Functional Design Document

3PL Portal [Internal IP]

# Introduction

The purpose of this Functional Design Document (FDD) is to provide an overview of the following components relevant to the 3PL Customer Portal:

* The functions of the 3PL Customer Portal
* The data tables exposed in the 3PL Customer Portal
* The security strategy, security roles and web roles to support the 3PL Customer Portal
* The user journeys relevant to the 3PL Customer Portal
* The automation that is needed to handle certain functions in the 3PL Customer Portal

# Function

## Project details

The high-level objective of the 3PL Customer Portal is to expose Dynamics 365 Finance & Operations data inside of a customer-facing website that only authenticated users can access.

* Goal
* High-level benefits

## Target audience

This information is required to inform team members about the audience for this document. We can provide a list of all the roles that come under our target audience category, similar to the following:

* Project managers
* Business analysts
* Consultants
* Developers
* Testers
* Architects

## Terminology

This section includes any special words or abbreviations used in the document.

|  |  |
| --- | --- |
| Terminology | Definition |
| Dual-Write | Dual-write is the out-of-box infrastructure usd to provide near-real-time interaction between Power Platform (Dataverse) and Finance & Operations applications. Dual-Write provides a tightly coupled, bidirectional integration between Finance and Operations Apps and Dataverse. |
| Web Role | A web role allows an external user to the organization to perform any special actions or access any protected content on the portal.  Web roles will also be used to differentiate the invoice visibility that will be provided in the portal. |
| Table Permissions | Table permissions are added to web roles to defines roles in the organization that correspond logically to the privileges and concepts of record ownership and access. A given contact can belong to any number of roles. |
| Basic Forms | Basic Forms are used to generate a layout, capture the data, provide read-only access or full editing capabilities for a Dataverse table row |
| Basic Form Metadata | The basic form Metadata contains extra behavior modification logic to augment or override the functionality of form fields that is otherwise not possible with native basic form editing capabilities |
| Lists | Lists define how a list of Dataverse rows is displayed on the site pages. They are defined by one or more table views and include functionality like pagination, filtering, and sorting |
| Webpage Access Control Rules | Webpage Access Control Rules are used to grant or deny access to the individual pages on the website. |

## Assumptions

This section includes any assumptions about the implementation of the 3PL Customer Portal.

|  |  |
| --- | --- |
| Assumptions | Description |
| 1 | In order for the integration to work, you must have two Azure Active Directory (Azure AD) applications set up for the finance and operations environment and two application users set up in the Dataverse environment. These application users should contain the appropriate application IDs. For the connection to work properly, you must the applications the relevant table permissions by using a security role. |
| 2 | End users who are configuring dual-write mappings should have System Administrator security roles assigned in both Dataverse and Finance & Operations apps environments. |
| 3 | Dual-write mappings can be accessed by multiple users if all the users and environments belong to a single tenant and the user has the required security and licenses assignment. |
| 4 | You can make Dataverse custom tables company-specific by adding a many-to-one (N:1) relationship between your custom tables and the out-of-box company table. You should also include the company foreign keys as part of the table key. To enable table maps for dual-write, you must define an alternal key in Dataverse and the value of the alternative key in Dataverse must match the key that is defined in the F&O app. |
| 5 |  |

## Risks

This section outlines any risks involved with the 3PL Customer Portal implementation and solution.

|  |  |
| --- | --- |
| Risk | Description |
| 1 | You cannot create legal table-specific maps valid only for some of the legal entities in F&O. Legal table mapping can be done when the Dataverse environment is linked. You cannot map table maps to a specific legal entity. |
| 2 | Finance and Operations apps do not permit the merging of records. Because of this, the merge functionality in Power Apps/Customer Engagement apps will not execute when the dual-write mapping is present on a table. |
| 3 |  |
| 4 |  |
| 5 |  |

# User Journeys

## Onboarding Process

The purpose of the ‘Onboarding Process’ User Journey is to allow users currently authenticated into the portal the ability to add additional users using self-service capabilities. To add an additional portal user (Contact), the original user would ensure they are logged in, navigate to Onboarding and select **Create** in the top right corner of the Users view. The user would fill out the form by providing First Name, Last Name, E-mail, and Mobile Phone). The Company Name is selected by default based on the authentication of the original user. Once the form is complete, the user will select **Submit**.

Once the user selects **Submit**, an automated background workflow called **Send Portal Invitation to Onboarded Users** runs. This workflow grants the new user a security role and sends a portal invitation to the designated email.

## View Invoices

The portal enables users to view Invoices by status by browsing the ***Invoices*** page of the 3PL Customer Portal. The Invoices are displayed in a list View highlighted in the ***Invoices*** table section of this FDD.

There are three distinct Invoice Visibility Use Cases that will be supported by the portal:

*Invoice To Company*

Contacts whose Primary Contact is the Invoice To Account must be able to see invoices associated to all companies associated to the customer.

*Sold To Company*

Contact whose Primary Contact is the Sold To Account must be able to see only the invoices associated with the Sold To Company

*Standard Visibility*

All of the standard visibility rules for the portals design should not be impacted while providing the described cross-company visibility.

