

Django's authentication system was used to manage login, logout, and user-based access control. The backend includes login and logout views, role checks, and redirection logic that distinguishes between regular applicants and staff or administrative users. This supports the separation between the applicant workflow and the administrative review workflow.

Django email functionality was used for system notifications. The upload workflow includes email notification logic for submitted documents. Email functionality was also used for review-status notifications. When a submitted document is accepted or rejected, the backend sends an email notification to the applicant through an SMTP backend. Email delivery failure does not block the review-status update.

Django internationalisation was used to support English and Greek. User-facing template strings are marked for translation, and shared JavaScript translation bundles are used to expose translated interface messages to frontend scripts. The `_js_i18n.html` template provides translated strings for the application form, map module, and administrative

interface, allowing JavaScript-driven feedback to remain consistent with the selected interface language.

Consequently, the selected technologies were suitable because they directly met the platform's functional requirements. Together, these technologies enabled a structured digital workflow for SORA application preparation, submission, and review.

5.2 Database Models Implementation

The database layer was implemented using Django models. Each model is defined as a Python class and translated by Django's Object-Relational Mapping mechanism into database tables. This implementation follows the database design described in Section 4.4, in which the system uses an application-centred structure rather than storing all application data in a single large table.

The two main application records are implemented through the Application model for SORA 2.0 and the Application25 model for SORA 2.5. Since SORA 2.5 has additional steps, the Application25 model includes additional implementation fields and relationships required by that version, including SORA 2.5 risk assessment data, containment data, population density, confirmation place, and the operation manual PDF.

Related application data is stored in separate models rather than directly inside the main application models. Shared data structures include PreEvaluation, DroneSpecification, TmprAssessment, StrategicMitigation, MapPolygon, SubmittedDocument, and AuditLog. Version-specific data is stored in separate models, such as RiskAssessment and ArcAssessment for SORA 2.0, and RiskAssessment25, ArcAssessment25, and Containment25 for SORA 2.5.

Most application-related models are linked through one-to-one relationships because each application has one of each main record. Strategic mitigations, however, are implemented through a many-to-many relationship because an application may contain multiple mitigation entries.

Both Application and Application25 include a UUID-based url_token and a hashed_id property. These identifiers allow the platform to display and use non-sequential public references instead of exposing internal database primary keys directly.

File-related data is implemented through separate fields and models depending on the file's purpose. The operation manual is stored directly on the Application25 model because it forms part of the application content. In contrast, signed documents uploaded for review are stored in the SubmittedDocument model because they require a review status, upload timestamp, document type, the uploading user, and a possible rejection reason.

The model implementation also supports lifecycle and traceability requirements. Application models include status values such as draft, completed, and expired, while submitted documents include review statuses such as pending, accepted, and rejected. The AuditLog model records important user actions with the responsible user, timestamp, and action details, supporting accountability in the application and review process.

5.3 Backend Implementation

The backend implementation provides the main application logic of the SORA application platform. It was implemented using Django URL routes, views, forms, models, and utility functions. These components process requests, validate submitted data, update database records, expose calculation endpoints, manage files, and support the administrative review workflow.

The backend is organised around the system's core workflows. These include authentication, applicant application management, form processing, map interaction, calculation services, document generation, file upload, document retrieval, and administrative review.

5.3.1 URL Routing and Endpoint Structure

The URL configuration defines the system's backend entry points. These routes map actions performed in the interface to the relevant Django views. The routes are organised by the major system functions.

Table 5.1: Main backend route groups and their implementation purpose

Endpoint group		Example routes	Purpose
Authentication and navigation		/, /home/, /logout/	Handles login, applicant home access, logout, and role-based navigation.
Applicant dashboard		/my_applications/	Displays the applicant's draft, completed, submitted, and expired applications.
SORA 2.0 application workflow	2.0	/application/new/, /application/<pk>/edit/, /application/<pk>/view/, /application/<pk>/delete/	Supports creation, editing, read-only viewing, and deletion of SORA 2.0 applications.
SORA 2.5 application workflow	2.5	/application25/new/, /application25/<pk>/edit/, /application25/<pk>/view/, /application25/<pk>/delete/	Supports creation, editing, read-only viewing, and deletion of SORA 2.5 applications.
Document generation and retrieval		/application/<pk>/generate-docx/, /application25/<pk>/generate-docx/, /application25/<app_id>/download-zip/, /submitted_documents/<pk>/	Generates application documents, downloads SORA 2.5 ZIP outputs, and retrieves submitted files.

Document upload and deletion		/document/upload/, /document/<pk>/delete/, /application25/<pk>/om-pdf/delete/, /application25/om-pdf/pending/delete/	Handles signed PDF upload, submitted document deletion, and SORA 2.5 operation manual PDF deletion.
Drone lookup	model	/api/drone-model/<pk>/	Retrieves stored drone model data for form autofill.
Map endpoints	support	/get_polygon/, /process_polygon/, /clear_markers/, /enable_button/, /reset_map/, /reserved_zones/	Supports polygon processing, marker clearing, map reset behaviour, reserved-zone retrieval, and map state management.
Calculation endpoints		/get_scv/, /get_sgrb/, /get_grc_25/, /get_containment_25/ /	Returns calculated values such as lateral containment volume, ground risk buffer, SORA 2.5 GRC, SAIL, and containment-related outputs.
Administrative review		/dashboard/, /document/<doc_id>/update-status/	Allows authorised staff to review submitted documents and update review statuses.
User management		/admin/	Allows authorised CAA staff to manage local user accounts through Django administration.

Using separate route groups makes the backend easier to maintain. The chosen organisation reflects the layered design described in Chapter 4, where interface actions are passed to backend views and then connected to model storage, utility logic, file handling, or rendered responses.

5.3.2 View and Form Processing

Django views coordinate the interaction between templates, forms, database models, utility functions, and client-side JavaScript. When a user performs an action, such as creating an application, saving a draft, completing an application, uploading a document, deleting an application, or updating a review status, the corresponding view receives the request and determines the appropriate response. Depending on the operation, the backend may return a rendered page, redirect the user, return a downloadable document, or return a JSON response for asynchronous interface updates.

Separate backend views are used for the SORA 2.0 and SORA 2.5 workflows. This separation is necessary because the two methodologies require different form sections, database relationships, calculation logic, and document outputs. However, both workflows follow the same general structure: the applicant creates or edits an application, enters the required assessment data, defines the operational area, saves the application as a draft or completes it, and later generates and uploads the required documentation.

The form layer connects user input with the database models. Django ModelForm classes are used for the different application sections, defining the relevant fields, widgets, choices, and validation rules. The forms also support draft saving by allowing incomplete data to be stored before final completion checks are applied.

The pre-evaluation form defines inputs for date, time, organisation, contact, and operation description. It also applies date and email validation, including a minimum operation date based on the Cyprus time zone. The drone specification form includes a selectable drone model for autofill, but the selected model is not stored directly in the DroneSpecification model. Instead, the predefined model populates the form fields, while the selected values are stored in the application's own drone specification record. This prevents later changes to the drone model database from altering previously completed applications.

Applicant-facing views are protected by authentication and ownership checks, so users can access only their own applications. Staff-facing views are restricted to authorised users and are used for submitted-document review, supporting-file access, and review status updates.

5.3.3 Shared Backend Services and Utility Logic

In addition to full-page views, the backend provides shared service endpoints that the interface uses during application completion. These endpoints allow the frontend to retrieve or calculate values without reloading the full page and are necessary because several parts of the interface update dynamically as the applicant completes the application.

The calculation endpoints return values used during form completion, including lateral containment volume, ground risk buffer, SORA 2.5 GRC, final GRC, SAIL, and containment-related outputs. The detailed calculation logic is discussed in Sections 5.5.3, 5.6.2, and 5.7.2.

The map-related endpoints support the storage and reuse of operational area data. The operational area is submitted as GeoJSON and stored with the application, allowing it to be loaded again during editing, completed application viewing, and document generation.

Shared backend utilities are also used for document generation, file handling, and audit logging. These utilities keep reusable logic outside templates and reduce duplication across the SORA 2.0, SORA 2.5, applicant, and staff workflows.

5.4 Applicant Entry Point and Application Management Dashboard

5.4.1 Applicant Home Page

The applicant's home page is the first instructional page displayed to regular users after login. It introduces the purpose of the SORA application platform and outlines the main steps the applicant must follow to complete the process.

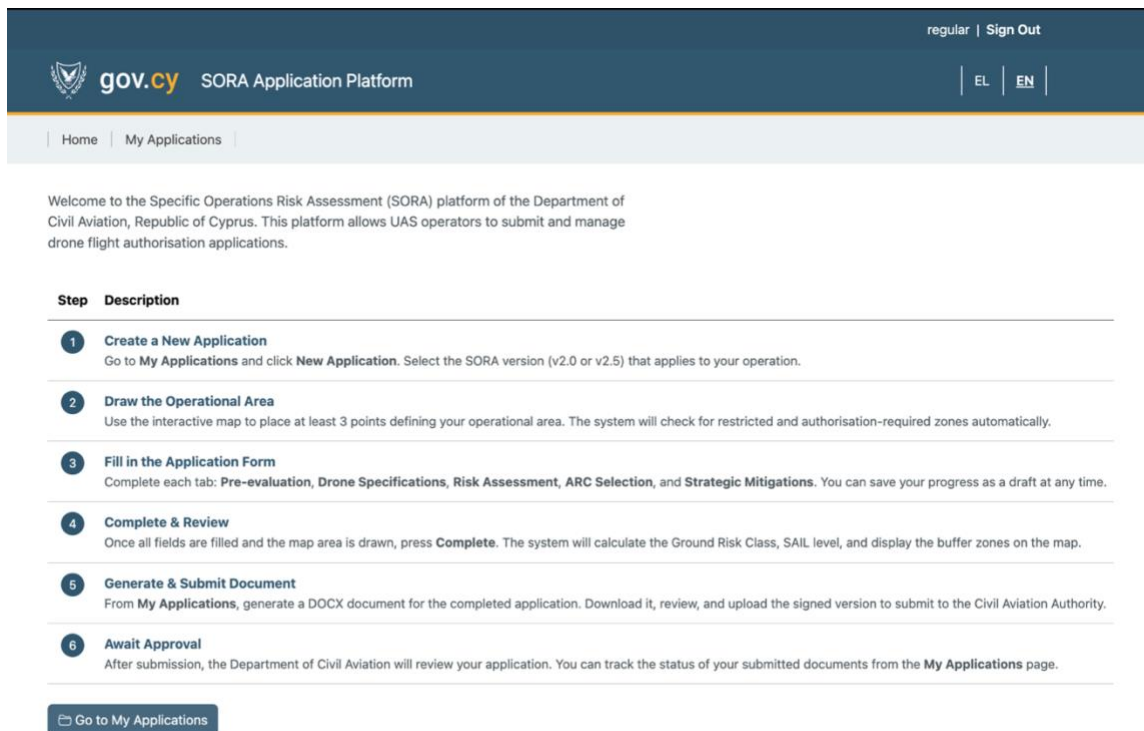


Figure 5.1: Applicant home page with workflow instructions.

