

# Sunward Portal Management System

## DOCUMENT INFORMATION

---

Prepared by	Appcentric Technologies
Client	Sunward
Project Name	Sunward Portal Management System
Prepared by	Arun
Module Name	Dynamic From Module
Pages	17

---

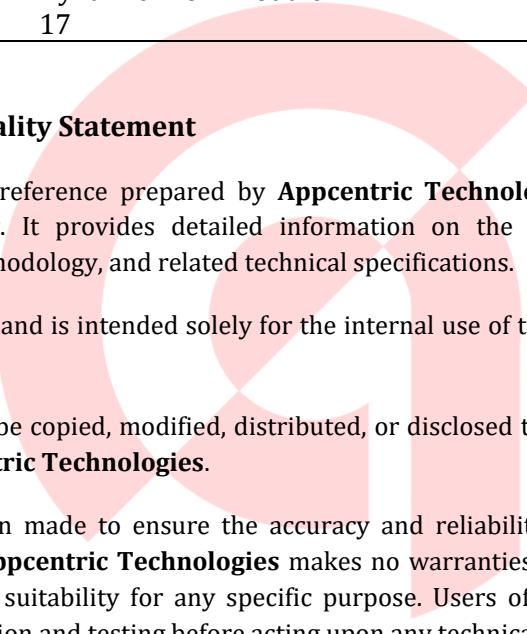
## Disclaimer and Confidentiality Statement

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 .....	4
2. Objectives .....	5
2.1 Enable Configurable Forms Without Code Changes.....	5
2.2 Standardize Form Creation Across the Project.....	5
2.3 Improve Development Speed & Reduce Maintenance .....	5
2.4 Support Complex Business Rules Dynamically .....	5
2.5 Implement Robust Security Controls .....	5
2.6 Maintain Complete Audit Trail & Version History .....	5
2.7 Provide a Unified User Experience.....	6
3. System Modules & Key Functions .....	6
3.1 Attribute Master .....	6
3.2 Dynamic Form .....	7
3.2.1 Dynamic Rendering Engine (Frontend) .....	7
3.2.2 Data Management Module .....	7
3.2.3 Workflow & Release Module .....	8
3.2.4 Security & Role Management Module .....	8
3.2.5 Audit & Logging Module .....	8
4.Functional Components .....	8
4.1 Form Definition & Configuration .....	8
4.2 Dynamic UI Rendering .....	10
4.3 Data Capture & Persistence.....	11
4.4 Workflow & Release Management .....	11
4.5 Security & Access Control .....	11
4.6 Audit & History .....	12
4.6 Reporting & Export.....	12
5. Business Logic.....	12
5.1 Form Lifecycle Logic .....	12
5.2 Validation & Rule Execution .....	13
5.3 Security & Permission Logic .....	13
5.4 Release & Approval Logic .....	13

5.5 Audit & Versioning Logic.....	14
5.6 Auto-Numbering & Derived Values.....	14
5.7 Integration & Consistency Rules .....	14
6. Summary .....	17



**AppCentric  
Technologies**

## 1. Project Overview

A Dynamic Form is a flexible form system where:

- Form structure (Sections, Fields, Validation, Rules) is not hard-coded.
- Everything is stored in database tables and rendered at runtime.
- Admins can add or modify fields without deploying code.

This allows the project to support any number of custom forms for different workflows.

### ◆ Form Builder (Admin)

- Create Sections (Header → Questions → Options)
- Add field types (Textbox, Date, Dropdown, File Upload, Checkbox...)
- Validation Rules (Required, Min/Max, Regex, Visibility rules)
- Sorting & Section arrangement

### ◆ Dynamic Rendering (Frontend)

- Client UI loads structure from DB and **builds form automatically**
- Conditional visibility
- Repeatable sections
- Auto-numbering
- Custom templates (DxGrid, DxForm, etc.)

### ◆ Data Management

- Save data in JSON or relational tables
- Auto-versioning
- View, Edit, Compare previous versions

### ◆ Security & Roles

- Section-level permissions:
- View / Edit / Hide
- Release & Approve workflow

### ◆ Audit Trail

- Tracks:
- Before/After values

- Released by / Modified by
- Timestamps
- Used for compliance and reporting

## 2. Objectives

### 2.1 Enable Configurable Forms Without Code Changes

Create a flexible engine where admins can design forms (sections, fields, rules) from the database—eliminating the need for UI or backend code modifications.

### 2.2 Standardize Form Creation Across the Project

Provide a **single, reusable framework** to manage all forms instead of building separate screens for each requirement.

### 2.3 Improve Development Speed & Reduce Maintenance

- Decrease development time for new forms
- Reduce bugs caused by duplicated form logic
- Lower long-term maintenance cost

### 2.4 Support Complex Business Rules Dynamically

Allow:

- Field-level validation
  - Conditional visibility
  - Auto-numbering
  - Dynamic dropdowns
  - Repeatable sections
- All controlled through metadata.