## Company Overview

## 

Requesting Support

The portal lets users submit support cases using the Customer self-service functionality. To submit a support request (Case), the user would ensure they are logged in, navigate to Support, and select Create in the top right corner of the List of support cases. The user would fill out the form by providing Case Title, Case Type, Priority & Case Description). The Company Name & Contact is selected by default based on the authentication of the original user. Users can optionally also attach a supporting file/document to the request (Case). Once the form is complete, the user will select **Submit**.

## Payment of Invoices

The portal enables portal users (Customers) to make payments on open invoices. The high-level components involved and interaction between them are as explained below.

Diagram

Description automatically generated

1. The application consisting of server and client-side renders the UI necessary
2. On the client side (web page), we will use the payment gateway’s embedded form to capture the credit card details which will send it to the payment gateway directly bypassing your application, but advising you some kind of a reference (or token or [nonce](https://en.wikipedia.org/wiki/Cryptographic_nonce)) associated with that transaction.
3. The application then retrieves the payment status from the payment gateway using that reference.

The application will not “see” the client customers details, but only operates using abstract references.

The whole solution will consist of the following elements:

* eWAY payment gateway — you’ll need to create a (free sandbox) account there.
* CDS entity “Payment” which will represent a record of a successful payment
* The payment page which embeds the eWAYs “Pay now button”.
* Processing page that will request payment status information and submits the record in CDS
* Receipt page that will display the transaction reference — this can be replaced with your business process (e.g. allocate resources, open access, dispatch goods etc).
* Portal Entity Form for submitting a new Payments record.
* An Azure Function which will help us to avoid writing a plugin for Portal.

### EWay Partner SandBox Credentials:

|  |  |
| --- | --- |
| Username | Password |
| [cama@avantiico.com](mailto:cama@avantiico.com) | yaiL1yer!!!! |

# Dual-Write Setup and Configuration

The 3PL Portal Solution is dependent on the configuration of Dual-Write as an integration method. Dual-Write provides a tightly coupled, bidirectional integration between finance and operations apps and Dataverse. Any data change in finance and operations apps causes writes to Dataverse, and any data change in Dataverse causes writes to finance and operations apps.

Graphical user interface, application

Description automatically generated

## Dual-Write Table Mapping

The following tables are mapped between Dataverse and Dynamics 365 Finance:

|  |  |  |
| --- | --- | --- |
| Dataverse Table Name | Finance Table Name | Notes |
| Global Product (msdyn\_globalproduct) | All Products | Contains all the products available in finance and operations apps, both the released products and the non-released products. |
| Contact (contacts)\* | CDS Contacts V2 | This template synchronizes all primary, secondary and tertiary contact information for both customers and vendors. |
| Party (msdyn\_party) | CDS Parties |  |
| Product (product) | CDS released distinct products |  |
| Order (salesorder) | CDS sales order headers |  |
| Order Product (salesorderdetail) | CDS sales order lines |  |
| Quote (quote) | CDS sales quotation header |  |
| Quote Product (quotedetail) | CDS sales quotation lines |  |
| Product Color (msdyn\_productcolor) | Colors |  |
| Complimentary Closing (msdyn\_complimentaryclosing) | Complimentary closings |  |
| Product Configuration (msdyn\_productconfiguration) | Configurations |  |
| Sales Contact Person Title (msdyn\_salescontactpersontitle) | Contact person titles |  |
| Contact for Customer or Vendor (msdyn\_contactforparties) | Contacts V2 |  |
| Currency (transactioncurrencies) | Currencies |  |
| Customer Group (msdyn\_customergroups) | Customer groups | This template synchronizes customer group information. |
| Customer Payment Method (msdyn\_customerpaymentmethods) | Customer payment method | Maurice or Michael to demo this table to us. This template synchronizes customer payment information. |
| Account (accounts) | Customers V3 | This template synchronizes customer master information for commercial and organizational customers. |
| Contact (contacts)\* | Customers V3 | This template synchronizes customer master data for consumers and end users. |
| Decision Making Role (msdyn\_decisionmakingroles) | Decision making roles |  |
| Employment Job Function (msdyn\_employmentjobfunctions) | Employment job functions |  |
| Financial Dimension Format (msdyn\_financialdimensionformats) | Financial dimension format |  |
| Financial Dimension (msdyn\_dimensionattribute) | Financial dimensions |  |
| Warehouse Aisle (msdyn\_warehouseaisles) | Inventory aisle |  |
| Company (cdm\_companies) | Legal entities |  |
| Internal organization (mysdyn\_internalorganizations) | Legal entities |  |
| Loyalty Level (mysdyn\_loyaltylevels) | Loyalty levels |  |
| Ship Via (msdyn\_shipvias) | Modes of delivery |  |
| Internal organization (msdyn\_internalorganization) | Operating unit |  |
| Party Electronic Address (msdyn\_partyelectronicaddress) | Party contacts V3 |  |
| Payment Day(msdyn\_paymentdays) | Payment days CDS | This template synchronizes payment day lines reference data for both customers and vendors. |
| Payment Schedule (msdyn\_paymentschedules) | Payment schedule | This template synchronizes payment, schedule reference data, for both customers and vendors. |
| Personal Character Type (msdyn\_personalcharactertype) | Personal character types |  |
| Price Customer Group (msdyn\_pricecustomergroups) | Price customer groups |  |
| Product Category (msdyn\_productcategories) | Product categories | Each of the product categories, and information about its structure and characterisitics are contained in the product category table. |
| Product Category Assignment (msdyn\_productcategoryassignments) | Product category assignments | To assign a product to a category, the product category assingments table can be used. |
| Product Category Hierarchy (msdyn\_productcategoryhierarchies) | Product category hierarchies | You use product hierarchies to categorize or group products. The category hierarchies are available in Dataverse using the Product Category Hierarchy Table. |
| Product Category Hierarchy Role (msdyn\_productcategoryhierarchyroles) | Product category hierarchy roles | Product hierarchies can be used for different roles in D365 F&O. They specific which category is used in each role the product category role table is used. |
| Product Dimension Group (msdyn\_productdimensiongroups) | Product dimension groups | Defines which product dimensions define the product. |
| Product Specific Unit Conversion (msdyn\_productspecificunitofmeasureconversion) | Product specific unit conversions |  |
| Invoice (invoice) | Sales invoice headers V2 |  |
| Invoice Product (invoicedetails) | Sales invoice lines V2 |  |
| Sales Order Origin (msdyn\_salesorderorigins) | Sales order origin codes |  |
| Tax Group (msdyn\_taxgroups) | Sales tax groups |  |
| Salutation (msdyn\_salutation) | Salutations |  |
| Operational Site (msdyn\_operationalsites) | Sites |  |
| Product Size (msdyn\_productsizes) | Sizes |  |
| Product Storage Dimension Group (msdyn\_productstoragedimensiongroups) | Storage dimension groups |  |
| Product Style (msdyn\_productstyles) | Styles |  |
| Terms of Delivery (msdyn\_termsofdeliveries) | Terms of delivery |  |
| Payment Term (msdyn\_paymentterms) | Terms of payment |  |
| Product Tracking Dimension Group (msdyn)producttrackingdimensiongroups) | Track dimension groups | Represents the method used to trak the product in inventory. |
| Unit Conversion (msdyn\_unitofmeasureconversions) | Unit conversions |  |
| Unit (uom) | Units |  |
| Vendor Group (msdyn\_vendorgroups) | Vendor groups |  |
| Vendor Payment Method (msdyn\_vendorpaymentmethods) | Vendor payment method | This template synchronizes vendor payment method information. |
| Vendor (vendor) | Vendors V2 | Leverages new out of box vendor concept. |
| Warehouse Zone Group (msdyn\_warehousezonegroup) | Warehouse zone groups |  |
| Warehouse Zone (msdyn\_warehousezone) | Warehouse zones |  |
| Warehouse (msdyn\_warehouse) | Warehouses |  |
| Worker (cdm\_worker) | Worker |  |