The page is rendered through the shared base layout, which includes the gov.cy header, service name, navigation bar, logout control, and language selector. The navigation bar provides access to the home page and the My Applications page. For authorised users, it also displays access to the administrative dashboard.

5.4.2 My Applications Dashboard

The applicant dashboard is implemented through the My Applications page. This page acts as the main management area for applicant-owned SORA applications. From this page, applicants can create new applications, continue drafts, view completed applications, generate documents, upload signed files, monitor review status, and delete applications or uploaded documents.

The page includes a sticky action bar with the page title and a New Application button. Selecting this button opens a version-selection modal, where the applicant chooses whether to create a SORA 2.0 or SORA 2.5 application.

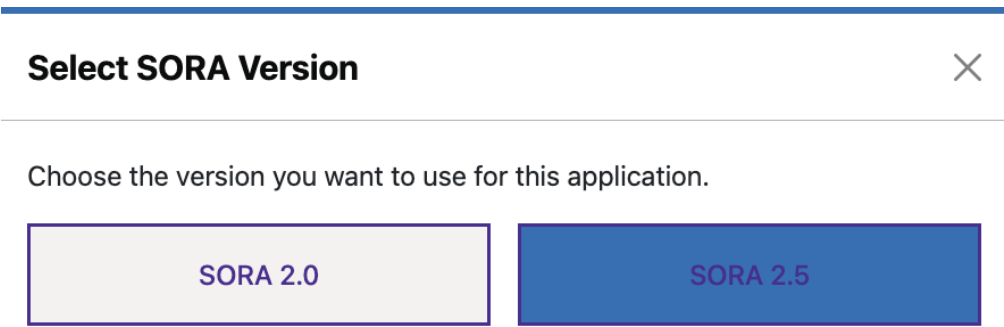


Figure 5.2: Version selection modal for creating a new SORA application.

The dashboard is organised into application tables grouped by lifecycle state and version. Draft applications are displayed separately from completed SORA 2.0 and SORA 2.5 applications. Each row presents the public application identifier, title, status, creation date, and available actions.

My Applications

+ New Application

▼ Draft Applications

ID	Title	Status	Created	Actions
#181291	Application #181291	DRAFT	May 20, 2026	Edit

▼ Completed Applications (SORA 2.5)

ID	Title	Status	Created	Actions	Uploaded	Review	Document
#290278	title	COMPLETED	May 20, 2026	View Generate Upload	May 20, 2026 01:04	PENDING	PDF ZIP

▼ Completed Applications (SORA 2.0)

ID	Title	Status	Created	Actions	Uploaded	Review	Document
#259538	title	EXPIRED	Feb 09, 2026	View Generate Upload	Feb 09, 2026 12:33	EXPIRED	PDF

Figure 5.3: My Applications dashboard showing draft and completed application tables.

The backend view for this page prepares the application data shown to the applicant. It retrieves the user’s applications, identifies draft and completed records, retrieves submitted documents, and determines whether applications have expired based on their

planned operation date. This allows the dashboard to show not only the applications themselves but also their current validity and review status.

Draft applications can be edited or deleted. Expired drafts are marked or displayed separately so that the applicant can distinguish them from applications that may still be completed. Completed applications provide actions for read-only viewing, document generation, signed document upload, and deletion. For SORA 2.5, the dashboard also supports ZIP-related document handling where multiple generated outputs are involved.

The dashboard displays submitted-document status information, including pending, accepted, rejected, and expired states. When a rejection reason exists, the applicant can access the reviewer's feedback from the same page.

Client-side JavaScript supports dashboard interactions such as the version-selection modal, upload modal, and deletion confirmation prompts. These interactions avoid unnecessary intermediate pages while still requiring confirmation before destructive actions such as application or document deletion.

Document generation and signed upload are initiated from this dashboard, but their detailed implementation is discussed in Section 5.9.

5.5 Shared Application Components

Before examining the two versions separately, this section presents the implementation components that are shared by both versions. These shared components include the multi-step application form, the interactive map used for operational area definition, and the common UAS and operational volume calculations used to derive physical and buffer-related values.

5.5.1 Multi-Step Form Layout

The application form uses a split layout. The form is displayed on the left side of the page, while the interactive map is displayed on the right side.

The same general form layout is used for both SORA versions, but the visible tabs vary depending on the selected methodology. This allows the interface to preserve a consistent structure while rendering only the sections required by each version.

The screenshot displays the SORA Application Platform interface. At the top, there is a header with the gov.cy logo and navigation links: Home, My Applications, and Admin Dashboard. A user profile section shows 'user | Sign Out' and language options 'EL' and 'EN'. Below the header, a breadcrumb trail leads to 'Back to My Applications'. The main content area is divided into two columns. The left column contains a 'Pre-evaluation' tab and a series of checkboxes for reviewing statements before completing the application. The right column features a map of Cyprus with a red polygon indicating a restricted area. A legend in the bottom right corner of the map identifies 'Authorization Required' (red outline) and 'Restricted Area' (red fill). A 'Clear Map' button is located at the bottom of the map area.

gov.cy SORA Application Platform

user | Sign Out EL EN

Home My Applications Admin Dashboard

← Back to My Applications

Pre-evaluation Drone Specifications Ground Risk Air Risk TMPR & SAIL OSOs & Containment

Please review the following statements before completing this application. To proceed with a SORA assessment, all answers must be NO. A single YES means SORA does not apply to your operation.

- ☐ The Operation under the "Open" category:
- ☐ The Operation is covered by a "Standard Scenario" recognized by the competent authority:
- ☐ The Operation under the "Certified" category:
- ☐ The Operation is subject to a specific NO-GO from competent authority:
- ☐ The Competent Authority has determined that the UAS is "harmless" for the ground risk:

Title: *

Date: *

Time: *

Organization: *

Email: *

First Name: *

Last Name: *

Description: *

Click on the map to add points and define your operation area. Close the polygon by clicking the first point. Press Ctrl + Z to undo the last point, or use Clear Map below to remove the entire area.

Population Density

Legend

- Authorization Required
- Restricted Area

10 km 10 mi

Clear Map

Figure 5.4: Multi-step SORA application interface with split form and map layout.

The top section of the form includes a workflow action area with a navigation link back to the “My Applications” page and buttons to save the application as a draft or to complete it. Saving as a draft preserves incomplete progress, while completing the application triggers stricter checks and finalises the application state.

The form template includes hidden fields and data attributes that connect the interface to the backend and JavaScript logic. These include fields for polygon coordinates, population density, adjacent population density, and the active tab.

Additionally, the active tab is preserved during editing. The JavaScript form logic reads the stored active tab when the page loads and restores the corresponding tab. It also updates the hidden active-tab field whenever the user moves between tabs. This prevents the applicant from being returned to the first section after saving, reloading, or continuing the application later.

The form also supports a completed or read-only state. When an application has been completed, the form fields are disabled to prevent the submitted information from being accidentally modified. This allows the applicant or reviewer to inspect the completed application while preserving the integrity of the submitted data.

5.5.2 Interactive Map Component

The map allows the applicant to define the operational area by clicking points on the map and closing the polygon by selecting the first point. The resulting polygon is converted into GeoJSON and stored with the application.

The map component also includes revision controls. The applicant can undo the last point using keyboard interaction and can clear the entire map when the operational area must be redrawn. The map instructions inform users that they can add points, close the polygon, use Ctrl + Z to undo the last point, and use the clear-map button to remove the operational area.

The map integrates information on restricted or no-fly zones. Restricted zones are loaded from static spatial data and displayed as map layers. The map script checks whether selected points or generated areas intersect restricted zones. If the user attempts to place a point within a restricted zone or create an invalid operational area, the interface can prevent the action or provide feedback to the user. This supports safer operational area definition and reduces the risk of submitting with unsuitable flight geography.

Population-density data is also loaded as GeoJSON and can be displayed through the map overlay. This supports the SORA 2.5 workflow, where population density is used in the ground risk and containment calculations.

The map component is also connected to live calculation behaviour. When the applicant changes values such as characteristic dimension, maximum speed, operational scenario, mitigation selections, final ARC, or outdoor assembly information, the map and form scripts can trigger updates to risk-related outputs. The map therefore connects spatial input with backend calculation endpoints.



Figure 5.5: Interactive map component showing operational area definition and spatial overlays.

It also includes dynamic reserved-area awareness. When the applicant selects or edits the operation date, the frontend requests reserved-zone data from the backend. The backend returns the saved operation polygons of previously completed and accepted applications for that date as GeoJSON. These polygons are displayed on the Leaflet map as reserved zones and stored on the client side for validation. When the applicant draws a new operational area, the map checks the proposed polygon against both static restricted zones and dynamic reserved zones. If the polygon overlaps a reserved zone, it is treated as invalid and the applicant is prompted to select another location.

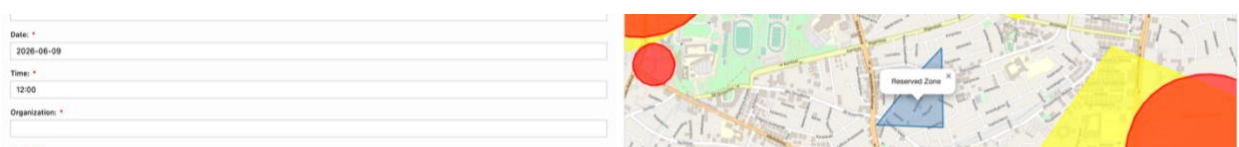


Figure 5.6: Interactive map showing a reserved operational area for the selected operation date.

After completion, the saved polygon and related spatial outputs can be displayed in the completed application view. The same rendered map is also used during document generation when a map image is inserted into the generated DOCX output, and is shown in the view format.

5.5.3 Shared UAS and Operational Volume Calculations

Both versions use a common set of UAS physical and operational volume calculations. These calculations are implemented in the backend utility layer and are used to derive values such as terminal velocity, kinetic energy, lateral containment volume, vertical containment volume, and ground risk buffer. In the implementation, SORA 2.0 uses the DroneValues class, while SORA 2.5 uses the DroneValues5 class. Although the two classes support different SORA-specific assessment logic, they share the same core equations for the UAS physical values and buffer-related outputs.

The first shared calculation is the UAS's terminal velocity. The system calculates terminal velocity as:

$$V_t = \sqrt{\left(\frac{2mg}{C_d \rho A}\right)}$$

Where V_t is the terminal velocity, m is the maximum take-off mass of the UAS, g is gravitational acceleration, C_d is the drag coefficient, ρ is air density, and A is the cross-sectional area of the UAS. In the implementation, the constants used are:

$$\begin{aligned} g &= 9.81m/s^2 \\ C_d &= 0.47 \\ \rho &= 1.2985 kg/m^3 \end{aligned}$$

Therefore, the implemented equation is:

$$V_t = \sqrt{\left(\frac{2 \cdot m \cdot 9.81}{0.47 \cdot 1.2985 \cdot A}\right)}$$

After terminal velocity is calculated, the system calculates kinetic energy as:

$$KE = \frac{1}{2}mV_t^2$$

where KE is the kinetic energy, and V_t is the calculated terminal velocity. These values are stored in the drone specification data and reused by the application view, the completed application record, and the generated documentation.

