Microsoft IT Showcase

Creating a sales productivity solution with PowerApps

At Microsoft, sellers rely on a powerful set of tools that support the sales process and improve collaboration. To further enhance the seller experience and increase productivity, we created SalesHub, a mobile application built with PowerApps. SalesHub gives sellers access to curated pre-sales assets and tools they need in a simple, mobile application. SalesHub makes it easier for sellers to prepare for customer meetings and helps our teams to work better together.

Augmenting the sales environment at Microsoft

To prepare for customer engagements, sellers spend hours each week connecting to multiple data sources to get the right information about the customer, the opportunity at hand, and the activities that have been completed. In addition, our sales force is constantly mobile, so they have limited time for training and learning about new products, promotions, and offers. While they can complete these tasks on their laptop, it is not always practical for someone who is always on the go. Some of the pain points our sellers typically face include:

- Spending time searching multiple data repositories.
- Getting easy access to the latest sales playbook.
- Keeping up to date on new products, programs, and initiatives while on the go.
- Keeping track of all the steps in a complex sales opportunity.

Developing a single solution with PowerApps

We realized that with their growing portfolio, our sellers needed a more efficient and flexible method to access pre-sales material. We also wanted to augment the customer relationship management (CRM) system by providing a mobile app that gives sellers real-time visibility into all aspects of an opportunity. Our team assembled some basic goals that we wanted to achieve for our sellers:

- Unify and present information from multiple data sources.
- Provide learning opportunities for sellers on the go.
- Create a platform that was accessible from anywhere.

With these goals in mind, we developed a plan to create an app that would offer curated content to our sellers and give them the tools they need to be productive on the go. We selected PowerApps as the platform for our app unification project. PowerApps gave us the flexibility to merge our apps into a single experience, offer a user-friendly and attractive UI, and integrate with other common cloud and mobility data sources.

PowerApps is a collection of software services that offer a more direct, simple way to develop and integrate apps. It connects to existing cloud services and data sources and helps to quickly build apps to meet specific needs. PowerApps also offers enterprise-grade capabilities combined with easy-to-use authoring. PowerApps consists of four major components:

- **web.powerapps.com.** This is the portal where you manage and share the apps you build.
- **PowerApps Studio.** You use PowerApps Studio to build powerful apps with easy-to-use visual tools.
- **PowerApps Mobile.** Apps developed in PowerApps run on mobile devices using PowerApps Mobile. Use PowerApps Mobile to run apps on Windows, iOS, and Android devices.
- **PowerApps admin center.** Use to administer PowerApps environments and other components.

Prototyping PowerApps

Using PowerApps, a member of our team designed a functional prototype application to present how our application, SalesHub, would look and interact with the variety of data sources and tools in the sales ecosystem. The prototyping process was done by a power user on our team, not developer. It took the user a week to develop a functioning prototype, which enabled us to put the prototype in front of stakeholders so they could make a business decision quickly. The ability to quickly create a basic
solution like this in PowerApps accelerated our development process. It enabled us to embrace an agile approach to development and ensured that our business requirements were met at every step in the development process.

System architecture

All parts of SalesHub are connected through the interface, including data sources, user interface, functions, and in-app processing shown in Figure 1.

Figure 1. SalesHub high-level design architecture

PowerApps provides a GUI for connecting to data sources, such as Azure SQL and multiple APIs. SalesHub also includes the following components:

- **Azure SQL.** An Azure SQL database hosts app data. User information, content, Content Data Finder (CDF) data, and usage analytics are all stored in the database.

- **QnA Maker.** The Modern Workplace Bot (MW Bot) module uses the QnA Maker Azure bot service. The service automatically generates an API endpoint once the knowledge base is populated. The connection to the bot was originally made by following a tutorial called *PowerBots—Creating smart apps with PowerApps and Microsoft Bots.*
During implementation, the development team decided to use an Azure Function App wrapper around the QnA Maker. The PowerApps custom connector interface required an authentication method (Ocp-Apim-Subscription-Key). Users had to enter a long subscription key that, for security purposes, could not be circulated to thousands of users worldwide. The solution was to create a C# function that receives the user’s request and connects to the bot API with the key in its request, instead of using the PowerApps connection directly.