\*Invoice Visibility Use Cases will require adjustments to the Finance and Dataverse Tables to allow the Parent Party Number of the Invoice To Customer and Sold To Customer to be included in the Invoice Data Map.

# Data Management Settings

## Business Unit Hierarchy

# Portal Management Settings

## Field-level Security Profiles

Field-level security is used to implement security for sensitive fields. Customers may want to control the create and update operations on sensitive fields. For example, your customer may want to only allow service managers to create and update the Service Cost field. Field-level security is maintained using field-level security profiles. In this section, all the field-level security profiles should be mentioned, as this is required for our Dynamics 365 CE implementation.

|  |  |
| --- | --- |
| User | Description |
| Service Manager | Only the service manager can apply a discount to the total service amount. |

## Dataverse Security Roles

The security of Dataverse tables and their data is implemented using security roles. These are a group of permissions that define which operations a Dynamics 365 CE or Power Platform user can perform on an entity. Dynamics 365 and Power Platform contain many out-of-the-box security roles that can be modified based on requirements.

In this section, we provide details about the security roles that will be used for the solution.

|  |  |  |
| --- | --- | --- |
| Business unit | Security roles | Changed name |
| HIMBAP Auto | System Administrator (OOB) | HIMBAP Admin |
| HIMBAP Auto | System Customizer | HIMBAP Customizer |
| Support | Customer Service Representative | HIMBAP Support |
| Servicing | CSR Manager (Copy of CS Manager) | HIMBAP Service Manager |
| Servicing | Salesperson | HIMBAP Technician |

## Web Roles

Web roles are created to allow users to perform any special actions or access any protected content and data on the site. Web roles link to contacts, table permissions, and page permissions. Because contacts can be assigned multiple web roles, they can be provided cumulative access to site resources.

All authenticated users (contacts) are automatically assigned to the Authenticated Users web role.

A site can be visited by anonymous users (unauthenticated) and given access to assets through the Anonymous Users web role.

The following web roles have been configured for the 3PL portal:

* Administrators
* Authenticated Users
* Anonymous Users

## Table Permissions

Access to Dataverse records is automatically restricted in Power Pages when using forms, lists, and other components accessing Dataverse tables. To allow access to Dataverse records in Power Pages sites table permissions are configured and associated with web roles. For a table permission to take effect, it must be linked to one or more web roles. Users who belong to web roles are granted the privileges you select for the associated table permission.

Power Apps Portals offer four different Access types. Depending on the access type you choose, the selected table permission and privileges apply to the users from the selected roles for the following records.