The second shared group of calculations concerns the operational volume and buffer-related values. The implementation uses fixed assumptions for horizontal GPS inaccuracy, position-holding error, map error, and vertical GPS inaccuracy:

$$S_{gps} = 3m$$

$$S_{pos} = 3m$$

$$S_k = 1m$$

$$H_{gps} = 4m$$

The response distance is calculated as:

$$S_{rz} = V_0 \cdot 1s$$

where V_0 is the maximum velocity of the UAS. Since the response time is one second, this becomes:

$$S_{rz} = V_0$$

The lateral containment volume is calculated as:

$$S_{cv} = S_{gps} + S_{pos} + S_k + S_{rz}$$

or using the implemented constants:

$$S_{cv} = 3 + 3 + 1 + V_0$$

If a parachute is selected, the system also calculates the lateral parachute manoeuvre distance:

$$S_{cm,par} = V_0 \cdot t_{par}$$

where t_{par} is the parachute deployment time. With parachute use, the lateral containment volume becomes:

$$S_{cv} = S_{gps} + S_{pos} + S_k + S_{rz} + S_{cm,par}$$

The vertical response height is calculated as:

$$H_{cv} = H_{fg} + H_{gps} + H_{rz}$$

where H_{fg} is the flight geography height entered by the applicant. In implemented form, this becomes:

$$H_{cv} = H_{fg} + 4 + (V_0 \cdot 0.7)$$

The ground risk buffer is calculated using the simplified 1:1 rule:

$$S_{grb} = V_0 \sqrt{\frac{2H_{cv}}{g}} + \frac{1}{2}CD$$

However, the default implemented calculation uses the 1:1 rule unless the alternative calculation method is selected in the utility logic. If a parachute is selected, the vertical parachute manoeuvre component is also added:

$$H_{cm,par} = V_0 \cdot t_{par} \cdot 0.7$$

Therefore, with parachute use, the implemented ground risk buffer becomes:

$$S_{grb} = H_{cv} + \frac{1}{2}CD + H_{cm,par}$$

These shared values support both SORA workflows. They are used to calculate drone-related outputs, generate map buffers, display completed application data, and populate generated documents. Although the UAS physical and buffer calculations are shared, the two SORA versions diverge after this stage. For this reason, the shared equations are presented in this section, while the version-specific risk calculation logic is discussed separately in Sections 5.6.2 and 5.7.2.

5.6 SORA 2.0 Implementation

The SORA 2.0 workflow is implemented as the legacy and comparative application flow supported by the platform. When the applicant selects SORA 2.0 from the version-selection modal, the system opens the SORA 2.0 form and stores the submitted data through the Application model and its related assessment records.

The SORA 2.0 workflow uses the shared components described in Section 5.5, including the multi-step form layout, map component, and shared UAS physical calculations. This section, therefore, focuses only on the elements specific to SORA 2.0.

5.6.1 SORA 2.0 Workflow Structure

The SORA 2.0 application workflow follows a guided form structure. In the current implementation, the SORA 2.0 form is organised into five visible tabs: Pre-evaluation, Drone Specifications, Ground Risk, Air Risk, and TMPR.



Figure 5.7: SORA 2.0 application form showing the tab-based workflow.

The first two tabs collect the shared application and UAS data used by both SORA versions. The version-specific workflow primarily begins on the Ground Risk and Air Risk tabs. In the Ground Risk tab, the applicant selects the operational scenario and enters the M1, M2, and M3 mitigation information. These inputs are used to calculate the initial GRC, mitigation reductions, and final GRC.

The Air Risk tab stores the initial and final ARC values. The applicant does not manually enter the final SAIL, it is calculated by the backend from the final GRC and final ARC.

The TMPR tab stores Tactical Mitigation Performance Requirement information through the shared TmprAssessment model. In SORA 2.0, the system derives whether the operation is VLOS or BVLOS from the selected operational scenario. TMPR text fields are required only where the derived operation type and final ARC make them relevant.

This workflow keeps the SORA 2.0 implementation shorter than that of SORA 2.5 while still preserving the main assessment steps required by the methodology.

5.6.2 SORA 2.0 Risk Calculation Implementation

The shared UAS physical and operational volume calculations were described in Section 5.5.3. Therefore, this subsection focuses only on the SORA 2.0-specific calculation logic. This includes the determination of the intrinsic Ground Risk Class, the application of M1, M2, and M3 mitigation values, the calculation of the final Ground Risk Class, and the determination of the final SAIL using the final GRC and final ARC.

The initial Ground Risk Class is determined using the selected operational scenario and the UAS category. The UAS category is derived from the characteristic dimension and kinetic energy calculated from the drone specification data.

UAS category	Condition used in implementation
Category 1	$CD \leq 1m$ and $KE < 700J$
Category 2	$CD \leq 3m$ and $KE < 34,000J$
Category 3	$CD \leq 8m$ and $KE < 1,084,000J$
Category 4	$CD > 8m$ and $KE \geq 1,084,000J$

The operational scenario determines the row of the SORA 2.0 GRC table. The implemented lookup table is shown below.

Operational scenario	Category 1	Category 2	Category 3	Category 4
VLOS/BVLOS over controlled ground area	1	2	3	4
VLOS in sparsely populated environment	2	3	4	5
BVLOS in sparsely populated environment	3	4	5	6
VLOS in populated environment	4	5	6	7
BVLOS in populated environment	5	6	8	10
VLOS over gathering of people	7	8	10	N/A
BVLOS over gathering of people	8	10	N/A	N/A

The initial Ground Risk Class is therefore calculated as:

$$iGRC = GRCTable(scenario, UASCategory)$$

where scenario is the operational scenario selected by the applicant, and UASCategory is the category determined from the characteristic dimension and kinetic energy.

After the initial GRC has been calculated, the system applies the M1, M2, and M3 mitigation values. M1 represents strategic ground-risk mitigations, M2 represents mitigations that reduce ground-impact effects, and M3 represents emergency response planning.

Mitigation	Condition	Implemented value
M1	Inspection only	-1
M1	Weather monitoring and UAV performance monitoring with low validation	-1
M1	Weather monitoring and UAV performance monitoring with medium validation	-2
M1	Weather monitoring and UAV performance monitoring with high validation	-4
M2	Training, procedures, and parachute with low validation	0

M2	Training, procedures, and parachute with medium validation	-1
M2	Training, procedures, and parachute with high validation	-2
M3	ERP not fully supported	+1
M3	ERP, ERP training, and ERP records with low validation	+1
M3	ERP, ERP training, and ERP records with medium validation	0
M3	ERP, ERP training, and ERP records with high validation	-1

The final Ground Risk Class is calculated by applying the mitigation values to the initial GRC:

$$fGRC = \max(1, iGRC + M1 + M2 + M3)$$

The maximum function ensures that the final GRC cannot fall below 1. The backend then stores the initial GRC, the mitigation values, and the final GRC in the RiskAssessment record.

The final SAIL is calculated from the final GRC and the final ARC selected in the air risk assessment.

Final GRC	ARC-a		ARC-b		ARC-c		ARC-d	
GRC ≤ 2	I		II		IV		VI	
GRC = 3	II		II		IV		VI	
GRC = 4	III		III		IV		VI	
GRC = 5	IV		IV		IV		VI	
GRC = 6	V		V		V		VI	
GRC = 7	VI		VI		VI		VI	
GRC > 7	Category operation	C	Category operation	C	Category operation	C	Category operation	C

In implementation terms, the SAIL is calculated as:

$$SAIL = SAILTable(fGRC, finalARC)$$

where *fGRC* is the final Ground Risk Class after mitigation and *finalARC* is the final air risk class selected in the air risk section. The resulting SAIL is stored in the ArcAssessment model and reused in the completed application view and generated documents.

When the SORA 2.0 application is completed, the backend calculates the initial GRC, applies M1/M2/M3, calculates the final GRC, determines SAIL, stores the results, and returns the calculated values to the interface.

5.6.3 SORA 2.0 Stored Outputs

After completion, the SORA 2.0 application stores both applicant-entered data and calculated assessment outputs. These values become the stable record used for read-only viewing, document generation, and later submission.

The stored outputs include pre-evaluation data, drone specification values, terminal velocity, kinetic energy, operational scenario, initial GRC, mitigation values, final GRC, initial ARC, final ARC, final SAIL, TMPR information where required, strategic mitigation entries, and the saved map polygon.

The completed application is displayed in read-only mode so that the applicant or reviewer can inspect the submitted values without modifying them. The stored outputs are also reused during document generation, where the system inserts the saved application data, calculated values, mitigation information, TMPR content, and map image into the generated SORA 2.0 document.

Saving these values at completion time preserves a stable assessment record for viewing, document generation, and submission.

5.7 SORA 2.5 Implementation

The SORA 2.5 workflow is implemented as the platform's main version-specific application flow. It uses the shared application components described in Section 5.5. However, SORA 2.5 requires additional data and calculation logic. For this reason, the system implements SORA 2.5 through the separate Application25 model and related assessment models.

The Application25 model links the main application record to the related pre-evaluation data, drone specification data, ground risk assessment, air risk assessment, TMPR data, containment data, strategic mitigation entries, map polygon data, population-density value, and operation manual file.

5.7.1 SORA 2.5 Workflow Structure

The SORA 2.5 application workflow is organised into seven visible tabs: Pre-evaluation, Drone Specifications, Ground Risk, Air Risk, TMPR and SAIL, OSOs and Containment, and Operation Manual. This extended tab structure reflects the additional assessment and documentation requirements of SORA 2.5. Unlike the SORA 2.0 workflow, which focuses mainly on pre-evaluation, drone data, ground risk, air risk, and TMPR, the SORA 2.5 workflow includes additional containment-related outputs and an explicit operation manual upload step.

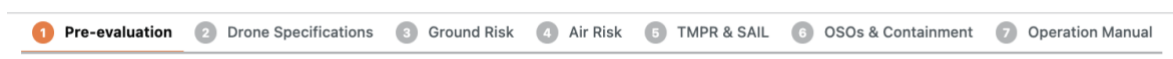


Figure 5.8: SORA 2.5 application form showing the extended seven-tab workflow.

The first tab, Pre-evaluation, collects the initial application and operation information. This includes the application title, planned date and time, organisation, contact details, operation description, and pre-evaluation questions. These values provide the basic contextual information for the application and are used later during document generation.

