

Technical Design Document (TDD)

Power Apps Prototype - Delivery Tracking System

1. Document Information

Item	Description
Project Name	Delivery Tracking System
Application Type	Microsoft Power Apps Canvas App
Version	1.0
Author	Adeline Lee
Date	8 September 2026
Purpose	Prototype application for recording, storing and searching delivery requests.

2. Project Overview

2.1 Purpose

The Delivery Tracking System is a Power Apps Canvas application that allows users to capture customer delivery details, store the records in an Excel table, and locate existing records from a searchable gallery.

- Create a new delivery record.
- Capture the customer name, address and phone number.
- Record item quantity, delivery option and fragile-item status.
- Search and view saved delivery records.

2.2 Objectives

- Provide a simple, visually guided delivery-entry interface.
- Store delivery records in the DeliveryTable Excel data source.
- Clear selected entry fields after a successful save.
- Provide navigation between the welcome, entry and search screens.

3. System Architecture

3.1 Technology Stack

Component	Technology
Front end	Microsoft Power Apps Canvas App
Data storage	Excel Online table
Primary data table	DeliveryTable
Interaction model	Canvas controls, Patch(), Reset(), Navigate() and Gallery controls

3.2 Data Flow

The user enters delivery details on Screen1. Selecting Save As executes Patch() to create a new row in DeliveryTable. The Reset() calls then clear the customer name, address, phone, item count and fragile controls. Screen2 presents saved records through Gallery1 and accepts a search term through Searchinput.

4. Data Model Design

4.1 Delivery Entity

Table: DeliveryTable

Field name	Data type	Source control / note
CustomerName	Text	CustomerNameTextInput.Text
CustomerAddress	Text	AddressTextInput.Text
CustomerPhone	Text	PhoneTextInput.Text; text preserves leading zeroes and symbols
Fragile	Boolean	Fragile.Value
ItemCount	Number / choice value	ItemCount.Selected.Value
DeliveryOption	Text / choice value	DeliveryOption.Selected.Value
__PowerAppsId__	System-generated text	Connector-managed unique row identifier

4.2 Example Record

Customer	Address	Phone	Fragile	Items	Delivery option
Jamie	River Valley	67674545	True	1	Deliver to Door Step

5. User Interface Design

The application contains three screens: MainStart, Screen1 and Screen2.