Global access - Applies the selected table permission and privileges to the users from the selected roles for all records.

Contact access - Applies the selected table permission and privileges to the users from the selected role associated to the signed-in user.

Account access - Applies the selected table permission and privileges to the users from the selected role associated to the signed-in user's account.

Self-access - Applies the selected table permission and privileges to the users from the selected role for only their own Contact record.

Following table permissions are configured for the 3PL Customer Portal:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Name | Table | Access Type | Relationship | Permission | Type | Web Roles |
| Accounts | Account (account) | Contact Access | contact\_customer\_accounts | Read, Append, Append To | Parent | Administrators, Authenticated Users |
| Cases | Case (incident) | Account Access | incident\_customer\_accounts | Read, Create, Append, Append To | Parent | Administrators, Authenticated Users |
| Activities | Activity (activitypointer) |  | Incident\_ActivityPointers | Read, Create, Append, Append To | Child | Administrators, Authenticated Users |
| Appointment | Appointment (appointment) |  | Incident\_Appointments | Read, Append, Append To | Child | Administrators, Authenticated Users |
| Notes | Note (annotation) |  | Incident\_Annotation | Read, Create, Append, Append To | Child | Administrators, Authenticated Users |
| Phone Call | Phone Call (phonecall) |  | Incident\_Phonecalls | Read, Append, Append To | Child | Administrators, Authenticated Users |
| Portal Comment | Portal Comment (adx\_portalcomment) |  | incident\_adx\_portalcomments | Read, Write, Create, Delete, Append, Append To | Child | Administrators, Authenticated Users |
| Task | Task (task) |  | Incident\_Tasks | Read, Create, Append, Append To | Child | Administrators, Authenticated Users |
| Contacts | Contact (contact) | Account Access | contact\_customer\_accounts | Read, Write, Create, Delete, Append, Append To | Parent | Administrators, Authenticated Users |
| Invoices | Invoice (invoice) | Account Access | Invoice\_customer\_accounts | Read | Parent | Administrators, Authenticated Users |
| Invoice Lines | Invoice Product (invoicedetail) | Global Access |  | Read | Parent | Administrators, Authenticated Users |

## Webpage Access Control Rules

Makers control user access to site webpages through webpage access control rules. These rules allow you to control the publishing actions that a web role can perform across the pages of your website. Also, you can control which pages are visible to which web roles. Settings can be adjusted to make content available anonymously for public access, or to restrict access to users who have specific roles.

There are two types of access control rules: Grant Change and Restrict Read.

Grant Change

Use a Grant Change rule to allow a user who has the web role associated with the rule to publish content changes for this page, and all child pages of this page. Grant Change rules take precedence over Restrict Read rules.

Restrict Read

Use a Restrict Read rule to limit the viewing of the contents of a page (and its child pages) to specific users. In comparison, Grant Change is a permissive rule (it grants users the ability to do something), whereas Restrict Read is a restrictive rule in that it restricts an action to a limited set of users

Webpage access control rules are then associated with web roles to only allow change/read permissions to the users assigned to specific web roles

On the 3Pl Customer Portal only the Home page is openly accessible to anonymous users. All other pages are restricted to Administrators & Authenticated Users only. The table below summarizes all the access control rules configured for the 3PL portal

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Right | Web Page | Access Type | Web Roles |
| Grant Change to Administrators | Grant Change | Home | All Content | Administrators |
| Grant Change to Content | Grant Change |  | All Content | Administrators |
| Restrict Read on Account Summary | Restrict read | View Company | All Content | Administrators, Authenticated Users |
| Restrict Read on Billing | Restrict read | Billing | All Content | Administrators, Authenticated Users |
| Restrict Read on Customer Service | Restrict read | Support | All Content | Administrators, Authenticated Users |
| Restrict Read on Invoices | Restrict read | Invoices | All Content | Administrators, Authenticated Users |
| Restrict Read on Onboarding | Restrict read | Onboarding | All Content | Administrators, Authenticated Users |
| Restrict Read on Support | Restrict read | Customer Service | All Content | Administrators, Authenticated Users |
| Restrict Read on User setup | Restrict read | User Setup | All Content | Administrators, Authenticated Users |

## Lists

A list is a data-driven configuration that you use to add a webpage that will render a list of records without the need for a developer to surface the grid in the portal. By using lists, you can expose records for display on portals.

The grid supports sorting and will be paginated if the number of records is larger than the page size specified. The list also supports multiple views. If more than one view has been specified, a drop-down list will be rendered to allow the user to switch between the various views.

If Web Page for Details View has been specified, each record will contain a link to the page, and the Id of the record will be appended to the query string along with the Id query string parameter name. The behavior of the target form (read-only or edit) will be determined by the configuration of the form mode and the table permissions assigned to the web roles associated with the user.

Lists configured for the 3PL Customer Portal are as follows:

|  |  |  |
| --- | --- | --- |
| List | Table | Views |
| Account Invoices | Invoice | All Invoices |
| Portal Contacts | Contacts | Portal Contacts View |
| Portal Invoices | Invoice | All Invoices, Open Balances |
| Service Cases | Case | All Cases |

## Basic Forms

Basic forms are created in Microsoft Dataverse and then placed into webpages in the portal or used in conjunction with subgrids and lists to build out complete web applications. The basic form contains relationships to webpages and additional properties to control the initialization of the form within the portal. The relationship to webpages allows dynamic retrieval of the form definition for a given page node within the website.