1 Pre-evaluation
2 Drone Specifications
3 Ground Risk
4 Air Risk
5 TMPR & SAIL
6 OSOs & Containment
7 Operation Manual

Please review the following statements before completing this application. To proceed with a SORA assessment, **all** answers must be **NO**. A single **YES** means SORA does not apply to your operation.

- ☐ The Operation under the "Open" category:
- ☐ The Operation is covered by a "Standard Scenario" recognized by the competent authority:
- ☐ The Operation under the "Certified" category:
- ☐ The Operation is subject to a specific NO-GO from competent authority:
- ☐ The Competent Authority has determined that the UAS is "harmless" for the ground risk:

Title: *

Date: *

Time: *

Organization: *

Email: *

First Name: *

Last Name: *

Description: *

[Reference: EU 2019/947 Article 5](#)

Figure 5.9: Pre-Evaluation Tab.

The second tab, Drone Specifications, collects the UAS technical data. These values include flight height, maximum take-off mass, maximum velocity, maximum wind speed, cross-sectional area, characteristic dimensions, parachute information, and deployment time. The shared UAS physical calculations described in Section 5.5.3 are applied to these values.

1 Pre-evaluation 2 **Drone Specifications** 3 Ground Risk 4 Air Risk 5 TMPR & SAIL 6 OSOs & Containment 7 Operation Manual

Drone Model Or Add Custom *

Select model

Max Takeoff Mass (kg): *

Max Velocity (m/s): *

Max Wind Speed (m/s): *

Cross Sectional Area (m²): *

Characteristic Dimensions (m): *

Flight Height (m): *

☐ Parachute

Deploy Time (s)

Reference: [EASA Easy Access Rules — AMC1 Article 11, Annex A](#)

Figure 5.10: Drone Specifications Tab.

The third tab, Ground Risk, implements the SORA 2.5 ground risk assessment. In this workflow, the initial GRC is not determined using the same operational-scenario table as before. Instead, it is calculated using the UAS characteristic dimension, maximum speed, population density of the selected operational area, and whether the operation is over a controlled ground area. The SORA 2.5 ground risk mitigation inputs are stored through the RiskAssessment25 model, which contains M1A, M1B, M1C, and M2 mitigations.

1 Pre-evaluation 2 Drone Specifications 3 **Ground Risk** 4 Air Risk 5 TMPR & SAIL 6 OSOs & Containment 7 Operation Manual

Ground Risk Class:
Draw operation area on map to see GRC

M1 (A) strategic mitigation - sheltering *
Specify the level of robustness:
☐ None ☐ Low ☐ Medium
•View robustness criteria

M1 (B) strategic mitigation - operational restrictions *
Specify the level of robustness:
☐ None ☐ Medium ☐ High
•View robustness criteria

M1 (C) tactical mitigation - ground observation *
Specify the level of robustness:
☐ None ☐ Low
•View robustness criteria

M2 Effects on UA impact dynamics are reduced *
Specify the level of robustness:
☐ None ☐ Medium ☐ High
•View robustness criteria

Ground Risk Class After Mitigations:
Draw operation area on map to see GRC

Reference: [EASA Easy Access Rules — AMC1 Article 11, Annex B](#)

Figure 5.11: Ground Risk Assessment Tab.

The fourth tab, Air Risk, captures the ARC-related information. The system stores the initial, adjacent, and final or residual ARCs using the SORA 2.5 air risk assessment model. This tab also contains the optional strategic mitigation section, where the applicant may add mitigation entries and provide a rationale if strategic air risk mitigations are applied. Strategic mitigations are not mandatory; if none are applied, the applicant selects the same residual ARC value as the initial ARC. The final SAIL is calculated from the final GRC and the residual ARC, but it is displayed in the following TMPR and SAIL tab rather than in the Air Risk tab.

1 Pre-evaluation
2 Drone Specifications
3 Ground Risk
4 **Air Risk**
5 TMPR & SAIL
6 OSOs & Containment
7 Operation Manual

Air Risk Class: *
Click on the button that corresponds to your operational scenario path on the flowchart.

Reference: [EASA Easy Access Rules – AMC1 Article 11, Annex C](#)

Strategic Mitigations
Strategic mitigations are optional. Add mitigation lines if applicable. Complete the Rationale to enable +.

Strategic Mitigation Field
Mitigation Rationale

Select Mitigation
Explain mitigation...

Residual ARC: *
If no strategic mitigations are applied, select the same value as the Initial ARC.

Select ARC

Figure 5.12: Air Risk Assessment Tab.

The fifth tab, TMPR and SAIL, displays the calculated final SAIL and supports the tactical mitigation part of the SORA 2.5 workflow. This tab stores the operation type, including VLOS, EVLOS, or BVLOS. It also conditionally displays the TMPR fields for detect, decide, command, execute, feedback loop, and assurance. If the operation type or residual ARC do not require TMPR, the interface displays a no-requirement message. If TMPR is required, the relevant text fields are shown for applicant input.

The screenshot shows the 'TMPR & SAIL' tab selected in a navigation bar with seven items: 1 Pre-evaluation, 2 Drone Specifications, 3 Ground Risk, 4 Air Risk, 5 TMPR & SAIL, 6 OSOs & Containment, and 7 Operation Manual. The main content area has three sections:

- Type of operation ***: Three radio buttons are present: 'Visual line of sight (VLOS)', 'Extended visual line of sight (EVLOS)', and 'Beyond visual line of sight (BVLOS)'. None are selected.
- Tactical Mitigation Performance Requirements (TMPR) ***: A message box states 'TMPR is only required for BVLOS operations.'
- Final SAIL:**: A button labeled 'Draw operation area on map to calculate sail'.

At the bottom, a reference link is provided: [EASA Easy Access Rules — AMC1 Article 11, Annex D](#).

Figure 5.13: TMPR & SAIL Tab.

The sixth tab, OSOs and Containment, presents the containment-related part of the SORA 2.5 workflow. This section captures outdoor assembly information, displays the calculated containment level, and presents the applicable Operational Safety Objectives according to the calculated SAIL. The containment calculation depends on the SAIL, adjacent population density, and outdoor assembly information. The tab uses the SAIL value calculated earlier, but the main SAIL output is shown in the TMPR and SAIL tab.

The screenshot shows the 'OSOs & Containment' tab selected in the navigation bar. The main content area has two primary sections:

- Outdoor Assemblies *** (1km out of operational area): A dropdown menu labeled 'Select range' with a downward arrow.
- Containment Level:**: A button labeled 'Draw operation area on map to calculate containment'.

Below these sections, a reference link is provided: [EASA Easy Access Rules — AMC1 Article 11, Annex E](#).

At the bottom, a section titled **Applicable Operational Safety Objectives (OSOs)** contains the text: 'Complete Step 5 (ARC) to determine your SAIL and see the applicable OSOs.'

Figure 5.14: OSOs & Containment Tab.

The seventh tab, Operation Manual, handles the upload of the operation manual PDF. This tab is specific to SORA 2.5 and is treated as part of the application content rather than as the final signed submission document.

Operation Manual Guidelines

An Operation Manual (OM) describes how your operation is conducted safely and must be submitted with your SORA application. Expand each section below to see what must be included.

- Cover & Document Control
- Concept of Operations (ConOps)
- UAS & Equipment Description
- Crew & Personnel
- Normal Operating Procedures
- Abnormal & Emergency Procedures
- Maintenance Programme
- Training Programme
- Safety Management
- Record Keeping

Example Operation Manuals

The following are official EASA example Operation Manuals for reference. Use them as a template when preparing your own OM.

EASA Operations Manual Example SAIL II

An example of an Operations Manual developed by EASA to be used for SAIL II operations

EASA Operations Manual Example PDRA-S01

An example of an Operations Manual developed by EASA to be used for PDRA-S01 operations

Your Operation Manual *

Upload your completed Operation Manual (PDF only).

Select PDF

Choose File No file chosen

Upload

Figure 5.15: Operational Manual Tab.

5.7.2 SORA 2.5 GRC, SAIL, and Containment Calculation

The shared UAS physical and operational volume calculations were described in Section 5.5.3. Therefore, this subsection focuses on the SORA 2.5-specific calculation logic: initial GRC, mitigation effects, final GRC, SAIL, TMPR, and containment level.

The initial Ground Risk Class is calculated using the DroneValues5 utility logic. The calculation uses the maximum characteristic dimension, maximum speed, population density, and controlled-ground-area status. The system first determines a conservative

column index by comparing the UAS characteristic dimension category and the maximum speed category:

$$C = \max(CD, V)$$

Where CD is the category derived from characteristic dimension and V is the category derived from maximum speed. The implementation uses the following thresholds:

Column	Characteristic dimension threshold	Maximum speed threshold
1	$CD \leq 1m$	$V \leq 25 m/s$
2	$CD \leq 3m$	$V \leq 35 m/s$
3	$CD \leq 8m$	$V \leq 75 m/s$
4	$CD \leq 20m$	$V \leq 120 m/s$
5	$CD \leq 40m$	$V \leq 200 m/s$

The system then determines the GRC row. If the operation is conducted over a controlled ground area, the controlled-ground-area row is used. Otherwise, the row is selected from the population density of the selected operational area.

Row condition	Population-density condition
Controlled ground area	Controlled ground area selected
Very low density	$P < 5 \text{ people/km}^2$
Low density	$5 \leq P < 50 \text{ people/km}^2$
Medium-low density	$50 \leq P < 500 \text{ people/km}^2$
Medium density	$500 \leq P < 5,000 \text{ people/km}^2$
High density	$5,000 \leq P < 50,000 \text{ people/km}^2$
Very high density	$P \geq 50,000 \text{ people/km}^2$

The initial GRC is then obtained from the SORA 2.5 GRC lookup table:

Population / area condition	Col. 1	Col. 2	Col. 3	Col. 4	Col. 5
Controlled ground area	1	1	2	3	3
$P < 5$	2	3	4	5	6
$5 \leq P < 50$	3	4	5	6	7
$50 \leq P < 500$	4	5	6	7	8
$500 \leq P < 5,000$	5	6	7	8	9
$5,000 \leq P < 50,000$	6	7	8	9	10
$P \geq 50,000$	7	8	Certified	Certified	Certified

In implementation terms:

$$iGRC = GRC25Table(P_{row}, C)$$

where P_{row} is the population-density or controlled-ground-area row and C is the conservative column selected from the dimension and speed categories.

After the initial GRC is calculated, the system applies the SORA 2.5 mitigation values. Unlike SORA 2.0, which uses M1, M2, and M3, this version uses M1A, M1B, M1C, and M2. These fields are represented in the RiskAssessment25 model and form.

