

Exploring Microsoft Power Apps for creating an analysis definition capture tool

Quincy Kaka, MSD (UK) Limited, London, UK

ABSTRACT

In today's pharmaceutical industry, the need for efficient, user-friendly, and scalable solutions is paramount. Microsoft PowerApps, a low-code platform known for ease of use and rapid development capabilities, benefiting from a built-in database and seamless integration with existing systems, can enable healthcare professionals to create tailored applications. Presented here is a summary lessons learned from exploring Power Apps' viability for producing a stakeholder analysis definition metadata capture tool with capture, read, update and delete functionalities. Key areas discussed are ease of learning and adoption of the platform, integration and interaction with both internal and external databases, as well as challenges faced during proof-of-concept development. This user experience revealed Power Apps' potential for streamlining internal workflows, aiding maintenance of standards, and providing overviews of project performance metrics.

INTRODUCTION

To meet increased analytic throughput demands, Power Apps was explored for its viability as a development platform for creating an application to perform capture, read, update, and delete functionalities based on user interactions and inputs. The obtained metadata would guide downstream data analyses. This two-week exploration phase, undertaken with no prior Power Platform experience, resulted in a functional proof-of-concept demo. In this paper I share my experience and insights with using the platform and its tools.

Microsoft Power Platform is a set of tools designed to empower users to develop business solutions rapidly. Specific tools explored were Dataverse, the underlying database built into the platform, and Power Apps, the development tool, providing two application types: model-driven and canvas apps.



Figure 1, Graphic visualization of Microsoft Power Platform.

TYPES OF POWER APPS

An overview of key features and differences between model-driven and canvas apps is highlighted in the below table.

Table 1, Overview of canvas and model-driven app features.

Category	Model-driven apps	Canvas apps
Data platform	Dataverse only	Dataverse + many others using connectors
Design experience	No-code component focused design	Manipulation of control properties using Power Fx expressions
UI control	Limited, predominantly customization	Full control
App consistency	High – differs predominantly based on the tables and views chosen	Often low, given the significant control the designers have of the user experience
Responsive & Navigation	Automatically responsive	Only where designed and applied using Power Fx formulas

Model-driven apps are built on top of Microsoft Dataverse and thus require all relevant data to be hosted there. Users choose screens and views to determine the content, flow and navigation of the application, through no-code component and logic building. While it has limited customizability of the user interface (UI), apps produced have a highly consistent feel.

Canvas apps offer higher UI customization and database choice flexibility. Users manually design screens by drag-and-dropping UI elements and defining their properties via Power Fx expressions, similar to Microsoft Excel formulas. For the metadata capture tool's necessary UI and functionality, a canvas app was chosen for its flexibility.

BUILDING CANVAS APPLICATIONS

Developing canvas applications follows a simple workflow: UI elements are inserted to screens via drag-and-drop to design page layouts, connected to data sources, formatted, and their interactions and logic defined. Each screen represents a page of the application.

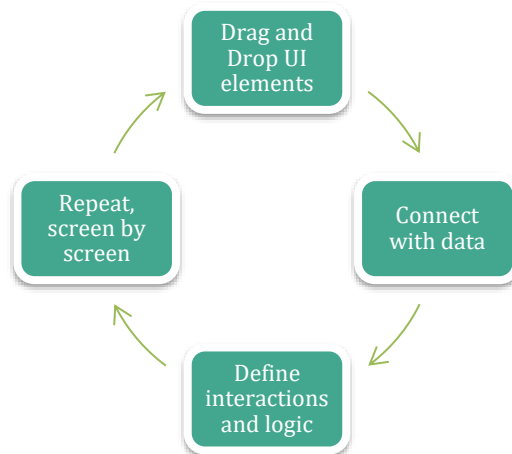


Figure 2, Canvas App workflow overview.

Power Apps can connect to over a thousand external data sources, including MS Excel, SharePoint, and cloud services like AWS S3 buckets, allowing great implementation flexibility. Alternatively, the built-in Dataverse database can be utilized, a secure cloud-based storage solution with version control and role-based access assignment features. Tables created in Dataverse are enriched with metadata and unique record identifiers.

DEVELOPMENT EXPERIENCE

Two key challenges were faced: to create an application that enables users to view, create and edit records within a database, and to be able to assign and control user access to the database.

DATABASE SOLUTION

Hosting data within Dataverse was a convenient solution thanks to its ease of integration and fast, dynamic response of live data to inputs from within the application. Each table within Dataverse is enriched with additional metadata such as unique record identifiers, creation and modification dates as well as details of users interacting with tables and records. These

features allowed very efficient querying of specific table records, as well as version control from whole tables to individual records.

Created On	Created By	Modified By	Modified On	Owner	Population
9/25/2024 3:45 PM	Quincy Kaka	Quincy Kaka	10/1/2024 12:35 PM	Quincy Kaka	921d8be0-8340-4554-9622-04031d32e5e6
9/25/2024 3:45 PM	Quincy Kaka	Quincy Kaka	9/25/2024 3:45 PM	Quincy Kaka	750e033d-31f2-4d18-bdab-4ecc7068439e

Figure 3, snapshot of metadata enrichment of tables hosted in Dataverse, including creation and modification dates, users and unique record identification IDs.

