

# Sunward Portal Management System

## DOCUMENT INFORMATION

Prepared by	Appcentric Technologies
Client	Sunward
Project Name	Sunward Portal Management System
Prepared by	Arun
Module Name	System Design Document Module
Pages	9

## Disclaimer and Confidentiality Statement

This document is a technical reference prepared by **Appcentric Technologies** for the **Sunward Portal Management System** project. It provides detailed information on the system architecture, technical approach, implementation methodology, and related technical specifications.

This document is **confidential** and is intended solely for the internal use of the authorized technical team of **Sunward**.

No part of this document may be copied, modified, distributed, or disclosed to any third party without prior written approval from **Appcentric Technologies**.

Although every effort has been made to ensure the accuracy and reliability of the technical information contained in this document, **Appcentric Technologies** makes no warranties, express or implied, regarding its completeness, accuracy, or suitability for any specific purpose. Users of this document are advised to perform the necessary verification and testing before acting upon any technical details provided herein.

AppCentric  
Technologies

## Table of Contents

Sunward Portal Management System .....	1
1. Project Overview .....	3
2. Objectives .....	4
3. System Modules & Key Functions .....	5
3.1 Onion Architecture Layers (Main Solution).....	5
3.2 Additional Projects in the Solution .....	6
4. Functional Components .....	7
4.1 User Authentication & Role-Based Access .....	7
4.2 Dashboard & Widgets.....	7
4.3 Document Module (FileAPI + DocumentViewer) .....	7
4.4 Dynamic Form Module.....	7
4.4 Reporting Module .....	7
4.4 Portal Settings .....	8
5. Business Logic.....	8
5.1 CQRS Pattern Implementation.....	8
5.2 Repository Pattern .....	8
5.3 Validation Rules.....	8
5.4 Document Control Logic.....	8
6. Summary .....	8

AppCentric  
Technologies

## 1. Project Overview

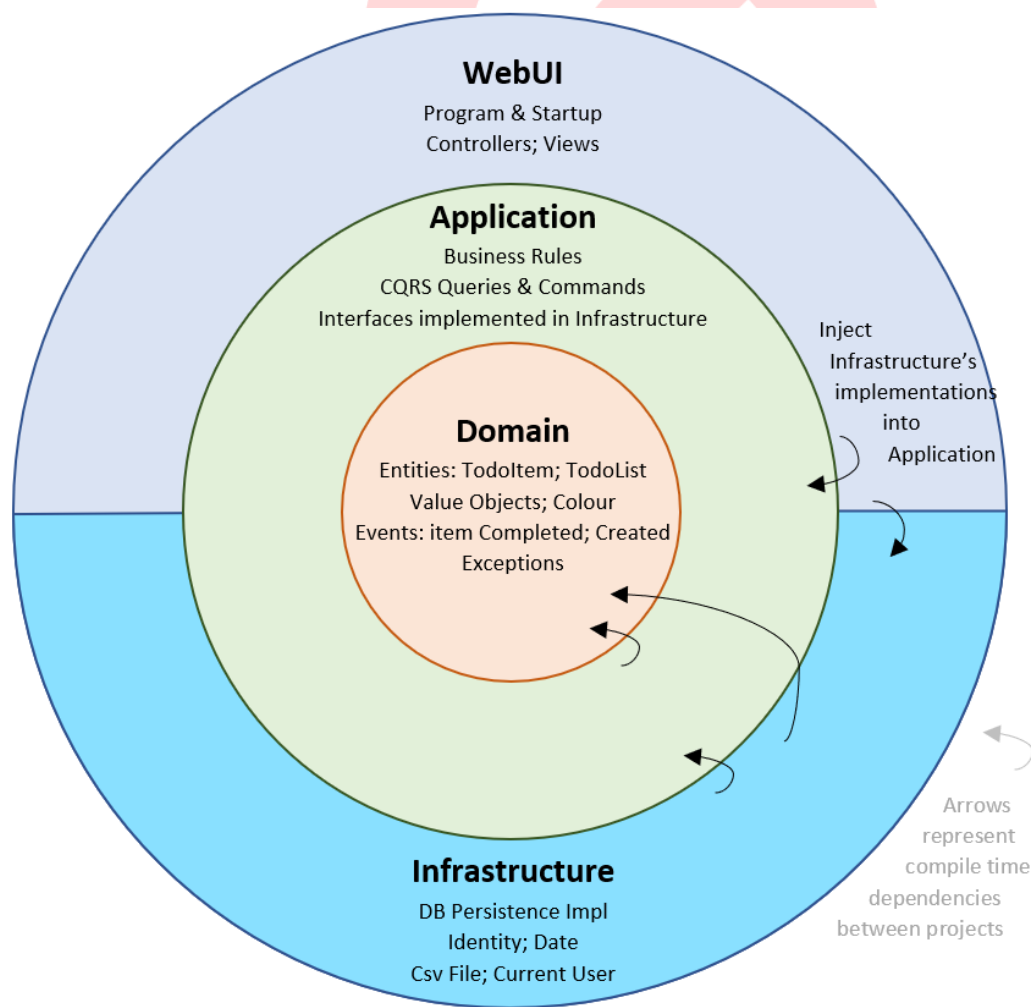
The **Sunward Portal Management System** is a modern enterprise web platform designed using **Onion Architecture**, **.NET 8**, **C#**, and **DevExpress Blazor**.