### 2.5 Implement Robust Security Controls

Provide **role-based and section-level access**:

- View / Edit / Hide
- Release / Approve workflow

Ensuring sensitive form data is accessed only by authorized users.

### 2.6 Maintain Complete Audit Trail & Version History

Track:

- Before & After values
- Who changed what

- Timestamps
- Released versions

For compliance, accountability, and troubleshooting.

## 2.7 Provide a Unified User Experience

Ensure:

- Consistent UI across all forms
- Data integrity
- Smooth dynamic rendering
- Integration with DevExtreme components (DxForm, DxGrid)

## 3. System Modules & Key Functions

The DMS consists of **two main components**:

### 3.1 Attribute Master

The **Attribute Master** is a core component of the Dynamic Form engine. It defines **each field/question** in a form using metadata so the UI can be rendered dynamically without writing code.

#### Purpose of Attribute Master

The Attribute Master stores the configuration of every field in the system, such as:

- Field type (Textbox, Dropdown, Date, Number, File Upload, Checkbox...)
- Label & Display name
- Data type
- Validation rules
- Default values
- UI behaviour (show/hide, read-only, mandatory)
- Lookup / option sources
- Sorting order
- Section mapping

This allows the system to generate UI controls dynamically.

#### Business Logic Related to Attribute Master

##### ✓ Rendering

- UI reads metadata → displays appropriate control.

- Validation rules provided by attribute metadata.

## ✓ Validation

- Required fields must be filled.
- Value must match datatype.
- Regex must pass if configured.

## 3.2 Dynamic Form

- Create and manage **Forms, Sections, and Attributes**
- Configure field types (Textbox, Dropdown, Date, File Upload, etc.)
- Define:
  - Validation rules
  - Mandatory / Optional fields
  - Conditional visibility rules
  - Auto-numbering settings
- Manage dropdown options and dynamic lookups
- Set sorting, grouping, and display order

### 3.2.1 Dynamic Rendering Engine (Frontend)

- Convert metadata from DB → actual UI elements
- Render DevExpress components dynamically (DxForm, DxGrid, etc.)
- Handle:
  - Conditional show/hide
  - Inline validation
  - Repeatable/Nested sections
  - Custom templates
- Support responsive UI across modules

### 3.2.2 Data Management Module

- Save user-entered data
- Store data in JSON or relational format
- Read, update, and delete form instances
- Manage versioning of form data
- Retrieve and display historical values

### 3.2.3 Workflow & Release Module

- Section-level release/approval
  - Track:
    - Released by
    - Date & time
    - Current status (Draft / Submitted / Released)
  - Prevent editing after release
  - Multi-step approval support (if required)
- 

### 3.2.4 Security & Role Management Module

- User-level access control
  - Section visibility permissions:
    - **View**
    - **Edit**
    - **Hidden**
  - Attribute-level access (optional)
  - Integration with application-wide authentication
- 

### 3.2.5 Audit & Logging Module

- Track changes at:
    - Form level
    - Section level
    - Field level
  - Store before/after values
  - Maintain full history per change
  - Export audit logs for reporting
- 

## 4. Functional Components

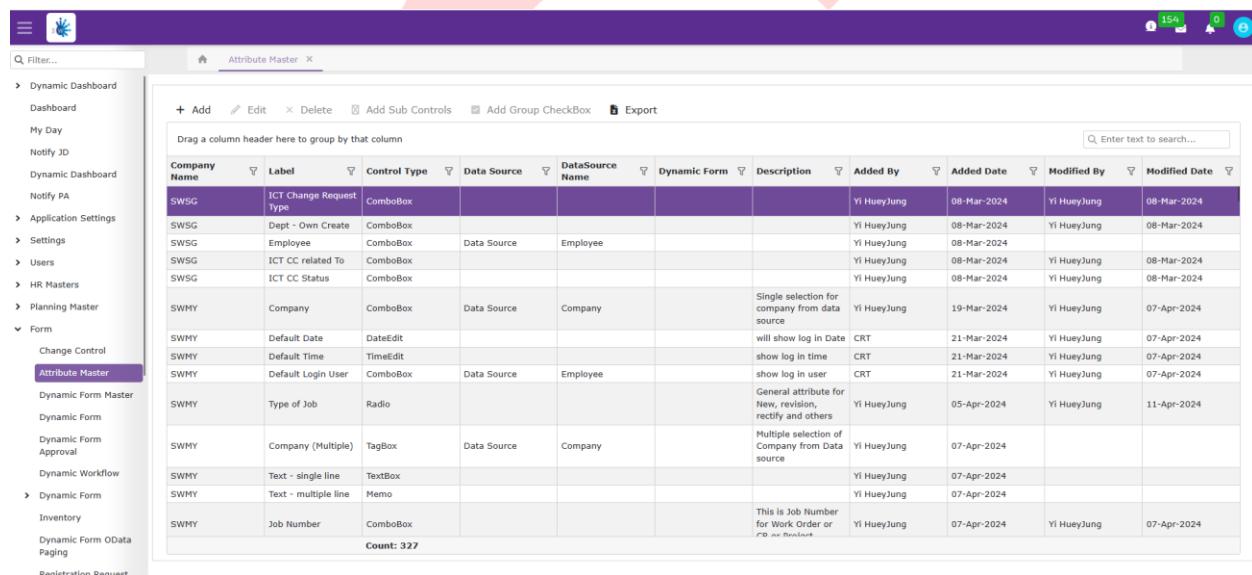
The Dynamic Form system is composed of the following functional components:

### 4.1 Form Definition & Configuration

**Purpose:** Allow admins to design and manage forms without code changes.

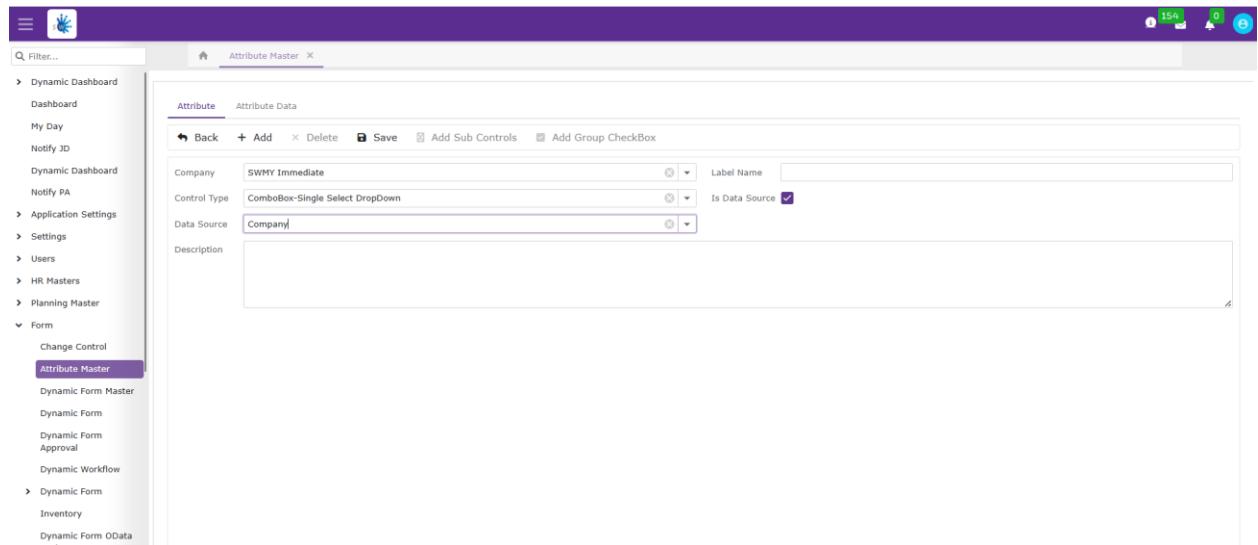
**Components:**

- **Form Designer**
  - Create / edit forms
  - Configure metadata (name, code, category, status)
- **Section Manager**
  - Add / order sections
  - Set instructions, visibility, sort order
- **Field (Attribute) Manager**
  - Define field type (text, number, date, dropdown, checkbox, grid, file upload, etc.)
  - Set default values, placeholders, help text
  - Configure validation rules (required, range, regex, max length)
- **Option / Lookup Configuration**
  - Manage static dropdown values
  - Configure dynamic lookups from master tables or APIs