Additionally, it is possible to create multiple environments on the platform so that development and production versions of the application and database can be hosted. Both tables and applications can be exported and imported between environments.

USING POWER APPS PLATFORM

The drag-and-drop design framework and formula-like logic definitions made Power Apps tools easy to accustom to, aided by the broad range of learning resources and online support available. A wide array of elements was available for creating an interactable interface, particularly for capturing user inputs with text inputs, dropdowns, forms, etc. These elements were linked to data with a few clicks and simple data transformations could be executed on them with ease.

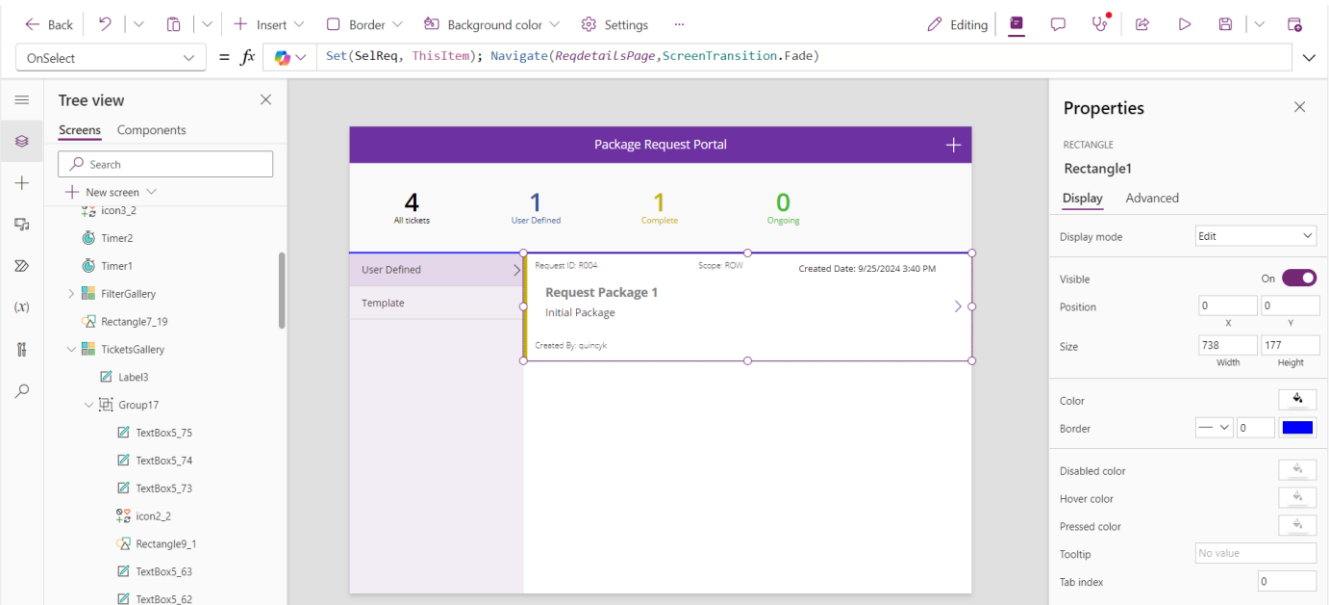


Figure 4, snapshot of Power Apps canvas app development user interface (UI). Screens representing app pages and UI elements can be navigated and accessed via the left panel. Properties of elements can be modified via the right panel, as well as through the function bar at the top. In the shown example logic is being defined for the condition of the element being clicked, setting a global variable and navigating to page.

The tool provided two methods of previewing the application. Firstly, a preview mode which launches a fully interactive app from the current screen. Secondly, elements can be clicked in a semi-interactive manner to trigger single steps of the application without leaving the developer mode. These options were fast and allowed testing of features as soon as they were developed in a seamless manner.

Something to note was that editing the application was only possible by one person at a time, which can be limiting for collaborative work.

DEALING WITH INCREASED COMPLEXITY

While Power Apps' high customizability was advantageous, creating processes with significantly complex logic led to an exponential increase in development difficulty. Scaling up create/read/update/delete functionalities was a challenge. Another was creating pages that display many interacting data dimensions and splitting these into excessive number of screens and pages to navigate through can be detrimental to user experience .

CONCLUSION

During this two-week exploration with no prior experience, Power Apps demonstrated great viability as a low-code application development platform. Leveraging Dataverse and multiple environments proved extremely useful and have been adopted for long-term use, with applicability outside Power Apps. Power Apps is recommended for producing solutions that augment project management, such as creating and maintaining databases or providing performance metric dashboards and project overviews. While challenges may be faced with highly complex process requirements, the ease of adoption by novice users can be of great benefit and save on IT resources by bringing development and maintenance inside teams.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Author Name: Quincy Kaka

Company: MSD (UK) Limited, London, UK

Address: 120 Moorgate, London, EC2M 6UR, United Kingdom

Email: quincy.kaka@msd.com

Website: MSD.com

Brand and product names are trademarks of their respective companies.

REFERENCES

Figure 1 - <https://learn.microsoft.com/en-us/power-apps/maker/data-platform/data-platform-intro>