Mitigation	Available level	Implemented value
M1A sheltering	None	0
M1A sheltering	Low	-1
M1A sheltering	Medium	-2
M1B operational restrictions	None	0
M1B operational restrictions	Medium	-1
M1B operational restrictions	High	-2
M1C ground observation	None	0
M1C ground observation	Low	-1
M2 effects reduced	None	0
M2 effects reduced	Medium	-1
M2 effects reduced	High	-2

The total M1 reduction is calculated as:

$$M1_{total} = M1A + M1B + M1C$$

The final GRC is then calculated as:

$$fGRC = \max(GRC_{floor}, iGRC + M1_{total} + M2)$$

where GRC_{floor} is the controlled-ground-area floor value for the selected column. This floor rule prevents the final GRC from being reduced below the minimum value allowed for that UAS dimension and speed category. The implementation explicitly stores this floor value during the initial GRC calculation and applies it during the final GRC calculation.

1 Pre-evaluation 2 Drone Specifications 3 **Ground Risk** 4 Air Risk 5 TMPR & SAIL 6 OSOs & Containment 7 Operation Manual

Ground Risk Class:
GRC: 5

M1 (A) strategic mitigation - sheltering *
Specify the level of robustness:
☐ None ☒ Low ☐ Medium
•View robustness criteria

M1 (B) strategic mitigation - operational restrictions *
Specify the level of robustness:
☐ None ☒ Medium ☐ High
•View robustness criteria

M1 (C) tactical mitigation - ground observation *
Specify the level of robustness:
☐ None ☒ Low
•View robustness criteria

M2 Effects on UA impact dynamics are reduced *
Specify the level of robustness:
☐ None ☒ Medium ☐ High
•View robustness criteria

Ground Risk Class After Mitigations:
GRC: 1

Reference: [EASA Easy Access Rules — AMC1 Article 11, Annex B](#)

Figure 5.16: Ground Risk tab showing calculated GRC and GRC after mitigations.

After the final GRC is available, the system calculates the SAIL using the final GRC and the residual ARC selected in the Air Risk tab. The residual ARC represents the remaining

air risk after any optional strategic air risk mitigations. If no strategic mitigations are applied, the residual ARC remains the same as the initial ARC.

$$SAIL = SAILTable(fGRC, finalARC)$$

The implemented SAIL lookup table is identical to the SORA 2.0 SAIL table.

The containment level is calculated after SAIL has been determined. The backend selects the applicable containment table according to the UAS characteristic dimension and maximum speed. If the UAS is outside the implemented standard envelopes, the system returns N/A, indicating that further analysis is required.

$$Containment = ContainmentTable(SAIL, P_{adjacent}, A_{outdoor})$$

For UAS with $CD \leq 1m$ and $V < 25m/s$, the first containment table is used.

SAIL	Outdoor assemblies > 400,000	Outdoor assemblies 40,000 - 400,000	Outdoor assemblies < 40,000
I-II	High	Medium	Low
III	Medium	Low	Low
IV	Low	Low	Low
V-VI	Low	Low	Low

For UAS with $CD \leq 3m$ and $V < 35m/s$, the first containment table is used.

SAIL	Outdoor assemblies > 400,000	Outdoor assemblies 40,000 - 400,000	Outdoor assemblies < 40,000 and $P_{adjacent} < 50,000 \text{ ppl/km}^2$	Outdoor assemblies < 40,000 and $P_{adjacent} < 5,000 \text{ ppl/km}^2$
I-II	Out of scope	High	Medium	Low
III	Out of scope	Medium	Low	Low
IV	Medium	Low	Low	Low
V-VI	Low	Low	Low	Low

where $P_{adjacent}$ is the adjacent population density and $A_{outdoor}$ is the outdoor assembly category selected by the applicant. The containment level is displayed in the OSOs and Containment tab together with the applicable OSO set.

1 Pre-evaluation
2 Drone Specifications
3 Ground Risk
4 Air Risk
5 TMPR & SAIL
6 OSOs & Containment
7 Operation Manual

Outdoor Assemblies • (1km out of operational area)

Below 40,000 people

Containment Level:

Low

Reference: [EASA Easy Access Rules — AMC1 Article 11, Annex E](#)

Applicable Operational Safety Objectives (OSOs)

Your SAIL: 4. The OSOs below apply to your operation. Items marked NR are not required and are hidden.

OSO #01	Operator is competent and/or proven	H
OSO #02	UAS manufactured by a competent and/or proven entity	M
OSO #03	UAS maintained by a competent and/or proven entity	M
OSO #04	UAS developed to authority recognised design standards	L
OSO #05	UAS designed considering system safety and reliability	M
OSO #06	C3 link performance is adequate for the operation	M
OSO #07	Inspection of the UAS to ensure it is in a condition for safe operation	M
OSO #08	Operational procedures are defined, validated and adhered to	H
OSO #09	Remote crew trained and current and fit to operate the UAS	M
OSO #10	Safe recovery from a technical issue	M
OSO #13	External services supporting UAS operations are adequate	H
OSO #16	Multi-crew coordination	H
OSO #17	Remote crew is fit to operate the UAS	H
OSO #18	Automatic protection of the flight envelope from human errors	H
OSO #19	Safe recovery from human errors	M
OSO #20	A human factors evaluation has been performed	M
OSO #23	Environmental conditions for safe operations defined, measurable and adhered to	M
OSO #24	UAS is designed and qualified for the adverse environmental conditions	H

Figure 5.17: OSOs and Containment tab showing containment level and applicable OSOs.

The implemented live calculation sequence is therefore:

$$iGRC \rightarrow fGRC \rightarrow SAIL \rightarrow Containment$$

The endpoint returns intermediate reasons when values cannot yet be calculated. For example, if drone specifications are missing, the response indicates that specifications are required. If population density is missing, the response indicates that density is required. If the final ARC is missing, the system can return the final GRC but not SAIL or

containment. If outdoor assembly information is missing, the system can return SAIL but not containment.

When the SORA 2.5 application is completed, the backend stores all user input and calculated values, displays a success modal with instructions for the next steps, and shows the buffered operational area.

5.8 Validation and Completion Feedback

The platform implements feedback and validation through a combination of Django form validation, backend completion checks, client-side JavaScript, map validation logic, and shared modal components.

When the applicant selects Complete, the system checks whether the application contains the required information before allowing the application status to change from draft to completed. The validation is not limited to ordinary form fields. It also checks whether the pre-evaluation conditions are acceptable, whether a valid operational area has been drawn on the map, whether the email address is valid, and whether all required fields have been completed. If one or more checks fail, the system displays an error modal explaining why the application cannot be completed. The modal can include separate messages for missing required fields, a missing map area, an invalid email address, or failed pre-evaluation conditions.

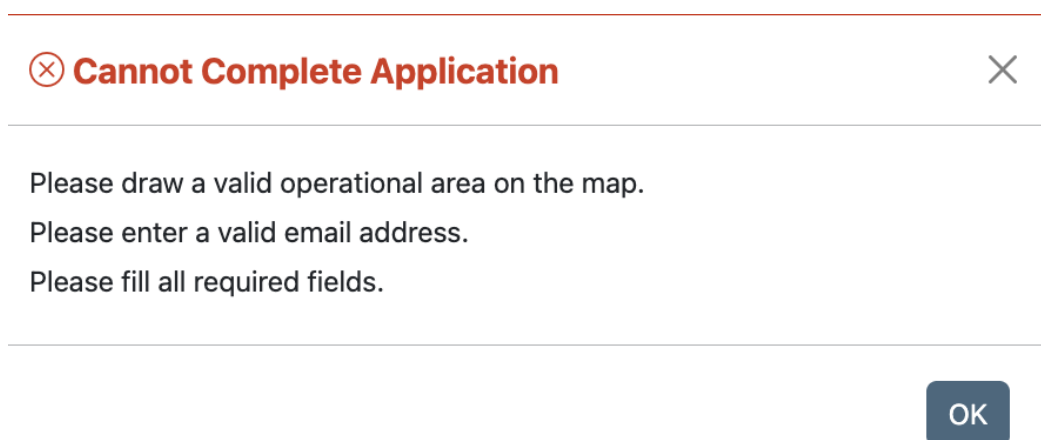


Figure 5.18: Error modal displayed when required application information is missing.

At the same time, the interface also marks missing or invalid fields after a failed completion attempt. Invalid fields are highlighted with red borders and error styling, but this styling is applied only after the applicant attempts to complete the form. This prevents the page from appearing invalid immediately when the applicant first opens a draft or starts a new application.



Organization: *

Email: *

evaggeliakringou

Figure 5.19: Missing or invalid fields highlighted after a failed completion attempt.

The validation logic also redirects the applicant to the relevant tab. When a required field is missing, the JavaScript identifies the tab that contains the first invalid field and activates it automatically. Map errors are handled through the modal because the map is displayed in a separate map column rather than inside a form tab.

The map component has additional validation behaviour because some errors are spatial rather than form-based. The system prevents invalid operational areas, including polygons with fewer than three points, shapes with no valid area, points placed inside restricted zones, and operational areas or generated buffers that overlap restricted flight zones. These errors are displayed through the same feedback modal structure, with map-specific messages and, where required, clearing behaviour after the modal is dismissed.

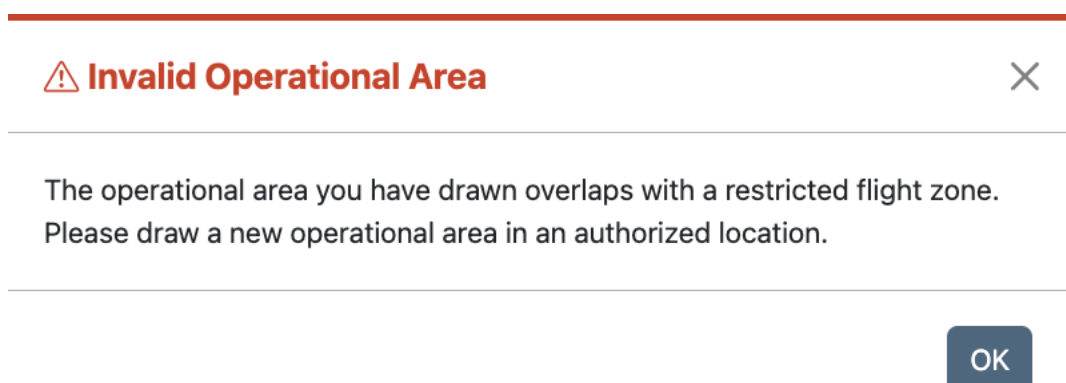


Figure 5.20: Map-specific warning displayed for an invalid or restricted operational area.

Map validation also applies to reserved operational areas. If the applicant draws a polygon that overlaps a reserved zone for the selected operation date, the system displays an invalid operational area modal and prevents the proposed area from being accepted. This uses the same validation approach as restricted-zone checking, but the reserved zones are dynamic because they are derived from accepted applications rather than static spatial datasets.

The map also provides feedback for authorisation-required zones. When the operational area or related buffers interact with relevant yellow zones, the interface can display zone information in the feedback modal. The modal may include expandable zone descriptions so that the applicant can inspect the relevant spatial information before proceeding.