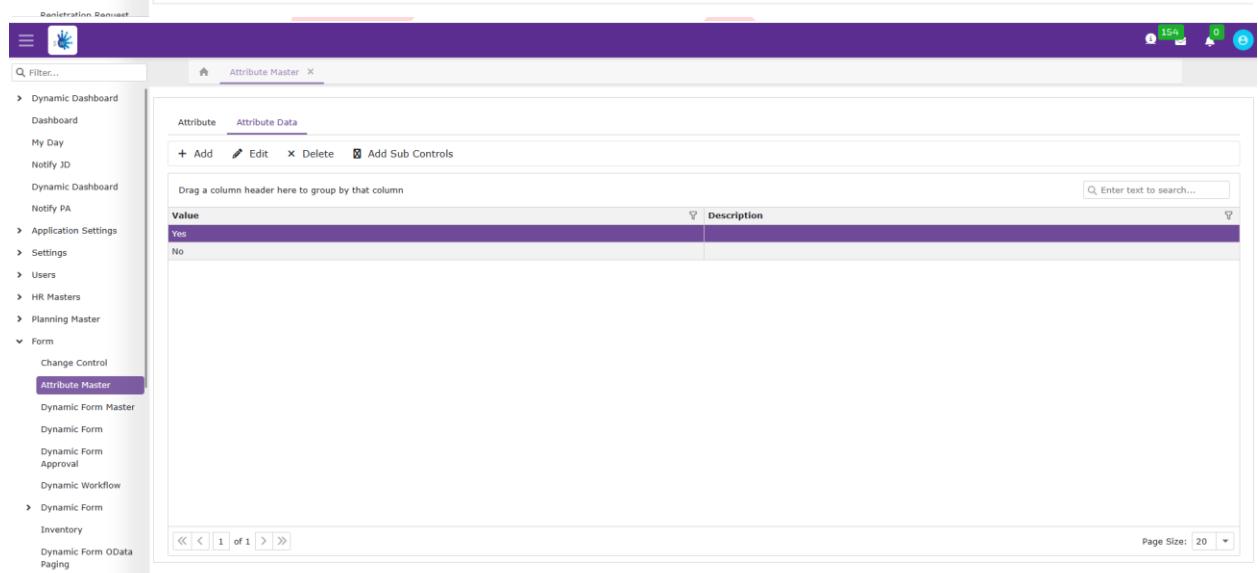


The screenshot shows the 'Attribute Master' screen within the AppCentric platform. The left sidebar contains a navigation menu with items like 'Dynamic Dashboard', 'Dashboard', 'My Day', 'Notify JD', 'Dynamic Dashboard', 'Notify PA', 'Application Settings', 'Settings', 'Users', 'HR Masters', 'Planning Master', 'Form', 'Change Control', 'Attribute Master' (which is selected and highlighted in purple), 'Dynamic Form Master', 'Dynamic Form', 'Dynamic Form Approval', 'Dynamic Workflow', 'Dynamic Form', 'Inventory', and 'Dynamic Form OData Paging'. The main content area is titled 'Attribute Master' and displays a table with 12 columns. The columns are: Company Name, Label, Control Type, Data Source, DataSource Name, Dynamic Form, Description, Added By, Added Date, Modified By, and Modified Date. The table lists various attributes such as 'ICT Change Request Type', 'Dept - Own Create', 'Employee', 'ICT CC related To', 'ICT CC Status', 'Company', 'Default Date', 'Default Time', 'Default Login User', 'Type of Job', 'Company (Multiple)', 'Text - single line', 'Text - multiple line', and 'Job Number'. Each row includes a small description of the attribute's purpose in the 'Description' column. The bottom of the table shows a total count of 327.

AppCentric  
Technologies



The screenshot shows the 'Attribute Master' configuration page for a 'ComboBox-Single Select DropDown'. The 'Control Type' is set to 'ComboBox-Single Select DropDown' and the 'Data Source' is set to 'Company'. The 'Label Name' field is empty.

The screenshot shows the 'Attribute Data' list view. It displays two rows: 'Yes' and 'No'. The 'Value' column contains 'Yes' and 'No', and the 'Description' column is empty. A search bar at the top right is empty.

## 4.2 Dynamic UI Rendering

**Purpose:** Convert metadata from DB into runtime UI.

**Components:**

- **Metadata Loader**
  - Fetch form, section, field, and rule definitions
- **Control Generator**
  - Map field type → UI control (e.g., textbox, date picker, DxGrid)
  - Apply formatting and layout
- **Rule Engine (Client Side)**
  - Handle show/hide logic
  - Enable/disable fields based on other values
  - Trigger client-side validations

---

#### 4.3 Data Capture & Persistence

**Purpose:** Manage user-entered data for each form instance.

**Components:**

- **Data Binding Layer**
    - Binds UI controls to in-memory data model
  - **Save / Update Handler**
    - Create new form instances
    - Update existing records (respecting release/lock status)
  - **Versioning Component**
    - Create new versions on significant change or release
    - Maintain previous versions for comparison and audit
  - **Data Storage Component**
    - Save data as JSON and/or normalized relational records
- 

#### 4.4 Workflow & Release Management

**Purpose:** Control the lifecycle of a form (Draft → Submitted → Released).

**Components:**

- **Status Manager**
    - Manage states: Draft, In Progress, Submitted, Released, Rejected, etc.
  - **Section Release Component**
    - Handle section-level release (per user/role)
    - Lock released sections from further edits
  - **Approval Component (if multi-level)**
    - Route forms to approvers
    - Capture approval/rejection remarks
- 

#### 4.5 Security & Access Control

**Purpose:** Ensure only authorized users can view or edit specific parts of a form.

**Components:**

- **Authentication Integration**
  - Use existing application login/session
- **Authorization Engine**
  - Map users/roles to section permissions (View / Edit / Hide)
  - Optional field-level restrictions
- **Permission Evaluation**
  - Enforce security at:

- Metadata load time (don't send restricted sections)
  - UI rendering (hide/disable fields)
  - Save operations (server-side recheck)
- 

## 4.6 Audit & History

**Purpose:** Provide complete traceability of changes.

**Components:**

- **Change Tracker**
    - Detect field-level changes
    - Capture old vs new values
  - **Audit Logger**
    - Store user, timestamp, action (Create, Update, Release, Approve)
  - **History Viewer**
    - Display previous versions
    - Compare two versions (highlight changed fields)
- 

## 4.6 Reporting & Export

**Purpose:** Support analytics and data extraction.

**Components:**

- **Query & Filter Component**
  - Filter records by date, status, user, etc.
- **Export Component**
  - Export to Excel / PDF / CSV
- **Summary / Dashboard Widgets**
  - Show counts, pending approvals, released forms, etc.