When creating a new basic form, three modes are available: Insert, Edit, or Read Only. The mode selected will determine if you are creating a new record from the portal, editing an existing record, or just displaying information about a record on the portal.

Following basic forms are configured for the 3PL Portal:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Table | Form | Tab | Mode |
| View Account | Accounts | Information | General | Read Only |
| View Invoices | Invoices | Information |  | Read Only |
| Onboard Users | Contacts | Invite Web Form | General | Insert |
| Create Case | Case | Case Web Form | Web\_form | Insert |
| View Case | Case | Case Web Form | Web\_form | Read Only |

## Basic Form Metadata

The basic form Metadata contains extra behavior modification logic to augment or override the functionality of form fields that is otherwise not possible with native basic form editing capabilities.

The table below shows the basic form metadata configured for different basic forms on the 3PL portal

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Form Name | Type | Attribute | Prepopulate Type | Prepopulate Attribute |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

## ER Diagram

This section provides information about the entities and their relationships to each other. The number of entities may change as they are created, so a consultant can introduce or reuse entities after discussing this with their team. An FDD must include an updated ER diagram; so, if there are any changes in the structure, the business analyst or functional consultant needs to update the FDD. The following diagram shows an example of an ER diagram:

This ER diagram represents out-of-the-box entities and custom entities that we are going to mainly use for our 3PL Portal Solution. As you can see, we are going to reuse some of the existing entities, as well as creating some custom entities.

# Entity Design

This section provides details about the entities and their relationships. Different types of information about the entity—such as forms, views, attributes, and relationship details—are provided here.

The 3PL Portal will leverage the following tables:

## Contact

In Dataverse, a ***Contact***represents a person, usually an individual, with whom a business unit has a relationship, such as a customer, a supplier, or a prospect. The contact table stores all information about a person such as an email address, street address, telephone numbers, and other related information. The contact is assigned to a ***Primary Account***, that is associated with a ***Primary Party***, which will be used to provide correct Invoice visibility for two specific Visibility Use Cases. In the 3PL Customer Portal, the ***Contact*** is invited with a ***Portal Invitation*** and tied to a ***Web Role*** to grant authentication to the portal and give the authorization to view data belonging to pre-determined tables.

## Users

A ***User*** is any person who works for a business unit who uses Dataverse. Each user has a user account. All users must be associated with only one business unit. This association controls which customer data the user will have access to. Included in the user’s account is information such as the user’s telephone numbers, email address, and a link to the internal user’s manager. Each user has privileges and rights to manage their own personal settings. Each user corresponds to a user in the Azure Active Directory for that organization. When a user is created, they must be assigned at least one security role.

## Party

A ***Party*** allows companies to be grouped to allow cross-company aggregation of information. Each Company will have a Parent Party assigned if they are to participate in a grouping of companies.

## Payment

The payment table is a custom table that represents a record of a successful payment once an Invoice has been paid. The Customer column will serve to associate payments to a customer record as part of the checkout logic.

### Payment Columns

* Name: Text
* Amount: floating-point number
* Customer: Look Up
* Verification: Text-512 length

### Payment Form

## Products

Product information contains all of the information related to the product and its definition, such as the product dimensions or the tracking and storage dimensions. A collection of table maps is created in Dual-write in order to sync products and related information.

The unified product experience brings the integrated product data model into Dataverse, so that all application users, including Power Platform users, can take advantage of the rich product data coming from finance and operations apps. The product data model differs between the two applications as follow.

The following model illustrates the product data model from Dynamics 365 Sales:

Diagram

Description automatically generated

The following model illustrates the product data model from finance and operations apps:

Diagram

Description automatically generated

These two product data models have been integrated in Dataverse as shown below:

Diagram

Description automatically generated

The Dual-write table maps for products have been designed to flow data one-way only, in near-real time from finance operations to Dataverse.

In this model, the product is represented by the combination of two tables in Dataverse: **Product** and **msdyn\_sharedproductdetails**. Whereas the first table contains the definition of a product (the unique identifier for the product, the product name, and the description), the second table contains the columns stored at the product level. The combination of these two tables is used to define the product according to the concept of the stockkeeping unit (SKU). Each released product will have its information in the mentioned tables (Product and Shared Product Details). To keep track of all products (released and not released), the **Global products** table is used.

Because the product is represented as a SKU, the concepts of distinct products, product masters, and product variants can be captured in Dataverse in the following way:

**Products with subtype product** are products that are defined by themselves. No dimensions have to be defined. An example is a specific book. For these products, one row is created in the **Product** table, and one row is created in the **msdyn\_sharedproductdetails** table. No product family row is created.

**Product masters** are used as generic products that hold the definition and rules that determine the behavior in business processes. Based on these definitions, distinct products that are known as product variants can be generated. For example, T-shirt is the product master, and it can have Color and Size as dimensions. Variants can be released that have different combinations of these dimensions, such a small blue T-shirt or a medium green T-shirt. In the integration, one row per variant is created in the product table. This row contains the variant-specific information, such as the different dimensions. The generic information for the product is stored in the **msdyn\_sharedproductdetails** table. (This generic information is held in the product master.) The product master information is synced to Dataverse as soon as the released product master is created (but before variants are released).

