

TECHNICAL DESIGN DOCUMENT

Employee Detail Recording System

Power Apps Canvas App Prototype

1. Document Information

Item	Description
Project Name	Employee Detail Recording System
Application Type	Microsoft Power Apps Canvas App
Primary Screen	Screen1
Data Source	Excel Online table: EmployeeTable
Version	1.0
Author	Adeline Lee
Date	8 September 2026
Purpose	Capture employee information and save it as a new data record.

2. Project Overview

2.1 Purpose

The Employee Detail Recording System is a single-screen Power Apps Canvas App that allows a user to enter an employee ID, name, phone number and department. Selecting Save Data creates a new record in EmployeeTable and clears the four input controls for the next entry.

2.2 Objectives

- Provide a simple employee information entry form.
- Create one new EmployeeTable record for each save action.
- Map each on-screen input to the correct data-source column.
- Clear the input controls after the save sequence completes.

3. System Architecture and Data Design

3.1 Technology Stack

Component	Technology / Role
User interface	Power Apps Canvas App
Business logic	Power Fx behavior formula on btn_Save_Data
Data storage	Excel Online table named EmployeeTable
Record creation	Patch with Defaults(EmployeeTable)

3.2 Logical Architecture

The user enters employee details in Screen1. The Save Data button reads the Text properties of the four input controls, uses Patch to create a default record in EmployeeTable, and then issues four Reset actions to return the controls to their default values.

Data flow: User input -> Save Data button -> Patch() -> EmployeeTable -> Reset() input controls

4. Data Model Design

4.1 Employee Entity

Table: EmployeeTable

Field name	Data type	Source / behavior
EmployeeID	Text	EmployeeID_TextInput1.Text
EmployeeName	Text	EmployeeName_TextInput2.Text
EmployeePhone	Text	Employee_Phone_TextInput.Text
EmployeeDept	Text	Employee_Dept_TextInput.Text
PowerAppsid	System-generated text/GUID	Managed by the Excel/Power Apps integration; not supplied by the formula.

4.2 Example Records

EmployeeID	EmployeeName	EmployeePhone	EmployeeDept
E3214	Jane Doe	81234567	HR
G4567	Jenny	98765432	Nursing

Design note: EmployeePhone is stored as text to preserve leading zeros, country codes and formatting.

5. User Interface Design

5.1 Screen1 - Employee Detail Form

Figure 1. Employee detail entry screen supplied for the prototype.

Purpose: Captures employee details and saves them to EmployeeTable.

Screen name: Screen1

5.2 Control Inventory

Control name	Type	Purpose
MainTitle	Label	Displays the employee form title.
EmployeeID_Label	Label	Identifies the Employee ID field.
EmployeeID_TextInput1	Text input	Captures EmployeeID.
Employee_Name_Label	Label	Identifies the Employee Name field.
EmployeeName_TextInput2	Text input	Captures EmployeeName.
Employee_Phone_Label	Label	Identifies the Employee Phone field.
Employee_Phone_TextInput	Text input	Captures EmployeePhone.
Employee_Dept_Label	Label	Identifies the Employee Department field.
Employee_Dept_TextInput	Text input	Captures EmployeeDept.
btn_Save_Data	Button	Creates a record, then resets the four inputs.

6. Implementation Logic and Verification

6.1 Save Data Button - OnSelect Formula

```
Patch(EmployeeTable, Defaults(EmployeeTable),
{
    EmployeeID: EmployeeID_TextInput1.Text,
    EmployeeName: EmployeeName_TextInput2.Text,
    EmployeePhone: Employee_Phone_TextInput.Text,
    EmployeeDept: Employee_Dept_TextInput.Text
});

Reset(EmployeeID_TextInput1);
Reset(EmployeeName_TextInput2);
Reset(Employee_Phone_TextInput);
Reset(Employee_Dept_TextInput);
```

6.2 Processing Sequence

- Patch targets EmployeeTable and uses Defaults(EmployeeTable), indicating that a new record is created.
- The record maps each EmployeeTable field to the Text property of its matching input control.
- Four Reset functions return the employee input controls to their configured default values.

6.3 Build Evidence

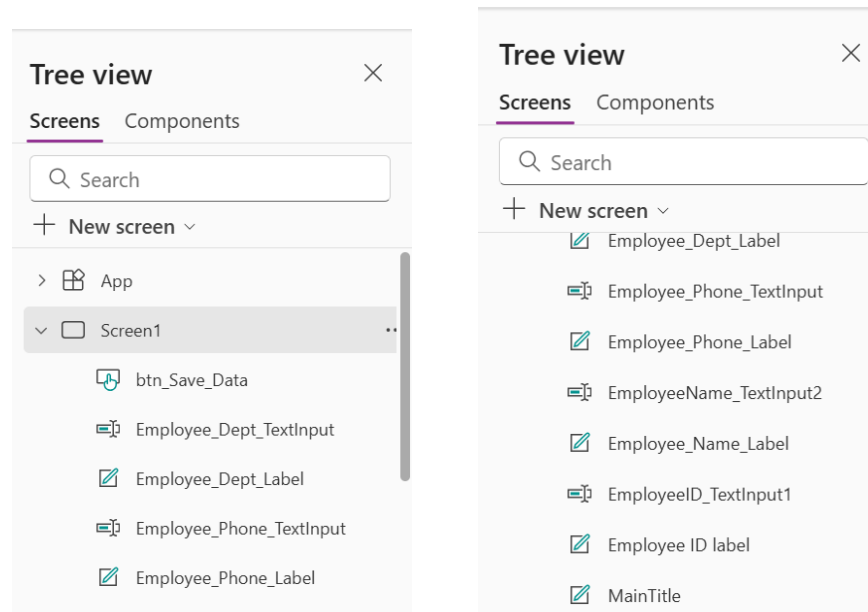


Figure 2. Power Apps tree view showing Screen1 controls.

EmployeeID	EmployeeName	EmployeePhone	EmployeeDept	PowerAppsid
E3214	Jane Doe	81234567	HR	56d02868-0c65-4e2e-93ff-bfb1d0b9edcb
G4567	Jenny	98765432	Nursing	d5315421-9b1b-4e44-bed3-289daf11a66a

Figure 3. EmployeeTable sample data after records have been saved.

7. Test Scenarios

Test	Action	Expected result
Save valid record	Enter all four values and select Save Data.	A new row is added to EmployeeTable with matching values.
Reset after save	Observe the form after the save sequence.	All four input controls return to their default values.
Repeated entry	Enter and save a second employee.	A separate new row is created; the first row remains unchanged.
Data mapping	Compare the saved row with the form values.	ID, name, phone and department appear in the correct columns.

8. Constraints and Recommended Enhancements

- The current formula has no required-field, phone-format or duplicate-EmployeeID validation.
- Add IfError and Notify so users receive a clear success or failure message, and reset fields only after a confirmed save.
- Excel Online is suitable for a small prototype; Dataverse or SharePoint is preferable when stronger security, relationships, concurrency and scale are required.

End of Technical Design Document