## 5. Business Logic

The Dynamic Form system implements business rules that govern how forms are created, validated, secured, approved, and stored. The core logic is metadata-driven so that behavior can be controlled without code changes.

---

### 5.1 Form Lifecycle Logic

- Each form instance follows a controlled lifecycle:

- Draft → In Progress → Submitted → Released → (Optional) Rejected / Reopened
  - Transitions between statuses are governed by:
    - User role (creator, reviewer, approver)
    - Section-wise completion and validation status
  - Once a form or section is **Released**, further edits are restricted based on configuration (read-only or allowed only via new version).
- 

## 5.2 Validation & Rule Execution

- Field-level validation rules are stored as metadata (e.g., required, min/max, regex, allowed values).
  - At runtime:
    - **Client-side validation** ensures immediate feedback to the user.
    - **Server-side validation** re-validates before saving to prevent tampering.
  - **Business rules** can:
    - Auto-calculate values (e.g., age from date of birth, totals from line items).
    - Ensure cross-field consistency (e.g., Start Date ≤ End Date, dependent field not empty if parent is selected).
  - Conditional rules control:
    - Show/hide of sections and fields.
    - Enable/disable of controls based on other field values.
- 

## 5.3 Security & Permission Logic

- Access is evaluated at:
    - **Form level** – who can create/view instances.
    - **Section level** – View / Edit / Hidden per user/role.
  - During metadata load, restricted sections/fields are either:
    - Not sent to the client at all, or
    - Marked read-only / hidden on UI.
  - On save/release, server-side checks ensure:
    - Users cannot modify sections they are not allowed to edit.
    - Unauthorized updates are blocked even if client is manipulated.
- 

## 5.4 Release & Approval Logic

- Section-level release logic:
  - A user with appropriate role can **release** a section once mandatory fields are valid.
  - After release, the section becomes locked (read-only) for normal users.

- Form-level release:
    - All required sections must be in a valid state before final release.
    - Optional **multi-level approval** can be configured (e.g., Maker–Checker).
  - Each release/approval action:
    - Captures user, timestamp, remarks, and status.
    - Updates the workflow state for reporting and tracking.
- 

## 5.5 Audit & Versioning Logic

- Every significant change (create, update, release, approval) generates:
    - An **audit record** with old and new values.
    - A **version** of the form data when configured (e.g., on release).
  - The system:
    - Maintains version numbers (v1, v2, v3...).
    - Allows viewing and comparison of any two versions.
  - Audit logic ensures:
    - Traceability: who changed what and when.
    - Compliance with organizational and regulatory requirements.
- 

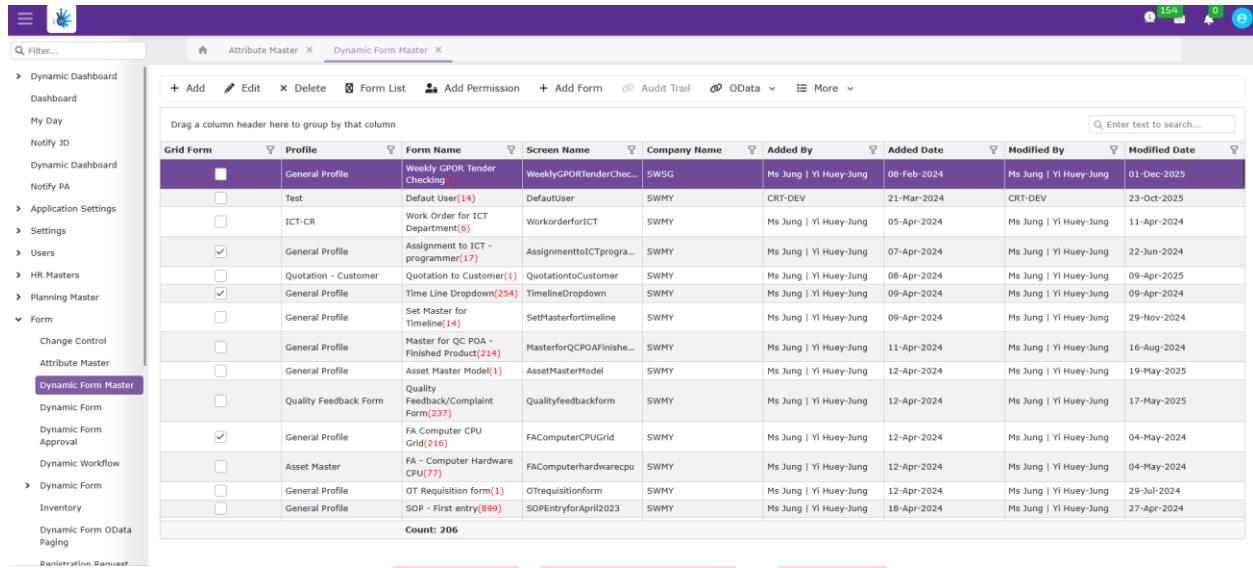
## 5.6 Auto-Numbering & Derived Values

- Certain fields (like reference numbers, case IDs, document numbers) are auto-generated based on:
    - Configured patterns (prefix, suffix, running number).
    - Context info (year, branch, department).
  - Derived fields are computed using:
    - Configured expressions (e.g., sum of child rows, date differences).
    - Or domain-specific logic in the service layer.
- 

## 5.7 Integration & Consistency Rules

- When integrated with other modules/systems:
  - Lookup values (e.g., customers, products, users) are validated against master data.
  - Business logic prevents inconsistent states (e.g., referencing inactive master records).
- On successful release:
  - Optional triggers can send data/events to external systems (via API calls, queues, etc.).
- Error handling ensures:
  - Transactions are rolled back on failure.