5.1 MainStart - Welcome Screen

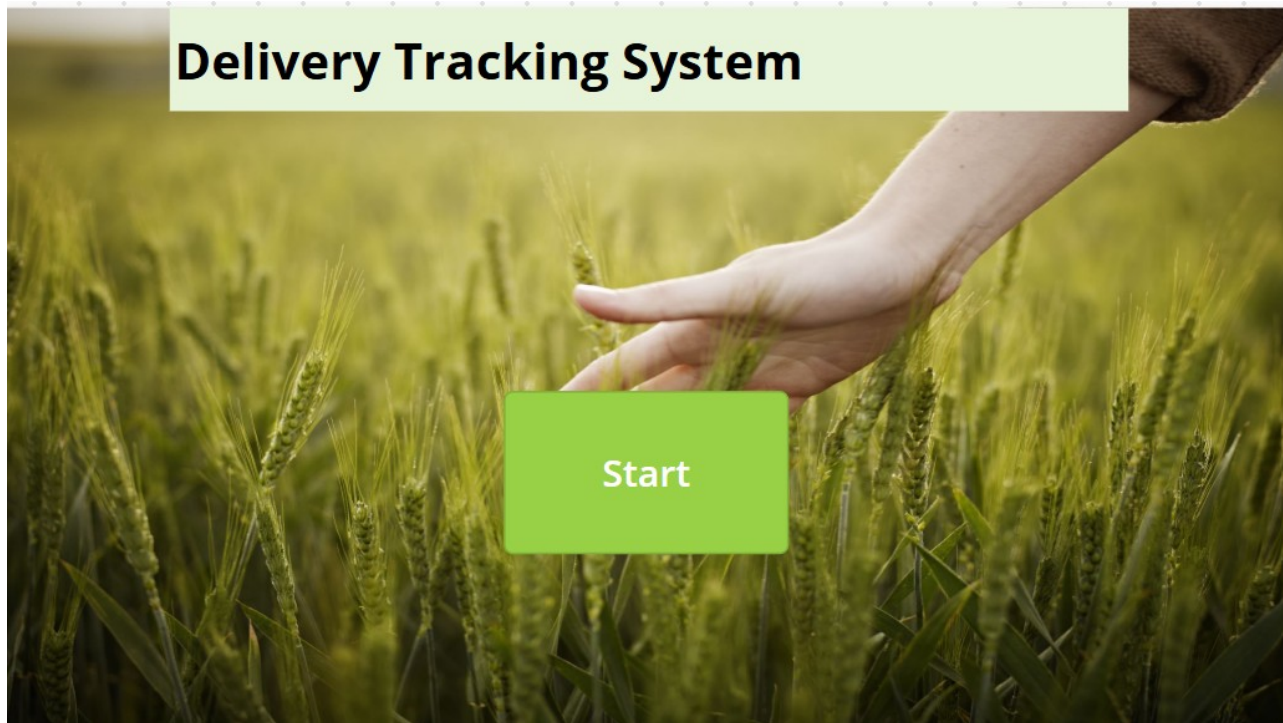


Figure 1. MainStart welcome screen

Purpose: Provides the application welcome page and entry point.

Screen name: MainStart

Control	Description
Label5	Displays the title "Delivery Tracking System".
Background image	Provides the field-themed application background.
Button3 (Start)	Opens the delivery-entry screen.

Expected Start Button Logic

```
Navigate(Screen1);
```

Note: The navigation formula is documented as the intended design based on the screen flow; its property formula was not included in the supplied formula extract.

5.2 Screen1 - Delivery Input Screen

Delivery Tracking System

Customer Name: MyName

Customer Address: My Address

Customer Phone: My Address

Item Count: 1

Delivery Option: ☒ Deliver to Door Step ☐ Self-pickup

☐ Fragile

Save As **Go To Search**

Figure 2. Delivery entry and save screen

Purpose: Captures and saves a new delivery request.

Screen name: Screen1

Control	Purpose
CustomerNameTextInput	Capture customer name.
AddressTextInput	Capture delivery address.
PhoneTextInput	Capture customer phone number.
ItemCount	Select the number of items.
DeliveryOption	Select Deliver to Door Step or Self-pickup.
Fragile	Indicate whether the delivery contains fragile items.
Save As	Create a new record and reset specified controls.
GoToSearch	Open Screen2 to search saved records.

6. Save Logic

6.1 Save As Button - OnSelect Formula

```
Patch(DeliveryTable, Defaults(DeliveryTable),
{
    CustomerName: CustomerNameTextInput.Text,
    CustomerAddress: AddressTextInput.Text,
    CustomerPhone: PhoneTextInput.Text,
    ItemCount: ItemCount.Selected.Value,
    DeliveryOption: DeliveryOption.Selected.Value,
    Fragile: Fragile.Value
});

Reset(CustomerNameTextInput);
Reset(AddressTextInput);
Reset(PhoneTextInput);
Reset(ItemCount);
Reset(Fragile);
```

6.2 Processing Behaviour

- Defaults(DeliveryTable) creates a new record rather than updating an existing row.
- Patch() maps each input control to the corresponding DeliveryTable column.
- Reset() returns the five listed controls to their configured default values.
- DeliveryOption is not reset by the supplied formula and therefore may keep its current selection after saving.

6.3 Expected Navigation Logic

```
Navigate(Screen2);
```

This is the intended GoToSearch behaviour inferred from the screen layout and control name; the exact property formula was not supplied.

7. Search and Record Display

7.1 Screen2 - Search Screen

Sally

Blossom Residences

87654321

>

Jenny

Kennidale Hills

98765432

>

Harriet

Greendale Spring

98763322

>

Jamie

River Valley

67674545

>

Back

Figure 3. Searchinput and Gallery1 record list

Purpose: Searches and displays existing delivery records.

Screen name: Screen2

Control	Description
Searchinput	Accepts the user search keyword.
Gallery1	Displays matching or available delivery records.
Back	Returns the user to Screen1.