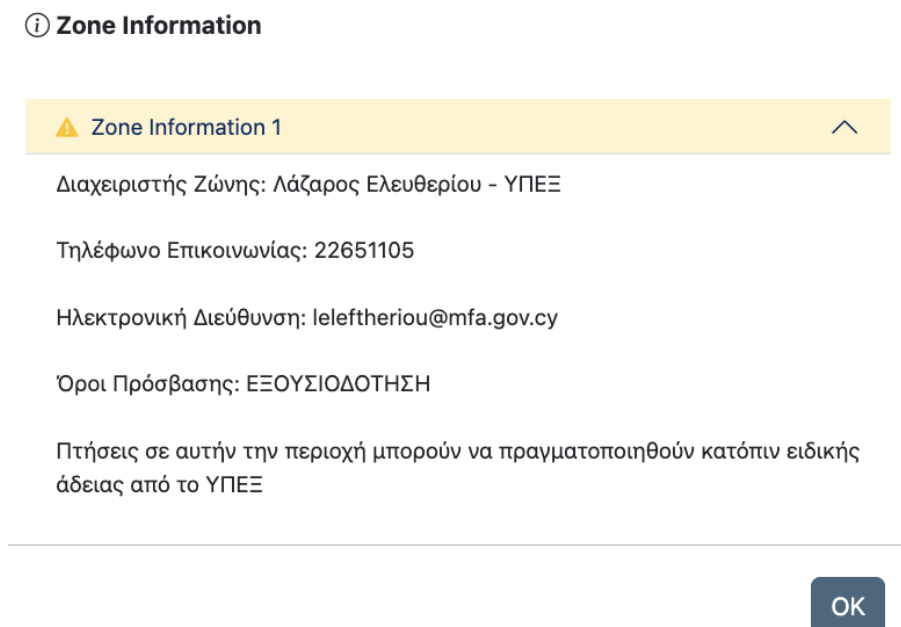


Figure 5.21: Zone information in Completion modal.

The system also protects users from accidentally losing unsaved progress. If the applicant attempts to leave the application form while there may be unsaved changes, the system displays an Unsaved Changes modal. This safeguard is important because the application form contains multiple data may require substantial user effort to complete.

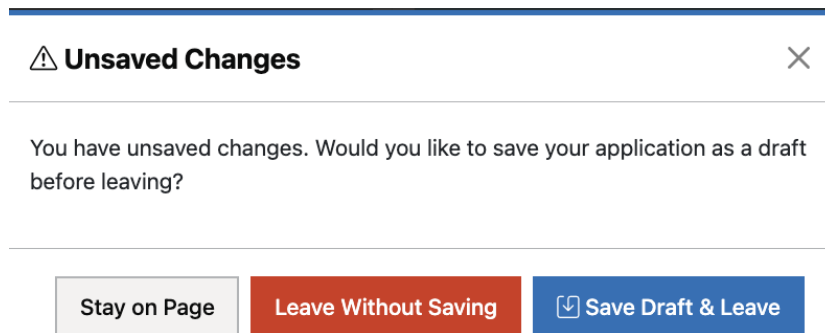


Figure 5.22: Unsaved changes modal shown before leaving the application form.

After successful completion, the system displays a success modal confirming that the application has been completed. The success modal includes the application identifier and title, and it also displays calculated outputs. The modal also informs the applicant that the completed application can now be found in the Completed Applications table and that the next step is to generate the submission document for the CAA.

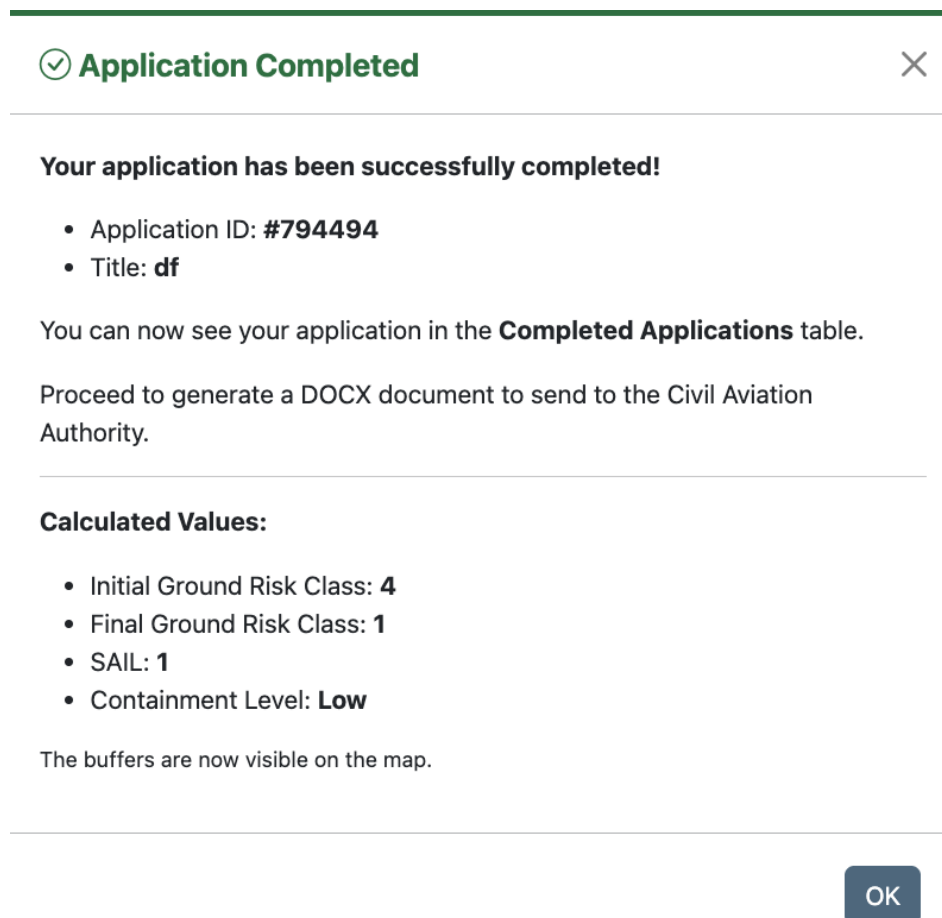


Figure 5.23: Success modal displayed after successful application completion.

The successful completion process also changes the state of the interface. Once an application has been completed, the form fields are disabled, and the completed application becomes available in read-only mode. The system also updates the displayed calculation outputs and shows the map buffers associated with the completed application. This prevents the user from accidentally changing the completed record while still allowing the applicant to inspect the application data, calculated values, and operational area.

The feedback implementation, therefore, supports both usability and data correctness. Error modals explain why completion is blocked; red field highlighting identifies missing inputs; tab navigation directs the applicant to the relevant section; map warnings handle spatial errors; the unsaved-changes modal protects draft progress; and the success modal confirms that the application has moved into the completed state. These behaviours reduce the likelihood of incomplete applications, make correction easier for the applicant, and preserve the integrity of the completed application record.

5.9 Document Generation and Submission Workflow

After an application has been completed, document generation and submission are handled from the My Applications dashboard. Completion stores the application data and calculated assessment outputs, but it does not submit the final signed documentation to the CAA. Submission requires a separate process: the applicant generates the required document output, reviews or signs it externally, and uploads the final file through the dashboard.

5.9.1 Document Generation

Document generation is available only for completed applications. When the applicant selects the generate action, the backend retrieves the relevant application, verifies ownership, checks that the status is completed, and then delegates document construction to the utility layer.

The main document-generation logic is implemented in `utils.py`. The Django view is responsible for access checks, application retrieval, map image preparation, and returning

the generated response, while the utility functions handle constructing the DOCX files from stored application data.

For SORA 2.0, the system generates a single DOCX file using `DocxGenerator.generate_docx()`. The function opens a predefined DOCX template and loads all the previously mentioned information into the template.

The generation utility also converts stored codes into readable document text. For example, it maps operational scenario values to textual descriptions, prepares ARC and OSO-related values, and formats calculation outputs before inserting them into the document. This keeps document construction outside the view and avoids duplicating formatting logic across the application.

The map image is handled separately because the map is rendered dynamically in the browser. The backend uses Selenium to open the completed application view, wait for the map layers to render, capture the map container, encode the screenshot, and pass on.

For SORA 2.5, the generation process produces a ZIP package rather than a single document. The ZIP package contains the generated SORA 2.5 outputs, including the operational authorisation application, the compliance matrix, and the traffic monitoring SORA document. The relevant SORA 2.5 generation functions are also implemented in `utils.py`. Some supporting templates are prepared for placeholder-based population, while the main traffic monitoring SORA document is populated in greater detail.

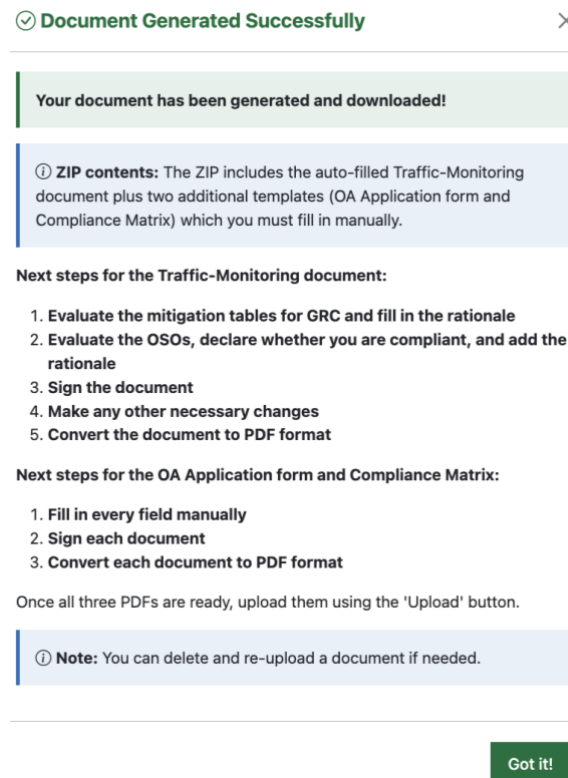


Figure 5.24: Document generation success modal showing the next steps for the applicant.

The generation success message informs the applicant that the document has been created and explains the next step: the applicant must open the generated file, review the content, complete any remaining externally required information if needed, sign or convert the document to PDF, and upload the signed version through the dashboard.

5.9.2 Signed Document Upload

The system does not implement the legal signing process internally. After downloading the generated output, the applicant reviews and signs or converts the document outside the platform. The final file is then uploaded from the completed applications table in the My Applications dashboard.

When the applicant selects the upload action, the interface opens an upload modal and passes the selected application identifier and SORA version to the upload form. The

backend receives the uploaded file, retrieves the corresponding completed application, and verifies that the application belongs to the authenticated user.

Signed submission files are stored through the SubmittedDocument model. Each uploaded document is linked to the selected application. The submitted document record stores the uploaded file, uploading user, document type, upload timestamp, and review status. New uploads are assigned the initial status Pending Review.