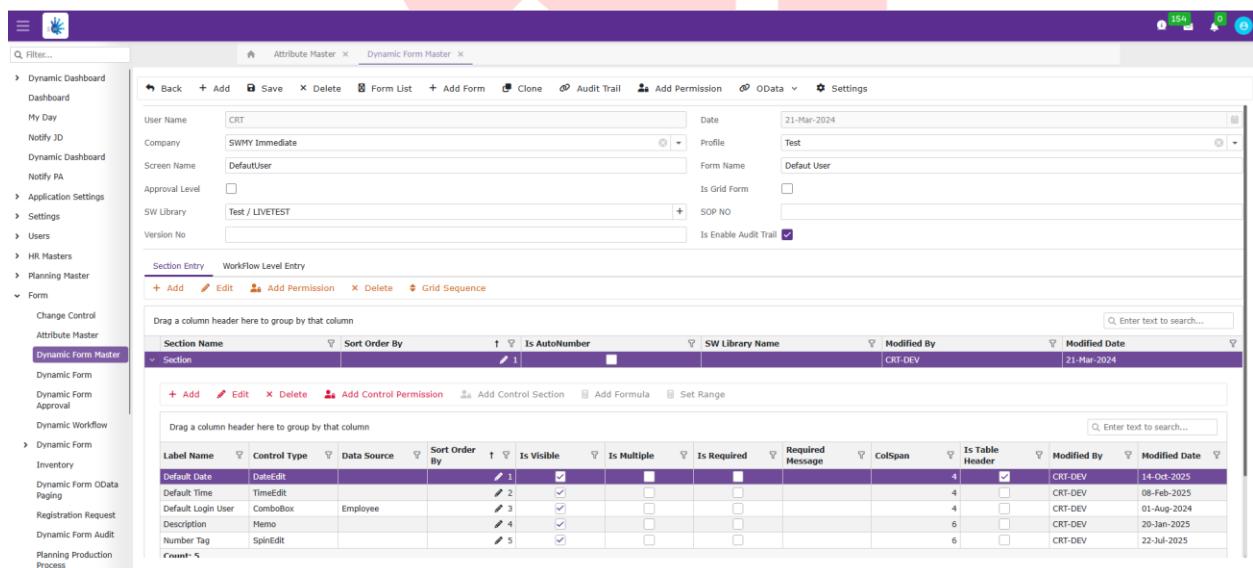
- User receives clear messages without exposing internal errors.



This screenshot shows a grid view of form entries. The columns include Grid Form, Profile, Form Name, Screen Name, Company Name, Added By, Added Date, Modified By, and Modified Date. The data shows various profiles like General Profile, ICT-CR, and Asset Master, along with their respective screen names, company names, and dates added or modified.