Search Logic

The exact Gallery1 Items formula was not included in the supplied formula extract. The implemented formula should filter DeliveryTable using Searchinput.Text and one or more searchable text columns, such as CustomerName, CustomerAddress or CustomerPhone.

Expected Back Button Logic

`Navigate(Screen1);`

8. Data Source Evidence

CustomerName	CustomerAddress	CustomerPhone	Fragile	ItemCount	DeliveryOption	PowerAppsid__
Sally	Blossom Residences	87654321				7bf722bb-ae52-42e0-b8f6-15d5c2e9e1fa
Jenny	Kennidale Hills	98765432	FALSE			72ee18e6-e836-4b9f-8088-c86f417b5c64
Harriet	Greendale Spring	98763322	TRUE			1fbe207d-edab-43df-9a3f-60c23015c922
Jamie	River Valley	67674545	TRUE		1 Deliver to Door Step	b11b1e5f-25f6-4ce5-93b2-07d1cc15f79a

Figure 4. DeliveryTable columns and example stored records

The data-source screenshot confirms the business columns CustomerName, CustomerAddress, CustomerPhone, Fragile, ItemCount and DeliveryOption, together with the connector-managed __PowerAppsid__ column.

9. Validation and Error Handling

No validation or error-handling logic was supplied. For a production implementation, the Save As formula should be extended to:

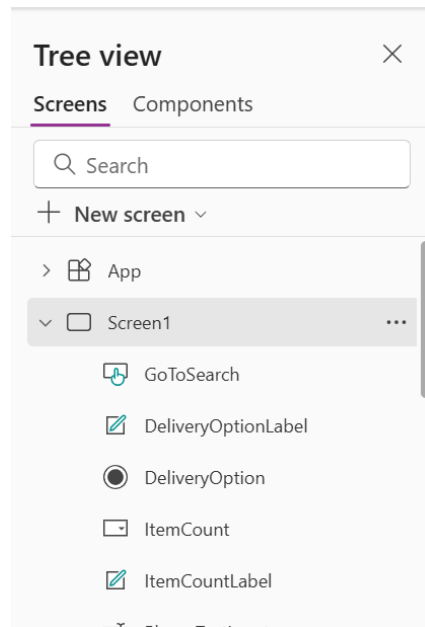
- Require the customer name, address, phone number, item count and delivery option.
- Validate the phone value and store it as text.
- Confirm that item count is numeric and greater than zero.
- Notify the user when the record is saved or when Patch() fails.
- Decide whether DeliveryOption should also be reset after saving.

10. Test Scenarios

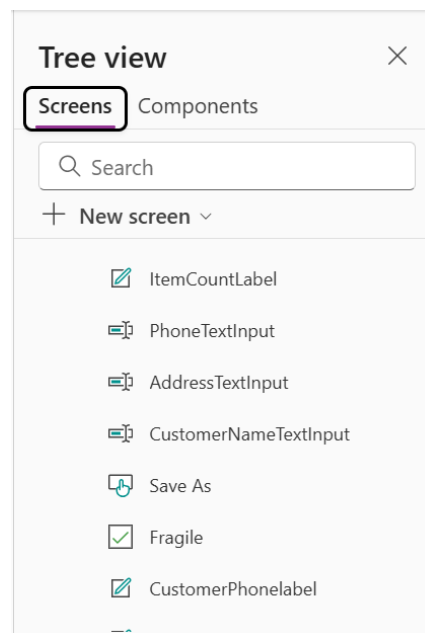
Test	Expected result
Save standard delivery	A new row is created with all six business fields.
Save fragile delivery	Fragile is stored as True.
Post-save reset	Name, address, phone, item count and fragile controls return to defaults.
Delivery option persistence	Selection remains unless Reset(DeliveryOption) is added.
Search records	Gallery1 shows records matching the search term.
Navigation	Start, Go To Search and Back open their intended screens.

Appendix A - Power Apps Control Tree Evidence

The supplied Tree view screenshots confirm the screen names and the principal controls used in the prototype.