**Distinct products** refer to all the products subtype product and all the product variants.

Diagram

Description automatically generatedWith the dual-write functionality enabled, the products from finance and operations will be synchronized in other Dynamics 365 products in **Draft** state. They are added to the first price list with the same currency used in the customer engagement app and using alphabetical sort on the price list name. In other words, they are added to the first price list in a Dynamics 365 app that matches the currency of your legal table where the product is released in a finance and operations app. If there is no price list for the given currency, a price list will automatically be created and the product will be assigned to it.

The current implementation of the dual-write plugins that associate the default price list to the unit look up the currency associated with the finance and operations app and find the first price list in the customer engagement app using alphabetical sort on the price list name. To set a default price list for a specific currency when you have multiple price lists for that currency, you must update the price list name to a name that is earlier in alphabetical order than any other price lists for that same currency. If it does not have any price list for the given currency, a new one is created.

By default products from finance and operations apps are synchronized to other Dynamics 365 apps in **Draft** state. To synchronize the product with **Active** state so that you can directly use it in sales order quotations, for example, the following setting needs to be chosen: **System> Adminstration > System administration > System settings > Sales** tab and select **Create products in active state = yes**.

When products are synchronized, you must enter a value for the **Sales unit** field in the finance and operations app, because it is a mandatory field in Sales.

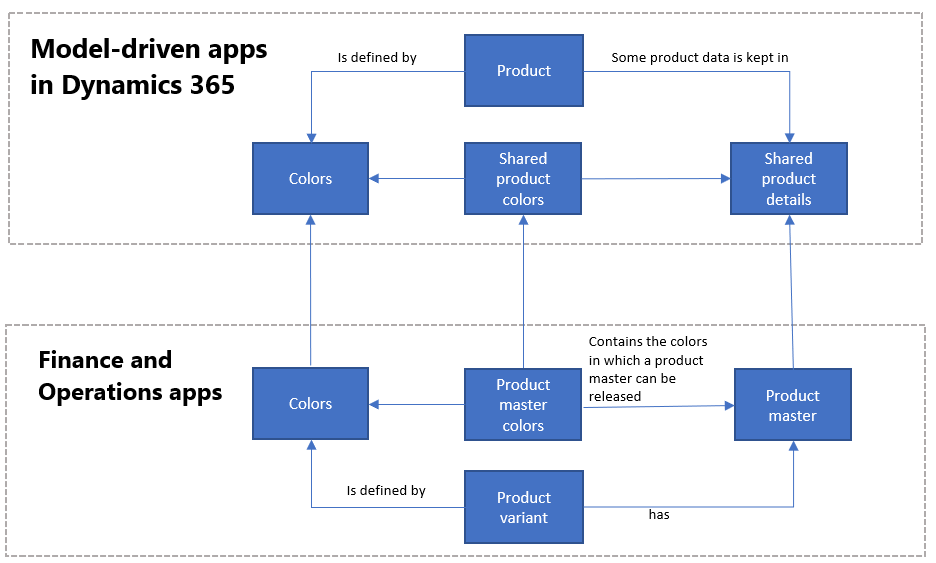
The creation of product families from Dynamics 365 Sales is not supported with the dual-write synchronization of products.

The synchronization of products happens from the finance and operations app to Dataverse. This means that the values of the product table columns can be changed in Dataverse, but when the synchronization is triggered (when a product column is modified in a finance and operations app), this will overwrite the values in Dataverse.

|  |  |
| --- | --- |
| Finance & Operations Apps | Customer Engagement Apps |
| CDS released distinct products | Product |
| Released products V2 | Msdyn\_sharedproductdetails |
| All products | Msdyn\_globalproducts |

## Product Dimensions

Product dimensions are characteristics that identify a product variant. The four product dimensions (Color, Size, Style, and Configuration) are also mapped to Dataverse to define the product variants. The following illustration shows the data model for the product dimension Color. The same model is applied to Sizes, Styles and Configurations.



|  |  |
| --- | --- |
| Finance & Operations Apps | Customer Engagement Apps |
| Colors | Msdyn\_productcolors |
| Sizes | Msdyn\_productsizes |
| Styles | Msdyn\_productstyles |
| Configurations | Msdyn\_productconfigurations |

When a product has different product dimensions (for example, a product master has Size and Color as product dimensions), each distinct product (that is, each product variant) is defined as a combination of those product dimensions. For example, product number B0001 is an extra-small black T-shirt, and product number B0002 is a small black T-shirt. In this case, the existing combinations of product dimensions are defined. For example, the T-shirt from the preceding example can be extra-small and black, small and black, medium and black, or large and black, but it can't be extra-large and black. In other words, the product dimensions that a product master can take are specified, and variants can be released based on these values.

To uniquely identify products between Dynamics 365 Finance and products in Dataverse the integration keys are used. For products, the **(productnumber)** is the unique key that identifies a product in Dataverse. It's composed by the concatenation of: **(company, msdyn\_productnumber)**. The **company** indicates the legal entity in finance and operations and **msdyn\_productnumber** indicates the product number for the specific product in finance and operations.