- **Azure Function App.** The function app was used to wrap the MW Bot API endpoint to simplify the user experience. Written in C# script and hosted on the SalesHub Azure subscription. The function app uses the Newtonsoft.Json framework to connect to the QnAMaker bot API. The function app acts as its own API and does not need authentication from the user.

- **LUIS Integration.** Language Understanding Intelligent Services (LUIS) is a service from Microsoft Cognitive Services that provides an interface to make bots that process language to a deeper degree than the QnA Maker. We integrated LUIS to enhance the capabilities of the MW Bot, to catch questions the QnA Maker missed.

  LUIS provides an API endpoint very similar to the QnA Maker, so integration was a matter of sending a second request with the same query to a different location. Currently, LUIS is only initiated when the QnA Maker cannot find a match in its knowledge base. LUIS attempts to diagnose the intent and associated entities (if detected), then the output is sent back to the QnA Maker as a refined query.

**Design considerations**

We took several app design elements into consideration when determining how our solution would function. These considerations helped shape how we would build the app modules and how individual modules and data repositories were designed, both on their own and in interaction with each other.

- **Performance.** Performance was a high-priority, non-functional requirement for the sales team. To increase the performance of the application, we wanted to prefetch data while the user reads the current screen. The primary design pattern is using collections in memory because the data populates faster if it’s already loaded. Prefetching is done by collecting data from external data sources.

- **Design of CDF entity relationships in PowerApps.** The CDF Orchestrator uses multiple entities within SalesHub. The tool is constructed with the programming features of PowerApps to create an apparent relationship between these entities. The relationships between Account Tracker, Activity Tracker, and Team Member make up the data for this tool. To display data for a given Account Tracker, we saved the selected Account Tracker in a local collection and passed it to the built-in filter() method.

- **High-risk components.** The Azure SQL database has the highest exposure to risk in SalesHub. Data is not necessarily sensitive, but the fact that the data is constantly in transit puts it at risk. The inability to easily access and review session logs due to the nature of abstraction in the PowerApps tool furthers this risk. To mitigate this risk, we ensured that the Azure subscription that hosts SalesHub complies with the standards laid out in the Secure DevOps Kit for Azure.
User experience components

The SalesHub experience was designed to be a central console for sellers that provides access to any part of the app they need in three clicks or less. The Welcome screen, shown in Figure 2, has links to all the apps, functional components, and information about the seller’s current quota attainment.

![Figure 2. The SalesHub app Welcome screen. Prototype version on the left and current release on the right.](image)

All SalesHub users have the same level of access to every module of the application. The first time a user enters the application, a new user record is created in the database and the application launches the home page. This table stores employee ID, color preference (automatically set to blue), user role, and the date the record was created. The user’s role is retrieved from a static list of technical sellers based on the user’s email alias. If the user’s email alias is not in the static list, then the role is assigned to the default value.

The Home screen

The main app functions link the user to individual app features. The home screen displays the list of SalesHub modules: What’s New, Content Finder, Learning Channel, CDF Orchestrator, Who, and MW Bot. The navigation bar at the bottom takes users to the User Favorites and Settings pages. The Data Protection Notice link takes the user to the Microsoft data protection notice web page.