For SORA 2.5, the upload logic also supports ZIP files containing supporting generated documents. When a ZIP file is uploaded, the backend extracts the relevant files, such as the operational authorisation and compliance matrix documents, and stores them as submitted document records linked to the same SORA 2.5 application.

After a successful upload, the interface displays confirmation feedback and updates the dashboard so that the applicant can see the uploaded document and its review status.

5.9.3 Applicant Submission Management

After submission, the applicant can monitor the uploaded document from the My Applications dashboard. The completed applications table displays the current review status of each submitted document, including Pending, Accepted, Rejected, or Expired. This allows the applicant to follow the review process from the same page used for generation and upload.

If the document is rejected, the dashboard provides access to the rejection reason entered by the reviewer. The reason for rejection, therefore, links the administrative review workflow back to the applicant-facing submission process.

The applicant can also access uploaded files from the dashboard. For SORA 2.0, the applicant can view the uploaded PDF. For SORA 2.5, the dashboard can additionally provide access to ZIP-related outputs where applicable. This keeps the generated and uploaded material associated with the completed application record.

The dashboard also allows uploaded documents to be viewed or deleted. Deletion is protected by a confirmation modal and removes both the stored file and the corresponding SubmittedDocument record. For SORA 2.5, supporting files are stored in the same package and are also removed to avoid leaving incomplete document sets.

My Applications

+ New Application

▼ Draft Applications

ID	Title	Status	Created	Actions
#181291	Application #181291	DRAFT	May 20, 2026	Edit

▼ Completed Applications (SORA 2.5)

ID	Title	Status	Created	Actions	Uploaded	Review	Document
#173225	UAS	COMPLETED	May 21, 2026	View Generate Upload	—	—	—
#722536	title	COMPLETED	May 20, 2026	View Generate Upload	May 21, 2026 23:26	PENDING	PDF ZIP
#601359	operation	COMPLETED	May 20, 2026	View Generate Upload	May 21, 2026 23:24	REJECTED	PDF ZIP
#290278	title	COMPLETED	May 20, 2026	View Generate Upload	May 20, 2026 01:04	ACCEPTED	PDF ZIP

▼ Completed Applications (SORA 2.0)

ID	Title	Status	Created	Actions	Uploaded	Review	Document
#259538	title	EXPIRED	Feb 09, 2026	View Generate Upload	Feb 09, 2026 12:33	EXPIRED	PDF

Figure 5.25: My Applications Dashboard displayed submission management.

5.10 Administrative Dashboard and Review Workflow

The administrative dashboard implements the staff-facing part of the review workflow. After an applicant uploads the signed submission file, the submitted material becomes available to authorised Civil Aviation Authority staff through the dashboard. This separates applicant submission from staff review and provides a controlled interface for reviewing completed applications and uploaded documents. In the staff interface, User Management provides controlled local account administration through Django administration.

Access to the dashboard is restricted to authorised users. The navigation bar displays the Admin Dashboard link only for staff or administrative users, and the backend also checks the user's role before allowing access. If a non-administrative user attempts to access the dashboard, the system redirects the user back to the applicant area.

The screenshot shows the 'Civil Aviation Authority Dashboard' for the 'SORA Application Platform'. The top navigation bar includes 'gov.cy', 'user | Sign Out', and language options 'EL' and 'EN'. The main navigation bar has links for 'Home', 'My Applications', and 'Admin Dashboard'. Below the header, the dashboard title 'Civil Aviation Authority Dashboard' is followed by the subtitle 'Review all submitted applications and documents'. A filter section allows filtering by status (All, Pending, Accepted, Rejected) and sorting by submission date (Newest First). The dashboard is divided into three sections: 'SORA 2.5 Applications', 'SORA 2.0 Applications', and 'Expired Applications'. Each section contains a table of applications with columns for User, Application ID, Issued for, Submitted, and Actions.

User	Application ID	Issued for	Submitted	Actions
regular	#290278	Jun 07, 2026 12:00	May 20, 2026	View Application View PDF Download ZIP Pending
user	#892531	Jun 09, 2026 12:00	May 19, 2026	View Application View PDF Download ZIP Pending

User	Application ID	Issued for	Submitted	Actions
user	#294405	May 24, 2026 18:59	Apr 25, 2026	View Application View PDF Pending

User	Application ID	Issued for	Submitted	Actions
regular	#259538	Feb 27, 2026 14:32	Feb 09, 2026	View Application View PDF EXPIRED

Figure 5.26: Administrative dashboard showing submitted applications awaiting review.

The dashboard also includes filtering and sorting controls. Submitted documents can be filtered by review status, including All, Pending, Accepted, and Rejected. Records can also be sorted by submission date, issued date, or user. The backend applies the selected filter and sorting option before rendering the dashboard, so the displayed records match the reviewer's selected criteria.

The backend view retrieves completed applications with submitted documents and prepares separate datasets for SORA 2.0 and SORA 2.5. It annotates applications with the latest submission date, applies status filtering, separates expired applications from active ones based on the planned operation date, and passes the prepared records to the dashboard template. For SORA 2.5, the backend also identifies the primary submitted document used for the main review controls.

Review decisions are made through the status control in the dashboard. A submitted document may be marked as Accepted or Rejected. If the reviewer selects Rejected, the interface requires a rejection reason before the decision can be submitted.

The status update is submitted asynchronously to the backend. The backend verifies that the user is authorised, retrieves the selected SubmittedDocument record, validates the requested status, and prevents documents that have already been accepted or rejected from being changed again. This prevents accidental modification of final review decisions.

If the document is accepted, the backend updates the review status to Accepted. If the document is rejected, the backend stores the rejection reason and updates the review status to Rejected. When the status is not rejected, any previous rejection reason is cleared. The backend then returns a JSON response confirming whether the update was successful.

The review workflow also supports applicant notification. When a document is accepted or rejected, the backend attempts to send an email notification using the contact email stored in the application's pre-evaluation data. Rejection notifications include the rejection reason where available, allowing the applicant to correct the submission and upload a revised document.

After the review decision is saved, the result becomes visible to the applicant in the My Applications dashboard. Pending, accepted, rejected, and expired statuses are therefore shown to the relevant applicant, while the review controls remain restricted to staff users.

The administrative dashboard completes the application lifecycle by allowing authorised staff to inspect submitted applications, review uploaded documents, accept or reject

submissions, record rejection reasons, and return the review outcome to the applicant-facing dashboard.

5.11 Internationalisation Implementation

The system implements internationalisation to support both English and Greek interfaces. This is relevant to the platform's Cypriot regulatory context, where applicants and CAA staff may need to use either language.

The implementation uses Django's built-in internationalisation mechanism. User-facing text in templates is marked using Django translation tags, while Python-side strings are marked using translation functions where required. This allows the same templates and views to display different text depending on the active language.

The base layout includes a language selector in the navigation bar. Users can switch between English and Greek through the header controls, which are available across the main pages because the navigation bar is shared by the platform layout.

The implementation also supports translated client-side messages. Several user-facing messages are produced through JavaScript, including modal text, validation feedback, and map-related instructions. These messages are passed through shared JavaScript internationalisation data so that dynamic interface feedback can follow the selected language.

This approach allows the platform to localise labels, buttons, instructions, modal messages, validation feedback, and map-related text without duplicating templates or separating the application logic by language.

Chapter 6

System Evaluation and Contribution

6.1 Challenges of the Traditional SORA Process	105
6.2 System Contribution	107
6.3 Comparison with the Manual Approach	108
6.4 Usability and Performance Discussion	109

This chapter evaluates the contribution of the developed system in relation to the traditional SORA application process. It discusses the main difficulties of the manual workflow, explains how the implemented platform addresses them, compares the digital workflow with the manual approach, and considers usability and performance. The evaluation is based on the system features implemented, as described in Chapter 5.

6.1 Challenges of the Traditional SORA Process

The traditional SORA process is complex because it requires applicants to combine regulatory interpretation, technical calculation, operational-area analysis, mitigation selection, and document preparation. In the manual approach, these tasks are usually performed through separate guidance documents, tables, templates, maps, and calculations. This makes the process difficult to complete consistently, especially for applicants who are not already familiar with the methodology.

A major challenge is the number of technical concepts involved. Applicants must understand terms such as GRC, ARC, SAIL, TMPR, OSOs, containment, mitigation robustness, operational volume, contingency volume, and ground risk buffer. These concepts are also connected to one another. For example, SAIL depends on ground and air risk, while containment depends on SAIL and surrounding operational conditions.

The process also requires movement between several sources of information. The applicant may need to consult regulatory text, identify the correct table, perform calculations, interpret map-based constraints, complete template fields, and then transfer the same information into the final application material. This increases the risk of missing fields, copying values incorrectly, or producing inconsistencies between the assessment and the submitted documents.

Manual calculation is another source of difficulty. Values such as ground risk class, final ground risk class, SAIL, containment level, terminal velocity, kinetic energy, and buffer dimensions depend on several inputs. If these calculations are performed manually or through separate tools, errors may occur due to incorrect input values, wrong table selection, or inconsistent interpretation of mitigation effects.

The map-related part of the process is particularly difficult to manage manually. The applicant must define the intended operational area, but this area is not only a location description. It affects population density, restricted-area interaction, operational volume, contingency volume, and ground risk buffer. In a manual workflow, the applicant may need to use separate map tools, identify whether the area intersects restricted or authorisation-required zones, retrieve or estimate population-density information, and then apply those results to the relevant SORA calculations. This separates the geographical information from the risk assessment logic.

The transition between SORA 2.0 and SORA 2.5 introduces another source of complexity. The two methodologies use different workflow structures and calculation rules in several areas. Applicants must understand which version applies, which fields are required, and how the outputs differ. Without a structured workflow, this distinction may cause confusion.

These challenges also affect CAA staff. Manually prepared submissions may differ in structure, completeness, terminology, and calculation traceability. Missing information or inconsistent values can slow the review process and require additional clarification from the applicant.

6.2 System Contribution

The developed system contributes a structured digital workflow for preparing, generating, submitting, and reviewing SORA applications. Its main contribution is the integration of application data, calculations, map information, generated documents, and review status within one platform.

The system guides applicants through the SORA workflow using a step-based application interface. Instead of requiring applicants to move independently between guidance material, calculation tables, external maps, and document templates, the platform organises the process into defined application stages. This is particularly relevant for SORA 2.5, which includes population-density-based ground risk assessment, containment calculation, OSO presentation, and operation manual handling. SORA 2.0 is retained as a supported legacy and comparative workflow.

The platform automates key calculation steps. It calculates drone-related physical values, operational volume parameters, ground risk class, final ground risk class, SAIL, and containment level where applicable. These values are generated from the submitted application data and reused across the interface, completed application views, and generated documents.

The system also integrates map-based operational area definition. Applicants draw the intended operational area directly on the map, while the platform stores the polygon and connects it with population-density data, restricted and reserved-area information, and buffer visualisation. This makes the relationship between geography and risk outputs clearer than in a manual workflow.