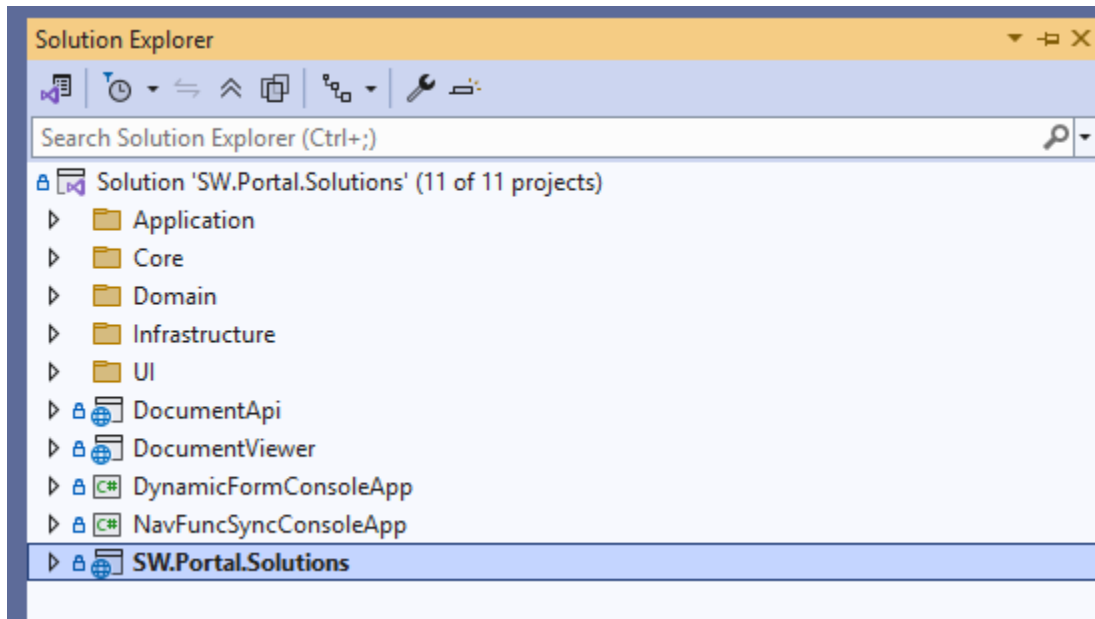
The solution integrates multiple modules such as **Document Viewer**, **Document API**, **Dynamic Form Console**, and **Navigation Functional Sync**, allowing centralized process automation for Sunward operations.

### The project supports:

1. High-performance .NET 8 backend
2. Modular Onion architecture (Application, Core, Domain, Infrastructure, UI)
3. DevExpress Blazor rich UI
4. SQL Server as the main database
5. Clean, scalable, testable structure

### Solution Architecture Screenshot





## 2. Objectives

The primary objectives of the Sunward Portal Management System are:

### 1. Centralized Web Platform

Provide a single, unified portal for internal operations, document management, and forms.