Grid Form	Profile	Form Name	Screen Name	Company Name	Added By	Added Date	Modified By	Modified Date
Test	General Profile	Weekly GPOI Tender Checking (1)	WeeklyGPORTenderChec...	SWSG	Ms Jung   Yi Huey-Jung	08-Feb-2024	Ms Jung   Yi Huey-Jung	01-Dec-2025
	Default User (14)	DefaultUser	SWMY	CRT-DEV	21-Mar-2024	CRT-DEV	23-Oct-2025	
	ICT-CR	Work Order for ICT Department (6)	WorkorderforICT	SWMY	Ms Jung   Yi Huey-Jung	05-Apr-2024	Ms Jung   Yi Huey-Jung	11-Apr-2024
	General Profile	Assignment to ICT programmer (17)	AssignmenttoICTprogra...	SWMY	Ms Jung   Yi Huey-Jung	07-Apr-2024	Ms Jung   Yi Huey-Jung	22-Jun-2024
	Quotation - Customer	Quotation to Customer (1)	QuotationtoCustomer	SWMY	Ms Jung   Yi Huey-Jung	08-Apr-2024	Ms Jung   Yi Huey-Jung	09-Apr-2025
	General Profile	Time Line Dropdown (254)	TimelineDropdown	SWMY	Ms Jung   Yi Huey-Jung	09-Apr-2024	Ms Jung   Yi Huey-Jung	09-Apr-2024
	General Profile	Set Master for Timeline (14)	SetMasterfortimeline	SWMY	Ms Jung   Yi Huey-Jung	09-Apr-2024	Ms Jung   Yi Huey-Jung	29-Nov-2024
	General Profile	Master for QC POA - Finished Product (214)	MasterforQCPOAFinishe...	SWMY	Ms Jung   Yi Huey-Jung	11-Apr-2024	Ms Jung   Yi Huey-Jung	16-Aug-2024
	General Profile	Asset Master Model (1)	AssetMasterModel	SWMY	Ms Jung   Yi Huey-Jung	12-Apr-2024	Ms Jung   Yi Huey-Jung	19-May-2025
	Quality Feedback Form	Quality Feedback/Complaint Form (237)	Qualityfeedbackform	SWMY	Ms Jung   Yi Huey-Jung	12-Apr-2024	Ms Jung   Yi Huey-Jung	17-May-2025
	General Profile	FA Computer CPU Grid (216)	FAComputerCPUGrid	SWMY	Ms Jung   Yi Huey-Jung	12-Apr-2024	Ms Jung   Yi Huey-Jung	04-May-2024
	Asset Master	FA - Computer Hardware CPU (77)	FAComputerhardwarecpu	SWMY	Ms Jung   Yi Huey-Jung	12-Apr-2024	Ms Jung   Yi Huey-Jung	04-May-2024
	General Profile	OT Requisition form (1)	OTRequisitionform	SWMY	Ms Jung   Yi Huey-Jung	12-Apr-2024	Ms Jung   Yi Huey-Jung	29-Jul-2024
	General Profile	SOP - First entry (899)	SOPEntryforApril2023	SWMY	Ms Jung   Yi Huey-Jung	18-Apr-2024	Ms Jung   Yi Huey-Jung	27-Apr-2024

Count: 206



This screenshot shows a form entry creation screen. It includes fields for User Name, Company, Screen Name, Approval Level, SW Library, Version No., and various checkboxes for audit trail and enable audit trail. Below this is a section entry table with columns for Section Name, Sort Order By, Is AutoNumber, SW Library Name, Modified By, and Modified Date.

Section Name	Sort Order By	Is AutoNumber	SW Library Name	Modified By	Modified Date
Section	1		CRT-DEV		21-Mar-2024

Count: 5

Attribute Master > Dynamic Form Master

Assignment to ICT - programmer

Drag a column header here to group by that column

Q Enter text to search...

No	Profile No	Date of Entry	Assignment Job Type	Job Number	Job Description	Assigned staff - Company	Employee	Expected Completion date for assignee	Task Status	Modified By
17	GP-24-044436	02-Jul-2024	Request to ICT department	J51Job No 5	Refers to Email	SWHY Sunward Malaysia	Chin Hwa Mr Wong   Wong			Ms Jung   Yi Huey-Jung
14	GP-24-043148	22-Jun-2024	Request to ICT department	J21Job No 2	Refers to Email	SWHY Sunward Malaysia	Chee Hen Chee Hen   Loh	25-Jun-2024	Approval HOD approval	Mr Wong   Chin Hwa-Wong
13	GP-24-043147	22-Jun-2024	Request to ICT department	J11Job No 1	Refers to Email	SWHY Sunward Malaysia	Chee Hen Chee Hen   Loh	25-Jun-2024	Approval HOD approval	Mr Wong   Chin Hwa-Wong

Count: 3

Attribute Master > Dynamic Form Master

Assignment to ICT - programmer

Form > Work Flow

Profile No

Login From: CRTDEV Login Company: SWMY Added By: CRTDEV Added Date: 05-Dec-2025 02:51 PM

Profile No:

Company: Select a Company... Division: Select a Division...

Department: Select a Department... Section: Select a Section...

SubSection: Select a Sub Section...

1. Assignment to ICT - programmer

Date of Entry: 12/5/2025

Assignment Job Type: Select Assignment Job Type.. Job Number: Select Job Number..

Job Description: Refers to Email In this Form

Employee: Select... Assigned staff - Company: Select Assigned staff - Company..

Task Status: Select Task Status.. Expected Completion date for assignee: Select Date..

# Technologies

Screenshot of the Dynamic Form Audit module interface. The left sidebar shows navigation links for Dynamic Dashboard, Notify PA, Application Settings, Settings, Users, HR Masters, Planning Master, Form, Change Control, Attribute Master, Dynamic Form Master, Dynamic Form, Dynamic Form Approval, Dynamic Workflow, Dynamic Form, Inventory, Dynamic Form OData Paging, Registration Request, Dynamic Form Audit, Planning Production Process, Scheduler, DMS, Reports, and Email.

The main content area displays a grid of audit logs. The columns include Deleted, Profile No, Prev User, Prev Update Date, Audit User, and Audit Date. A search bar at the top right allows users to enter text to search for specific entries.