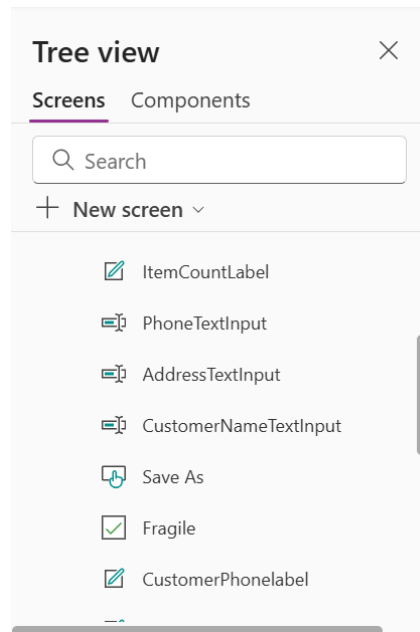


Screen1 controls: navigation, delivery option and item count

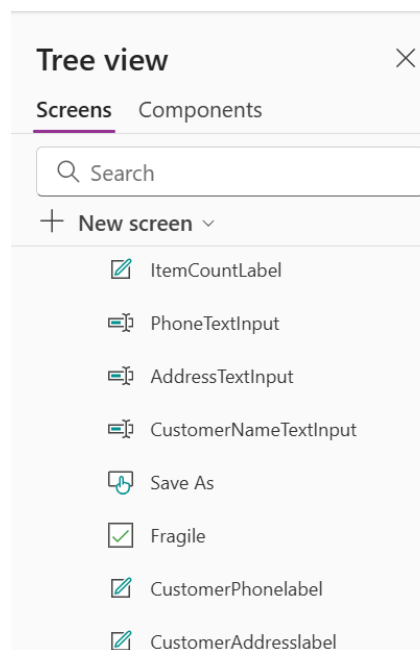


Screen1 controls: text inputs, save and fragile checkbox

Appendix A - Power Apps Control Tree Evidence (continued)

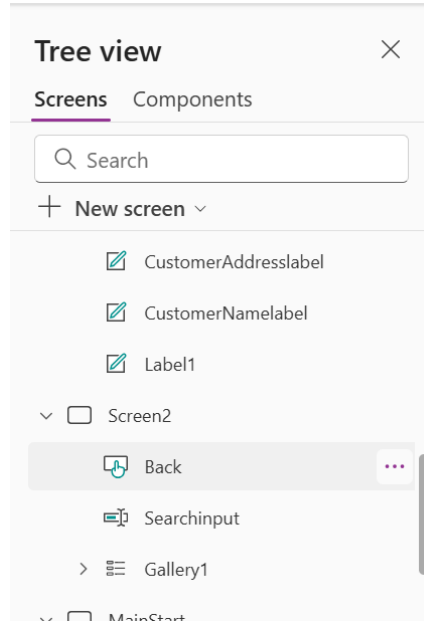


Screen1 controls: extended control list

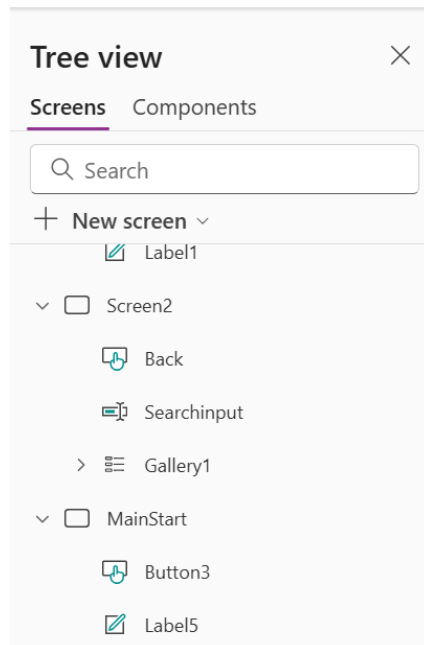


Screen1 labels and Screen2 Back control

Appendix A - Power Apps Control Tree Evidence (continued)



Screen2 Searchinput, Gallery1 and MainStart



Screen2 and MainStart control hierarchy

End of Technical Design Document.