### 2. Modular & Scalable Architecture

Using **Onion Architecture** ensures that business logic is independent from UI and infrastructure, making it scalable.

### 3. High Performance & Security

Built on .NET 8 to support faster execution, secure API endpoints, JWT authentication, and efficient DB queries.

### 4. Simplified Document Management

Using **DocumentAPI**, and **DocumentViewer** modules to manage all Sunward documents.

### 5. Consistent UI/UX

DevExpress Blazor components provide a fast, responsive user interface with consistent styling.

### 3. System Modules & Key Functions

#### 3.1 Onion Architecture Layers (Main Solution)

##### Domain Layer

- Contains **Entities, Value Objects, Enums**
- Pure business rules
- No dependency on external libraries

##### Application Layer

- Contains **CQRS Handlers, Interfaces, Business Rules**
- Includes application services for:
  - User management
  - Document operations
  - Form workflows
  - Notification routines

##### Infrastructure Layer

- Responsible for:
  - SQL Server database
  - EF Core repositories
  - Email service
  - File storage
  - Logging

##### UI Layer (DevExpress Blazor)

- Main Blazor Web App
- Razor pages, components
- DevExpress controls:
  - DxDataGrid
  - DxFormLayout
  - DxPopup
  - DxDateEdit / DxMaskedInput
  - DxScheduler

##### Core Layer

- Shared utilities
- DTOs
- Extensions
- Helpers

## 3.2 Additional Projects in the Solution

### **DocumentApi**

- Provides API endpoints for:
- Upload
- Preview
- Download
- Delete operations
- Used by FileViewer and SWPortal

### **DocumentViewer (Blazor Module)**

- Displays PDF/Image/Docs
- Integrated file previewer
- Controlled access based on permission

### **DynamicFormConsoleApp**

- Executes background tasks
- Processes dynamic forms and data sync

### **NavFuncSyncConsoleApp**

- Synchronizes navigation, permissions, roles across environments

AppCentric  
Technologies

## 4. Functional Components

Below are the core functional components of the system.

### 4.1 User Authentication & Role-Based Access

- Login
- Role-based UI rendering
- Permission-based menu filtering
- Integrated with SWPortal database

### 4.2 Dashboard & Widgets

- Real-time dashboard
- DevExpress Charts
- Statistics (Forms, Pending Tasks, Reports)

### 4.3 Document Module (FileAPI + DocumentViewer)

**Features:**

- Upload documents
- Version management
- Online viewing (PDF, Excel, Word)
- Secure storage
- Activity logs

### 4.4 Dynamic Form Module

- Create custom forms
- Approval workflow
- Auto validation rules
- Audit trail

### 4.4 Reporting Module

- DevExpress Reports
- Export: PDF, Excel
- Scheduled reports

#### 4.4 Portal Settings

- User management
- Role assignment
- Email configuration (SMTP)
- Application variables

### 5. Business Logic

The core business logic is implemented within **Application Layer** and **Domain Layer**.

#### 5.1 CQRS Pattern Implementation

- Query → fetch data
- Command → create, update, delete

#### 5.2 Repository Pattern

- All DB interaction via repositories
- EF Core used for abstraction

#### 5.3 Validation Rules

- Fluent Validation
- Custom validation attributes (e.g., Name)

#### 5.4 Document Control Logic

- Secure file paths
- Checking user rights before viewing
- File encryption (optional)

### 6. Summary

The **Sunward Portal Management System** is a robust, modular, enterprise-grade application developed using:

- **Onion Architecture**
- **.NET 8 + C#**
- **DevExpress Blazor**
- **SQL Server**



- **API + Blazor Hybrid Components**

The system structure ensures:

- ✓ High maintainability
- ✓ Reusable business logic
- ✓ Strict separation of concerns
- ✓ Secure document management
- ✓ Scalable modular design

It contains multiple integrated modules such as:

- SWPortal (Live)
- SWPortalUAT (Dev)
- FileAPI
- DocumentViewer
- DocumentAPI
- Dynamic Form Console
- Navigation Sync Console

This architecture and framework combination ensures long-term stability and ease of enhancements.

AppCentric  
Technologies