Deleted	Profile No	Prev User	Prev Update Date	Audit User	Audit Date
	SWMY-QF-00002	Wan Nor AzierahBinti Wan Mohd Nasir	10-Jun-2025 04:05:24 PM	Wan Nor AzierahBinti Wan Mohd Nasir	27-Oct-2025 10:01:00 AM
	SWMY-QF-00002	Wan Nor AzierahBinti Wan Mohd Nasir	19-May-2025 04:22:10 PM	Wan Nor AzierahBinti Wan Mohd Nasir	10-Jun-2025 04:05:24 PM
	SWMY-QF-00002	Wan Nor AzierahBinti Wan Mohd Nasir	19-May-2025 02:09:08 PM	Wan Nor AzierahBinti Wan Mohd Nasir	19-May-2025 04:22:10 PM
	SWMY-QF-00002	Wan Nor AzierahBinti Wan Mohd Nasir	19-May-2025 01:56:08 PM	Wan Nor AzierahBinti Wan Mohd Nasir	19-May-2025 02:09:08 PM
	SWMY-QF-00002	Wan Nor AzierahBinti Wan Mohd Nasir	19-May-2025 01:51:31 PM	Wan Nor AzierahBinti Wan Mohd Nasir	19-May-2025 01:56:08 PM
	Profile No: SWMY-QF-00003				
	SWMY-QF-00003	Wan Nor AzierahBinti Wan Mohd Nasir	10-Jun-2025 05:25:07 PM	NUR DAFINA FARISKA	11-Jul-2025 03:44:31 PM
	SWMY-QF-00003	Wan Nor AzierahBinti Wan Mohd Nasir	20-May-2025 10:01:20 AM	Wan Nor AzierahBinti Wan Mohd Nasir	10-Jun-2025 04:00:58 PM
	SWMY-QF-00003	Wan Nor AzierahBinti Wan Mohd Nasir	20-May-2025 09:59:05 AM	Wan Nor AzierahBinti Wan Mohd Nasir	20-May-2025 10:01:20 AM
	SWMY-QF-00003	Wan Nor AzierahBinti Wan Mohd Nasir	19-May-2025 06:02:27 PM	Wan Nor AzierahBinti Wan Mohd Nasir	20-May-2025 09:58:05 AM
	SWMY-QF-00003	Wan Nor AzierahBinti Wan Mohd Nasir	19-May-2025 05:59:31 PM	Wan Nor AzierahBinti Wan Mohd Nasir	19-May-2025 06:02:27 PM
	Profile No: SWMY-QF-00004				
	SWMY-QF-00004	NUR DAFINA FARISKA	18-Jun-2025 02:10:44 PM	Wan Nor AzierahBinti Wan Mohd Nasir	27-Oct-2025 10:39:36 AM
	SWMY-QF-00004	NUR DAFINA FARISKA	18-Jun-2025 11:14:08 AM	NUR DAFINA FARISKA	18-Jun-2025 02:00:49 PM
	SWMY-QF-00004	NUR DAFINA FARISKA	18-Jun-2025 10:19:17 AM	NUR DAFINA FARISKA	18-Jun-2025 11:14:08 AM
	SWMY-QF-00004	NUR DAFINA FARISKA	18-Jun-2025 09:58:12 AM	NUR DAFINA FARISKA	18-Jun-2025 10:04:16 AM
	Profile No: SWMY-QF-00005				
	SWMY-QF-00005	NUR DAFINA FARISKA	18-Jun-2025 02:11:05 PM	NUR DAFINA FARISKA	11-Jul-2025 03:44:05 PM
	SWMY-QF-00005	NUR DAFINA FARISKA	18-Jun-2025 11:13:39 AM	NUR DAFINA FARISKA	18-Jun-2025 02:11:05 PM
	SWMY-QF-00005	NUR DAFINA FARISKA	18-Jun-2025 10:42:22 AM	NUR DAFINA FARISKA	18-Jun-2025 11:13:39 AM
	SWMY-QF-00005	NUR DAFINA FARISKA	18-Jun-2025 10:21:22 AM	NUR DAFINA FARISKA	18-Jun-2025 10:30:45 AM
	Profile No: SWMY-QF-00006				
	SWMY-QF-00006	NUR DAFINA FARISKA	18-Jun-2025 02:11:27 PM	NUR DAFINA FARISKA	11-Jul-2025 03:43:47 PM
	Count: 617				

## 6. Summary

The **Dynamic Form Module** is a metadata-driven engine that allows forms to be created, modified, and rendered dynamically **without writing any code**. It provides a scalable, configurable, and reusable solution for building complex data-collection workflows across the application.

### ◆ Key Highlights

- Fully **configurable forms** using database metadata
- Supports multiple **field types**, validations, and conditional rules
- **Dynamic UI rendering** using metadata (sections, attributes, templates)
- Designed with robust **security, audit, and release workflow**
- Easy to extend, maintain, and adapt to new business scenarios

### ◆ Core Features

- Create Forms, Sections, Fields (Attributes) without deployment
- Support for:
  - Text, Number, Date, Dropdown, File Upload, Grid
  - Conditional visibility rules
  - Auto-numbering & formula fields
- Section-level permissions: View, Edit, Hide
- Versioning and Release/Approval workflow
- Complete Audit Trail (Before/After values)
- Data stored in JSON or relational format
- Integrations via REST APIs for lookups & external systems