For users of other Dynamics 365 apps, the product is identified in the UI with the **msdyn\_productnumber** (note that the label of the column is **Product number**). In the product form both the company and the msydn\_productnumber are shown. However, the (productnumber) column, the unique key for a product, is not shown.

If you build apps on Dataverse, you should pay attention to using the **productnumber** (the unique product ID) as the integration key. Do not use **msdyn\_productnumber**, because it's not unique.

* Unit of measure
* Conversions
* Additional tracked dimensions

## Customer

***Customer*** represents the integration of customer data that involves harmonizing the customer concept between two applications. The following illustration shows the customer data flow:

Diagram

Description automatically generated

Customers can be broadly classified into type types: commercial/organizational customers and consumers/end users. These two types of customers are stored and handled differently in finance and operations and Dataverse.

In finance and operations, both commercial/organizational customers and consumers/end users are mastered in a single table that is named **CustTable** (CustCustomerV3Entity), and they are classified based on the **Type** attribute. (If **Type** is set to **Organization**, the customer is a commercial/organizational customer, and if **Type** is set to **Person**, the customer is a consumer/end user.) The primary contact person information is handled through the SMMContactPersonEntity table. Customers may be grouped into Parties by setting the ???Parent Party?? Field. The current Visibility Use Cases only require one level of Parent Parties be associated with the customer.

In Dataverse, commercial/organizational customers are mastered in the Account table and are identified as customers when the **RelationshipType** attribute is set to **Customer**. Both consumers/end users and the contact person are represented by the Contact table. To provide a clear separation between a consumer/end user and a contact person, the **Contact** table has a Boolean flag that is named **Sellable**.

When **Sellable** is **True**, the contact is a consumer/end user, and quotations and orders can be created for that contact. When **Sellable** is **False**, the contact is just a primary contact person of a customer.

When a non-sellable contact participates in a quotation or order process, **Sellable** is set to **True** to flag the contact as a sellable contact. A contact that has become a sellable contact remains a sellable contact.

## Vendor

The term vendor refers to a supplier organization, or a sole proprietor who supplies goods or services to a business. Although vendor is an established concept in Microsoft Dynamics 365 Supply Chain Management, no vendor concept exists in customer engagement apps. However, you can overload the **Account/Contact** table to store vendor information. The integrated vendor master introduces an explicit vendor concept in customer engagement apps. You can either use the new vendor design or store vendor data in the **Account/Contact** table. Dual-write supports both approaches.

In both approaches, the vendor data is integrated between Dynamics 365 Supply Chain Management, Dynamics 365 Sales, Dynamics 365 Field Service, and Power Apps portals. In Supply Chain Management, the data is available for workflows such as purchase requisitions and purchase orders.

Diagram

Description automatically generated

### Account

#### Account Form

We will customize the out-of-box Account form to look like the following:

#### Account Views

We will be using the following views for our Account table:

|  |  |  |  |
| --- | --- | --- | --- |
| Default view | Change to | Sorting | Filter condition |
| Active Accounts | Active Customers | Name | Status Equals Active |
| Inactive Accounts | Inactive Customers | Name | Status Equals Inactive |
| Account Lookup View | Customer Lookup View | Name | Status Equals Active |
| Account Associated View | Customer Associated View | Name | Status Equals Active |
| Account Advanced Find View | Customer Advanced Find View | Name | Status Equals Active |

All of the views will have the following fields:

* Customer Name
* Customer Number
* Main Phone
* Address 1: City
* Primary Contact
* Email

### Contact

#### Contact Form

We will customize the out-of-box Contact form to look like the following:

**Relationships**

You can refer to the following details to set up a custom N: 1 relationship in the customer entity:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Primary entity | Secondary entity | Type of behavior | Relationship name | Lookup field name |
| User | Customer | Referential | him\_systemuser\_account\_Manager | Manager |

**Duplicate detection rules**

The following duplicate detection rule will be enabled on the customer entity:

Accounts with the same account name

## Invoices

## Table Permissions

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Name | Table | Access Type | Relationship | Permission | Type | Web Roles |
| Invoices by Account | Invoice (invoice) | Account access | Invoice\_customer\_accounts | Read-Append-Append to | Parent | Administrators, Authenticated Users, Customer Representative |

## Contact

We are going to use the existing contact entity to store contacts associated with the customer.

**Contact form**

We will be customizing an out-of-the-box Contact form, to look like the following:



**Attributes**

Please refer to the Appendix1 folder on the GitHub repository for this book, at <https://github.com/PacktPublishing/Implementing-Microsoft-Dynamics-365-Customer-Engagement/blob/master/Appendix1.xlsx>.

**Views**

We will be using the following views for our customer entity:

|  |  |  |
| --- | --- | --- |
| Default view | Sorting | Filter condition |
| Active Contacts | Name | Status Equals Active |
| Inactive Contacts | Name | Status Equals Inactive |
| Contact Lookup View | Name | Status Equals Active |
| Contact Associated View | Name | Status Equals Active |
| Contact Advanced Find View | Name | Status Equals Active |

*All of the views will have the following fields:*

Full name

Customer name

Email

Mobile phone

**Auto service**

If we are going to reuse the opportunity entity to store service information, we need to customize the opportunity entity form, as we have in the following example:



**Attributes**

Please refer to the Appendix1 folder on the GitHub repository for this book, at <https://github.com/PacktPublishing/Implementing-Microsoft-Dynamics-365-Customer-Engagement/blob/master/Appendix1.xlsx>.

