A MODEL DRIVEN APP FOR BOOKING A TAXI.

IXEAD

BY

ONESIMUS AWONIYI

Contents

Scope of Work

Proposal is based on the following:

- You need to create a **Model Driven App** (not Canvas app) to be used as a back-end system for a taxi company.
- The app should allow to create and manage customers. Each customer should be unique and the system should not allow duplicate customers.
- The app should allow to create and manage bookings. A booking represents a customer request for a taxi which is updated with all necessary information at all stages until the service is completed (trip ended).
- Users should be able to set the appropriate status of each booking via a dropdown.
- Each booking should be identified by a unique auto incremented number having a static prefix 'IXD' following with a 6 digits number.
- A customer can have multiple bookings.
- Set up the Booking form with columns (fields) you believe should be requested in a booking form. As part of the fields of the Booking form, include question 'Travelling with Pet?' (Yes/No). If this question is answered as Yes, a mandatory free text field 'Pet Comments' should show in the form. Otherwise, this free text field should be hidden and not mandatory.
- A booking can have one or multiple passengers.
- The booking should define the Pick-up and Drop Off locations. The customer might request multiple Drop Off locations. When inputting the Pickup and Drop Off locations, the user should also select the locality from a pre-defined list.
- The booking form should display a Business Process Flow to guide the booking agents to follow a standard process.
- Users should be able to select from different views by Status for the booking area.
- A dashboard should be available in the app displaying 3 different visuals based on different areas.
- Create a button with Java Script in the Booking form which when clicked it will set the Status of the Booking to 'Trip Ended'.
    - This button should have an icon.
    - This button should be visible only if the Status of the Booking is not 'Trip Ended'.

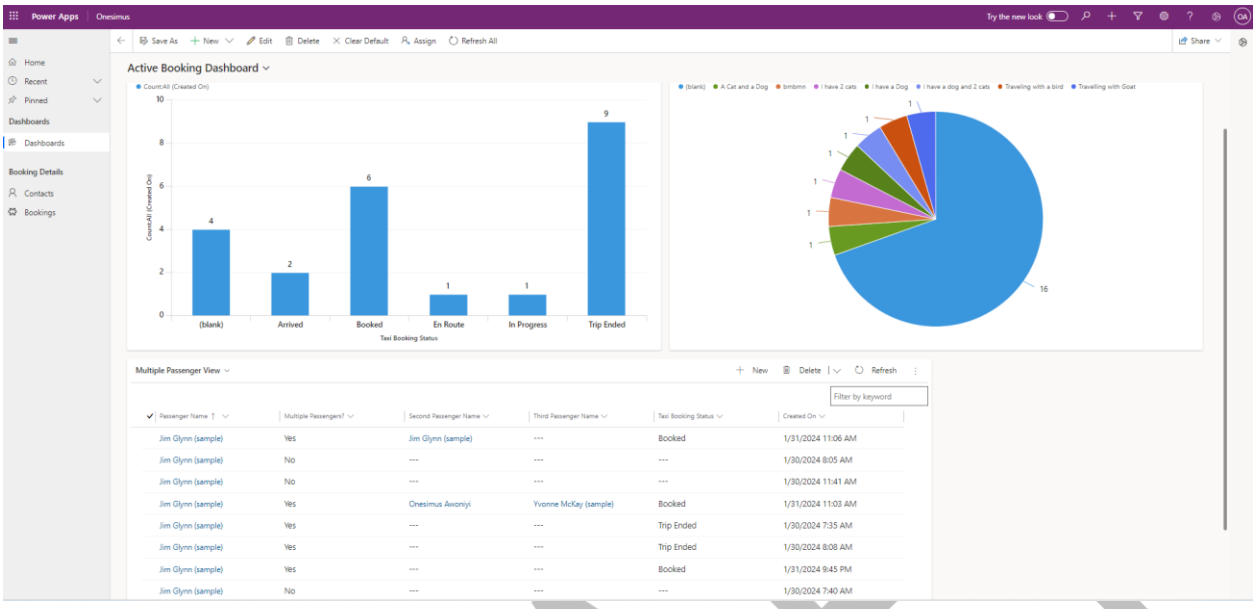
- Create a plugin which populates a text field called ‘Passengers Names’ shown on the Booking form with the names of the passenger’s comma separated.

Technical Solution:

The proposed technical solution for this project is to develop a custom Model-driven application leveraging Out-Of-Box processes, a dashboard for a holistic report overview and custom logics for process optimization. Model-driven app design is an approach that focuses on adding components such as forms, views, charts, and dashboards to tables using an app designer tool. Additionally, relationships connect tables together in a way that permits navigation between them and ensures data is not repeated unnecessarily. Power Automate will aid the automation of business processes, orchestrate processes end-to-end and trigger microservices, and move business data between systems on a schedule among others.