Another contribution is document generation from stored application data. The applicant does not need to manually transfer the same information into document templates. Instead, the system generates the required outputs from the completed application record. For SORA 2.5, this includes support for multiple generated outputs through a ZIP-based workflow.

The platform also connects applicant submission with administrative review. After external signing or conversion, the applicant uploads the signed PDF through the dashboard. The submitted document then becomes available to authorised Civil Aviation Authority staff, who can inspect the application, review the uploaded material, accept or reject the submission, and provide rejection reasons where required.

The system improves traceability by preserving application states and outputs. Drafts can be saved, completed applications become read-only records, calculated values are stored with the application, submitted documents have review statuses, and review actions can be linked to the relevant application record. The system therefore supports the SORA process without replacing regulatory judgement or the final decision of Civil Aviation Authority staff.

6.3 Comparison with the Manual Approach

The manual process and the developed system differ mainly in how information is collected, calculated, reused, and reviewed. In the manual approach, the applicant must coordinate these activities independently. In the developed system, they are handled through a connected digital workflow.

Aspect	Manual SORA Process	Developed System
User guidance	User must follow documents and templates manually.	User follows a guided application workflow.
Calculations	Performed manually or through separate tools.	Calculated automatically by the system.
Map use	Operation area may be described separately.	Operation area is drawn directly on an interactive map.
Data consistency	Same data may need to be repeated in different documents.	Stored data is reused for calculations and document generation.
Document preparation	Completed manually by the applicant.	Generated automatically from application data.

Review process	Submissions may be handled through separate communication channels.	Uploaded documents can be reviewed in the system.
Error risk	Higher risk of missing fields or inconsistent values.	Required data and calculated values are handled more consistently.
SORA versions	User must understand differences manually.	Separate workflows support SORA 2.0 and SORA 2.5.

The comparison shows that the developed system does not change the regulatory purpose of SORA. It changes the way the process is carried out by reducing fragmentation between forms, calculations, maps, documents, and review status.

The system also improves the connection between the completed digital assessment and the final submission material. Calculated values and application data are stored with the application and reused during document generation, reducing the risk of transcription errors.

The review stage is also more structured. Submitted documents are associated with the corresponding completed application, and review outcomes are visible to the applicant through the dashboard. However, the system does not replace human review. CAA staff remain responsible for the final decision.

6.4 Usability and Performance Discussion

The usability of the developed system is evaluated from the perspective of the implemented workflow rather than through a formal user study. The system improves usability by dividing the application into sections, preserving draft progress, displaying calculated outputs, and linking the application form with the operational map.

The tab-based interface avoids presenting the applicant with one long form. This is useful because the SORA process contains different categories of information, including applicant details, UAS specifications, ground risk, air risk, TMPR, containment,

operation manual handling, and submission documents. Separating these into sections makes the workflow easier to follow and allows the applicant to return to specific parts of the application when corrections are required.

The applicant dashboard supports application management. Draft applications, completed applications, generated documents, uploaded files, and review statuses are managed from the same page. This gives the applicant one main location for continuing unfinished work, viewing completed records, generating documents, uploading signed PDFs, and monitoring review outcomes.

The map component supports usability by allowing the operational area to be drawn directly inside the workflow. It also provides feedback on invalid polygons, restricted- and reserved-area interactions, population-density-related outputs, and map buffers after completion.

Live and automatic calculations help applicants understand the effect of entered values. Outputs such as GRC, final GRC, SAIL, containment level, terminal velocity, kinetic energy, and buffer dimensions are generated from application data. This is especially relevant in the SORA 2.5 workflow, where population density, mitigations, residual ARC, and containment are connected.

Completed applications are displayed in read-only mode. This prevents accidental modification of the stored assessment record while still allowing applicants and reviewers to inspect the submitted values, calculated outputs, map area, and related documents. The bilingual interface also supports usability in the Cypriot context by providing English and Greek language options within the same application logic.

From a performance perspective, most parts of the system are lightweight. Form submission, validation, database storage, and SORA calculations are handled efficiently by the Django backend because most outputs are based on formulas, lookup tables, and stored values.

The more demanding parts are map rendering and document generation. The map depends on client-side rendering through Leaflet and geospatial processing through JavaScript libraries and static spatial datasets. Document generation is more resource-intensive because it involves DOCX template processing, placeholder replacement, file packaging, and, where required, Selenium and ChromeDriver for map screenshot capture.

The current implementation is suitable for prototype and development use. Production deployment would require stronger infrastructure choices, including a production-grade database such as PostgreSQL, secure file storage, caching, background processing for document generation, and server optimisation. These improvements would support larger datasets, more users, and more reliable operational use.

Chapter 7

Conclusion

7.1 Conclusion	112
7.2 System Limitations	113
7.3 Future Work	115

7.1 Conclusion

The aim of this thesis was to design and implement a web-based platform that supports the preparation, completion, generation, submission, and review of SORA applications. The work focused on addressing the complexity of the traditional SORA process by providing a structured digital workflow that connects application data, risk calculations, map-based operational area definition, generated documents, submitted files, and administrative review.

The developed system was successfully designed and implemented in accordance with the requirements defined in Chapter 3. It supports both SORA 2.0 and SORA 2.5 workflows, with SORA 2.5 treated as the primary current methodology. This distinction allows the platform to support the different data structures, assessment steps, and document outputs required by each version.

For applicants, the system provides a guided application workflow through a multi-step interface. Applicants can create applications, save drafts, complete required assessment sections, define the operational area on an interactive map, view calculated outputs, generate application documents, and upload signed documents for review. The system

also supports validation and feedback mechanisms that help prevent incomplete applications from being submitted as completed records.

A central contribution of the implementation is the automation of risk-related calculations. The system calculates the main values required by the implemented SORA workflows. These values are stored with the application and reused in completed views and generated documents, reducing repeated manual work and improving consistency.

The system allows the applicant to draw the intended operational area directly within the application workflow. The map component supports spatial assessment, making the relationship between the operational area and risk assessment outputs clearer than in a manual process.

The developed system also supports CAA staff through an administrative review dashboard. The dashboard centralises the review process, creating a clearer connection between the completed digital application, the submitted document, and the review decision.

The thesis demonstrates how software engineering can support a regulatory risk assessment process without replacing regulatory judgement. The system provides structure, consistency, traceability, and workflow support, while the final review decision remains the responsibility of authorised CAA staff. Therefore, the developed system achieves the aim stated in Chapter 1 by providing a functional prototype for supporting SORA application preparation and review.

7.2 System Limitations

Although the developed system implements the main SORA application and review workflow, several limitations remain. The first limitation is that the accuracy of the assessment depends on the correctness of the data provided by the applicant. The system can check whether required fields are completed and whether selected values follow the expected format, but it cannot independently verify the factual accuracy of operational descriptions, drone specifications, mitigation claims, or uploaded documents.

A second limitation is that the system depends on the continued validity of the implemented SORA methodology. If SORA guidance, calculation rules, required fields, or Civil Aviation Authority procedures change, the platform must be updated accordingly. This is especially relevant for SORA 2.5, which is treated as the primary current workflow and may continue to evolve in practice.

The map-related functionality is also limited by the accuracy and availability of the spatial data used by the system. Population-density information, restricted and reserved areas, and authorisation-required zones must remain up to date for the map-based outputs to remain reliable. If the underlying datasets are incomplete or outdated, the system may not fully reflect the operational environment.

Document generation has similar limitations. The generated outputs depend on the structure and accuracy of the DOCX templates used by the system. If official document requirements change, the templates and placeholder logic must be revised. Some supporting SORA 2.5 document templates are also prepared for further integration and are not populated to the same level as the main generated document.

The system has not yet been evaluated through a formal user study with a larger group of UAS operators or Civil Aviation Authority staff. Further testing would be needed to assess usability, clarity, efficiency, and review effectiveness in real operational conditions.

The current implementation is also a prototype and would require additional work before production deployment. First, it does not integrate with CY Login or the GovCY Digital Services Factory authentication infrastructure. Official identity verification is therefore not performed through a governmental identity provider. This also includes stronger database infrastructure, secure file storage, deployment hardening, audit logging, backup mechanisms, performance optimisation, and broader security testing.

Finally, the system supports staff decision-making but does not make regulatory decisions independently. Civil Aviation Authority staff remain responsible for reviewing submitted material and making the final acceptance or rejection decision.

7.3 Future Work

Future work could first focus on integrating digital signatures into the submission workflow. At present, applicants generate documents, sign or convert them externally, and upload the signed PDF back into the platform. A secure digital signature process would reduce manual handling and make the submission workflow more complete.

A production version should integrate with CY Login or the GovCY Digital Services Factory authentication infrastructure, most likely through OpenID Connect. This would allow user registration and identity verification to be handled by the official government identity provider while the platform maintains the required local role and application data.

The platform could also be integrated with official CAA systems. This could support direct submission, official application numbering, reviewer assignment, status synchronisation, and formal record management.

Further user testing should be conducted with UAS operators, applicants, and CAA staff. Feedback from real users would help identify unclear terminology, interface issues, validation gaps, and possible improvements to the review workflow.

The notification system could also be extended. Future versions could include improved email templates, in-platform notifications, reminders for unfinished drafts, alerts for review decisions, and expiry-related notifications.

Document management could be improved through versioning. Generated documents, uploaded signed PDFs, revised submissions, and review decisions could be stored as a document history rather than only as the latest submitted file. This would improve traceability when an application is rejected, corrected, and resubmitted.

Further work is also needed for the additional SORA 2.5 document templates. The current implementation supports multiple generated outputs, but future development could complete the placeholder population and full integration of the operational authorisation and compliance matrix templates.

Additional technical improvements could include migration from SQLite to PostgreSQL, background processing for document generation, caching for geospatial data and lookup tables, production-grade file storage, and deployment optimisation. These improvements would support larger datasets, more users, and more reliable production use.

References

- [1] European Union Aviation Safety Agency, Easy Access Rules for Unmanned Aircraft Systems, Revision from July 2024. Cologne: EASA, 2024.
- [2] Department of Civil Aviation, Republic of Cyprus, “SORA Risk Assessment Ver2.0.” Available: <https://drones.gov.cy/sora-risk-assessment/>.
- [3] Department of Civil Aviation, Republic of Cyprus, “SORA Risk Assessment Ver2.5.” Available: <https://drones.gov.cy/sora-risk-assessment-3/>.
- [4] European Union Aviation Safety Agency, Specific Operation Risk Assessment SORA, SORA Workshop 2023 presentation material, 2023.
- [5] Joint Authorities for Rulemaking on Unmanned Systems, SORA v2.0 to v2.5: Summary of Changes for External Consultation, 2023.
- [6] Y. Grigoriou, A. Savva, P. Kolios, M. Chatzipanagiotis, and S. Timotheou, “Enhancing the Safety of Multi-UAS Urban Operations with SORA,” in 2024 International Conference on Unmanned Aircraft Systems (ICUAS), Chania, Greece, 2024.