**Views**

We will be using the following views for our auto service entity:

|  |  |  |  |
| --- | --- | --- | --- |
| Default view | Change to | Sorting | Filter condition |
| Open Opportunities | Open Auto Services | Topic | Status Equals Active |
| Closed Opportunities | Closed Auto Services | Topic | Status Equals Inactive |
| Opportunity Lookup View | Auto Service Lookup View | Topic | Status Equals Active |
| Opportunity Associated View | Auto Service Associated View | Topic | Status Equals Active |
| Opportunity Advanced Find View | Auto Service Advanced Find View | Topic | Status Equals Active |

*All the views will have the following fields:*

Vehicle

Topic

Customer

Contact

Service Date

**Relationship**

Set up the following custom N:1 relationship in the auto service entity:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Primary entity | Secondary entity | Type of behavior | Relationship name | Lookup field name |
| Vehicle | Auto Service | Referential | him\_vehicle\_opportunity\_Vehicle | Vehicle |

## Auto service line

We are going to reuse the opportunity product entity to store auto service line items, and we are going to use this form as it is.

## Case

We are going to use a Case form to store service support information for the auto service. We need to customize the Case entity form, as shown in the following screenshot:



**Attributes**

Please refer to the Appendix1 folder on the GitHub repository for this book, at <https://github.com/PacktPublishing/Implementing-Microsoft-Dynamics-365-Customer-Engagement/blob/master/Appendix1.xlsx>.

**Views**

We will be using the following views for the customer entity:

|  |  |  |
| --- | --- | --- |
| Default view | Sorting | Filter condition |
| Active Cases | Case Title | Status Equals Active |
| Resolved Cases | Case Title | Status Equals Resolved |
| Case Lookup View | Case Title | Status Equals Active |
| Associative Cases | Case Title | Status Equals Active |
| Case Advanced Find View | Case Title | Status Equals Active |

*All the views will have the following fields:*

Case Title

Case Number

Customer

Vehicle

Status Reason

Created On

**Relationship**

Set up the following custom N:1 relationship in our auto case entity:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Primary entity | Secondary entity | Type of behavior | Relationship name | Lookup field name |
| Vehicle | Case | Referential | him\_vehicle\_incident\_Vehicle | Vehicle |

## Vehicle

This is a custom entity used to hold vehicle information such as vehicle registration number, manufacture, build year, and so on. We will be using the following design for this entity form:

**Attributes**

Please refer to the Appendix1 folder on the GitHub repository for this book, at <https://github.com/PacktPublishing/Implementing-Microsoft-Dynamics-365-Customer-Engagement/blob/master/Appendix1.xlsx>.

**Views**

*Change all the views to have the following columns:*

Vehicle Number

Customer

Make

Model

Year

**Relationship**

Set up the following custom N:1 relationship in the vehicle entity:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Primary entity | Secondary entity | Type of behavior | Relationship name | Lookup field name |
| Customer | Vehicle | Referential | him\_account\_him\_vehicle | Customer |
| Model | Vehicle | Referential | him\_model\_him\_vehicle | Model |
| Manufacturer | Vehicle | Referential | him\_manufacturer\_him\_vehicle | Make |
| Year | Vehicle | Referential | him\_year\_him\_vehicle\_Year | Year |

## Manufacturer

This is a custom entity used to hold manufacturer information relating to the vehicle. We will be using the following design for this entity form:



**Attributes**

Please refer to the Appendix1 folder on the GitHub repository for this book, at <https://github.com/PacktPublishing/Implementing-Microsoft-Dynamics-365-Customer-Engagement/blob/master/Appendix1.xlsx>.

## Model

This is a custom entity used to record model details of the vehicle. We will be using the following design for this entity form:



**Attributes**

Please refer to the Appendix1 folder on the GitHub repository for this book, at <https://github.com/PacktPublishing/Implementing-Microsoft-Dynamics-365-Customer-Engagement/blob/master/Appendix1.xlsx>.

## Year

This is a custom entity used for year details. We are going to use the default form, without making any changes for this entity. There are no additional attributes created for this entity apart from the primary field, which is used to store the year name.

We are going to use all the other out-of-the-box entities that are present on the ER diagram as is, without making any changes.

## User reports

This section contains information about the records required. Dynamics 365 CE has many default records that can be used as they are, or they can be customized based on the requirements.

# Overview of Automation

## Workflows

### Send Portal Invitation to Onboarded Users

This workflow is a Power Automate flow that triggers when a new portal user is added by an existing portal user through self-service Onboarding capabilities within the Portal.

### Upload Invoices to SharePoint

This workflow is a Power Automate flow that triggers when a new invoice record is added to Dataverse via Dual-Write when an invoice is posted in D365 F&O. This flow uses the dynamic information from the invoice record & generates a PDF copy of the invoice record using a Word Online template that has XML mapping enabled. The template is stored on OneDrive for Business & is accessed by the flow at runtime to populate the invoice information. The flow then grabs the PDF copy of the invoice & uploads it to the Invoices folder of the SharePoint repository.