Modules and Features:

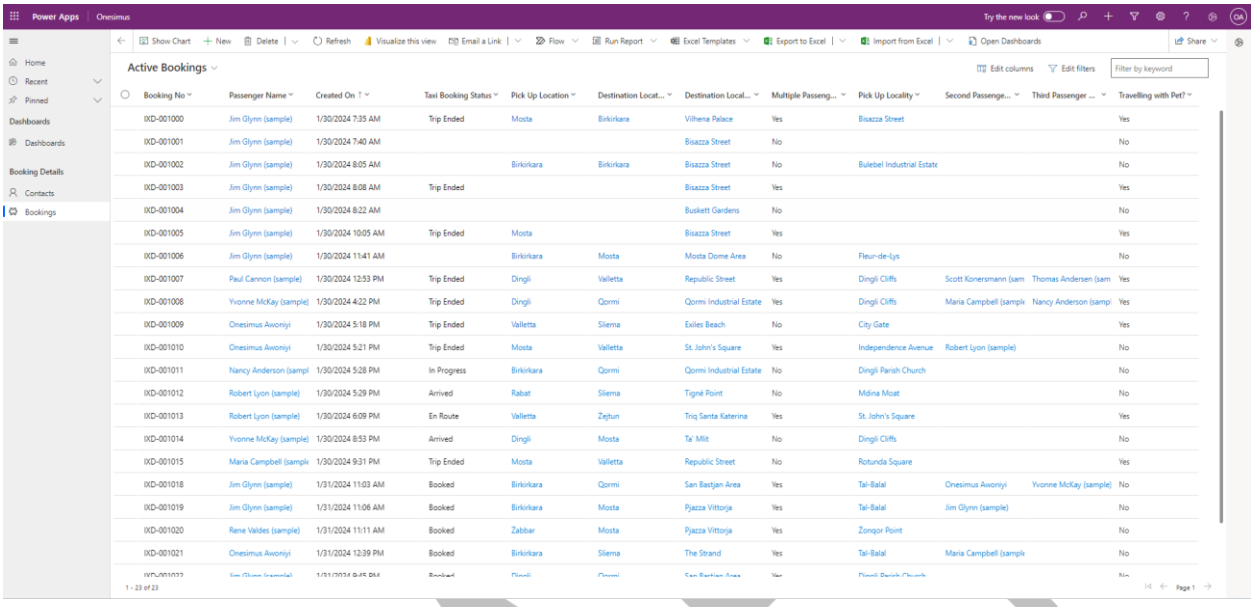
1. **Model-driven application**: Model-driven app design is an approach that focuses on adding components such as forms, views, and charts and dashboards to tables using an app designer tool. Additionally, relationships connect tables together in a way that permits navigation between them and ensures that data is not repeated unnecessarily. From the user's perspective, the model-driven apps offer an experience similar to the described, which is both accessible to many users and to the device used. The experience is similar to the diagram shown below. In the hypothetical example illustrated below, the app contains two tables (Customer and Booking), one dashboard, and multiple charts and views. Users can navigate between the tables using the left pane or via the dashboard.
2. **Dashboard**: Dashboards are collections of charts relating to Microsoft Dataverse tables. It's a container for one or more charts relating to a table. A dashboard allows charts, Power BI reports, and views of tables to be presented to the app user.
3. **Process of Record Creation**: Leverage on Power Automate to create a seamless business process flow that would walk the end user through the steps of requesting for a taxi/ride.