- **What’s New screen.** The newest content posted for the Content Finder and Learning Channel modules are displayed here. The first card is labelled Featured Content and is populated with the five newest items in the Content Finder (with links), which are assigned to the user’s role.
- **Content Finder screen.** The Content Finder links to content on SharePoint, which is displayed in a gallery by content title.
- **Learning Channel screen.** Users can select, search for, and view Learning Channels videos. Data for the Learning Channel screen is stored and accessed using the PowerApps SQL connector.
- **CDF Orchestrator screen.** The CDF Orchestrator is a case management tool used to track progress on Microsoft customer accounts. Three tabs, My accounts, My tasks, and Browse, list active customer accounts, tasks that are assigned to the user, and links to the activities in that task’s workload. Users can search all current Microsoft business customer accounts by CRM account name. The results provide a list of matching query result names and subsidiaries (or countries). CDF Orchestrator not only displays data from the data source, but also aggregates and calculates data based on user requests.
- **Who.** Who uses the built-in PowerApps Office 365 connector to enable users to search their Office 365 organization directory for employees by first and last name. When a search result item is selected, the application provides basic information such as job title, department, manager, and a list of reports. The Profile tab is selected by default, and the Contact tab shows contact name, office location, email, and mobile phone number. Users can send email or place a call directly from the app.
• **MW Bot.** The Modern Workplace Bot uses a PowerApps custom connection to look up question-answer pairs stored in the SalesHub QnA Maker Bot. Users can type in questions or keywords and the bot API returns the best matched answer from the knowledge base.

• **Favorites.** Favorites data is stored in the Azure SQL database using the PowerApps SQL connector. Links from Content Finder and videos from Learning Channel are on separate tabs at the top of the screen. Users can search for favorites by title on the selected tab.

• **Settings.** Settings, reached using the cog icon in the bottom navigation, lets users update color preferences and role settings. This data is stored in the Azure SQL database.

### Implementation and deployment

When developing and implementing SalesHub, we used the PowerApps platform to apply several significant engineering principles within the app.

• **Instrumentation.** PowerApps and Azure were the primary development tools used to produce the app. PowerApps was chosen because we wanted to control costs, and using Xamarin would require hiring a C# full stack developer. PowerApps was also used to build this application as a showcase for sellers to demo to customers and for the PowerApps Success Team. One benefit of the PowerApps enterprise subscription is that the Success Team provides continuous support to customers who are developing business apps. Connecting to SQL and some APIs was easy, and the web interface offered a useful GUI to connect to databases.

• **Security and usage logging.** In the PowerApps details, on the Analytics tab, is a Power BI dashboard that provides usage information such as the number of active users, device types, player versions, and sessions. Our team was interested in viewing user behavior, so we created a reporting mechanism that uses SQL and Power BI to visualize page visit and link trends.

• **Secure access.** One advantage of using PowerApps is the built-in security system. Since users can only access PowerApps through their organization login credentials, the security management that dictates all other Microsoft software access is part of the same subscription that PowerApps uses. There was no need to implement a security module for this project. Security logs are collected and managed by the PowerApps team.

• **Privacy.** Privacy was a high priority during the security review process. Personal information is collected for dashboard use, but the fact that the app is used internationally meant we needed to make sure we were compliant with the Azure subscription regulations and guidelines. This required us to run Azure analysis reports and scripts from the Secure DevOps Kit for Azure to update the security levels of various components.

• **Continuous integration and continuous deployment.** PowerApps hosts its own cloud-based version control, which reduces the amount of resources we had to spend on basic project management. Deployment is very easy because the app player that users access from their phones links to the currently published application. Publishing the app is as simple as pressing a button and users immediately have the latest version.

### Benefits

With SalesHub, we are improving seller productivity by reducing the time they spend looking for the right pre-sales assets, surfacing the right content, and enabling them to keep current on the latest product updates and initiatives. In addition, SalesHub is helping to increase adoption of key sales processes, which drives better visibility into complex sales opportunities and, ultimately, helps sellers close their deals faster and maximize revenue potential.

### Moving forward with SalesHub

SalesHub has become a key asset in our sellers’ toolkit, and it is a big part of the increased productivity and effectiveness our sellers are experiencing. PowerApps allowed us to quickly and efficiently put SalesHub in the hands of our sales and marketing teams, and the modular, cloud-based, mobile nature of SalesHub ensures that it can be adapted to grow and change with business demands. We will continue to grow SalesHub functionality in the future and integrate it deeper into our sales processes.
For more information

Microsoft IT Showcase
microsoft.com/itshowcase
Secure DevOps Kit for Azure

© 2017 Microsoft Corporation. This document is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS OR IMPLIED, IN THIS SUMMARY. The names of actual companies and products mentioned herein may be the trademarks of their respective owners.