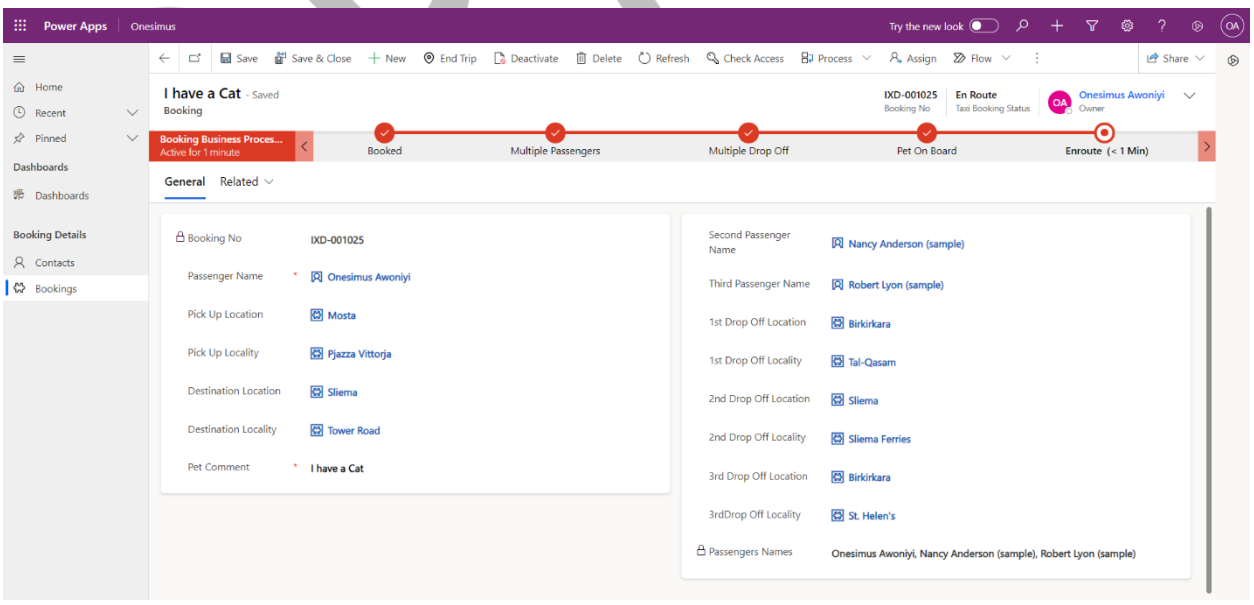


1 Functional Requirements

The Solution would satisfy the following functional requirements:

1. **Model-driven application**: Model-driven app design is an approach that focuses on adding components such as forms, views, and charts and dashboards to tables using an app designer tool. Additionally, relationships connect tables together in a way that permits navigation between them and ensures that data is not repeated unnecessarily. From the user's perspective, the model-driven apps offer an experience similar to the described, which is both accessible to many users and to the device used. The experience is similar to the diagram shown below.
2. **Dashboard**: Dashboards are collections of charts relating to Microsoft Dataverse tables. It's a container for one or more charts relating to a table. A dashboard allows charts, Power BI reports, and views of tables to be presented to the app user.
3. **Process of Record Creation**: Leverage on Power Automate to create a seamless business process flow that would walk the end user through the steps of requesting for a taxi/ride.


Power Apps
Onesimus
Active Bookings
Booking No
Passenger Name
Created On
Taxi Booking Status
Pick Up Location
Destination Locat...
Destination Local...
Multiple Passeng...
Pick Up Locality
Second Passenge...
Third Passenger ...
Travelling with Pet?


Power Apps
Onesimus
I have a Cat - Saved
Booking
IXD-001025
Booking No
En Route
Taxi Booking Status
Onesimus Awoniyi
Owner
Booking Business Proces...
Active for 1 minute
Booked
Multiple Passengers
Multiple Drop Off
Pet On Board
Enroute (< 1 Min)
General
Related
Booking No
IXD-001025
Passenger Name
Onesimus Awoniyi
Pick Up Location
Mosta
Pick Up Locality
Pjazza Vittorja
Destination Location
Sliema
Destination Locality
Tower Road
Pet Comment
I have a Cat
Second Passenger Name
Nancy Anderson (sample)
Third Passenger Name
Robert Lyon (sample)
1st Drop Off Location
Birkirkara
1st Drop Off Locality
Tal-Qasam
2nd Drop Off Location
Sliema
2nd Drop Off Locality
Sliema Ferries
3rd Drop Off Location
Birkirkara
3rdDrop Off Locality
St. Helen's
Passengers Names
Onesimus Awoniyi, Nancy Anderson (sample), Robert Lyon (sample)

2. Non-Functional Requirement

1. **Security:** The proposed solution will adhere to the highest standards of security to protect sensitive information.
2. **Scalability:** The solution will be scalable to accommodate growth and changes in the future.
3. **Performance**: The solution will be optimized for performance to ensure quick and efficient access to information.

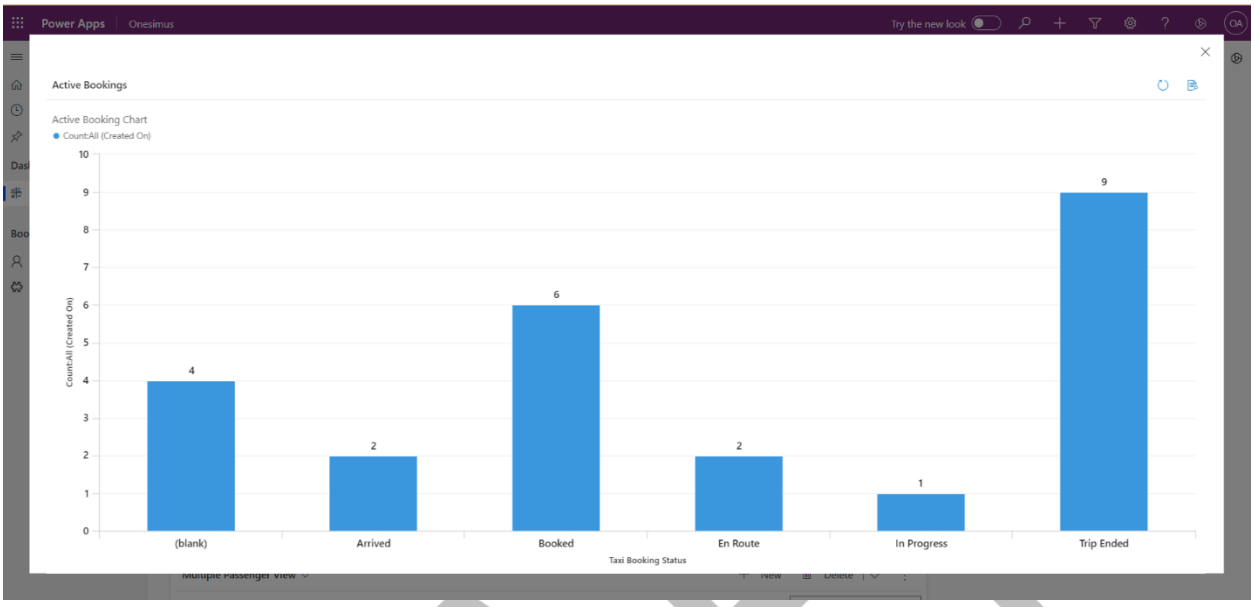
3. Solution Overview

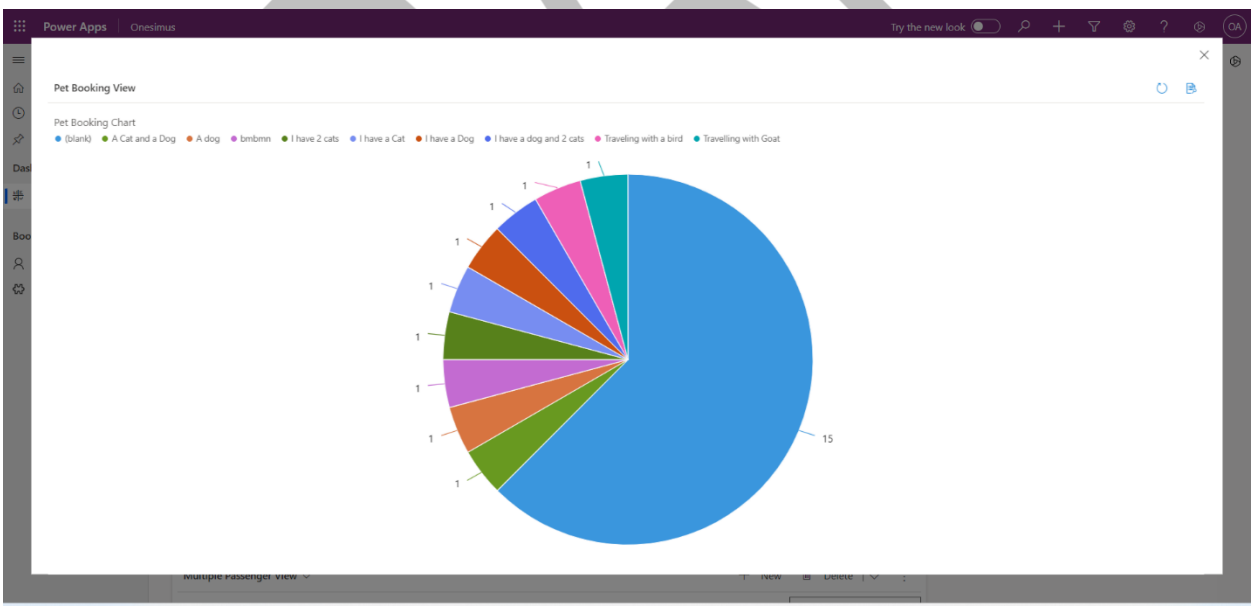
The proposed solution will utilize two Microsoft Power Platform products as listed below:

- Model-Driven App
- Microsoft Power Automate

3.1 Model-driven App

Model-driven app design is an approach that focuses on adding components such as forms, views, and charts and dashboards to tables using an app designer tool. Additionally, relationships connect tables together in a way that permits navigation between them and ensures that data is not repeated unnecessarily. From the user's perspective, the model-driven apps offer an experience similar to the described, which is both accessible to many users and to the device used. The experience is similar to the diagram shown below. In the hypothetical example illustrated below, the app contains three tables (challenges, ideas, team projects), one dashboard, and multiple charts and views. Users can navigate between the tables using the left pane or via the dashboard.


Power Apps
Onesimus
Try the new look
Active Bookings
Active Booking Chart
Count:All (Created On)
10
9
8
7
6
5
4
3
2
1
0
Count:All (Created On)
4
2
6
2
1
9
(blank)
Arrived
Booked
En Route
In Progress
Trip Ended
Taxi Booking Status

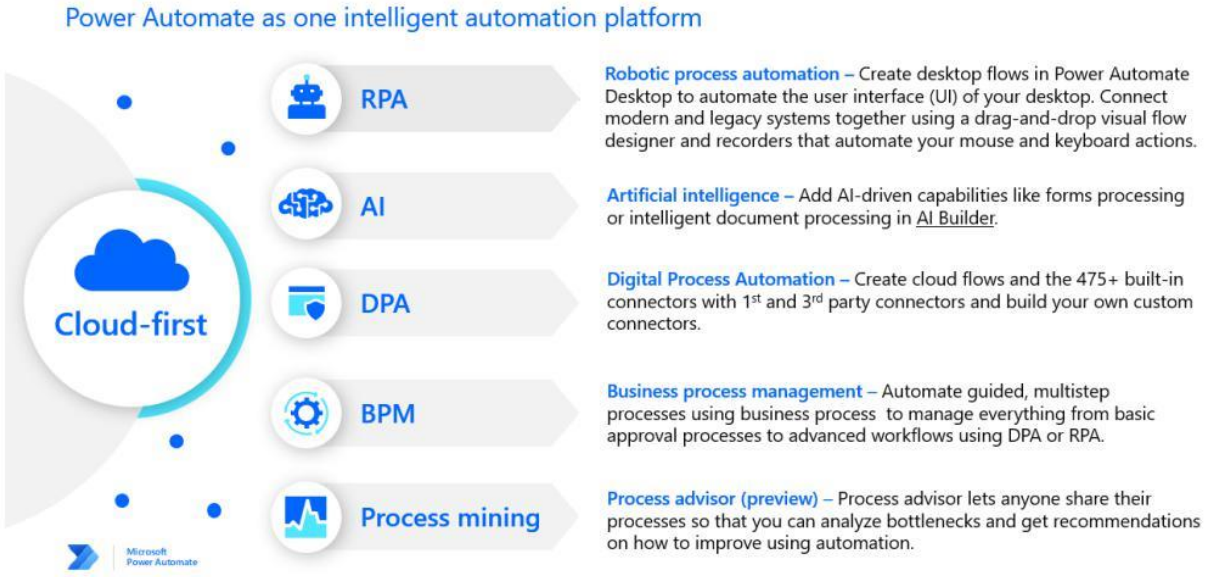

Power Apps
Onesimus
Try the new look
Pet Booking View
Pet Booking Chart
(blank)
A Cat and a Dog
A dog
bmbmn
I have 2 cats
I have a Cat
I have a Dog
I have a dog and 2 cats
Traveling with a bird
Travelling with Goat
1
1
1
1
1
1
1
1
1
15

3.2 Microsoft Power Automate

Microsoft Power Automate is a service that helps create automated workflows between business apps and services to synchronize files, get notifications, collect data, and more. Automated workflows orchestrate across services using connectors and can be triggered to run when events occur in other systems and services or scheduled to run at a specific time. There are two different types of flows available in Power Automate:

- Cloud flows are executed in the Power Automate service. These typically use API connectors to automate across applications, services, and databases in the cloud or on-premises—also known as Digital Process Automation (DPA).
- Desktop flows run on customers' virtual or physical machines. Previously called UI flows, these automations usually perform UI-based automations and are sometimes called Robotic Process Automation (RPA).

With Microsoft Power Platform, Automation is not an island and works as one intelligent automation platform.



Sincerely,

ONESIMUS AWONIYI
Ones.itpro@gmail.com
+2347036856913