## 1. Requirement Gathering

It is further divided into 4 sub-steps for a clean understanding or process.

## a. Identify Stakeholders:

We need to identify a key contact person who is either a domain expert or a leader expected to use the dashboard. This step is crucial for obtaining clarity on whom to communicate with and seek feedback from during the project.

# b. Understand Business Objectives:

Through meetings and calls with stakeholders we will get an outline of the whole endeavor primarily to achieve how this dashboard will help achieve a specific business goal.

# c. High Level Data Study:

This is just a high-level overview of data and not a dig deep. In this phase, data sources, column description, data type, volume or frequency or data quality is studied.

# d. Scope:

In this phase, key metrics, Document KPIs, deadlines, expectations etc are discussed with stakeholders.

After all the above steps, our overall objectives of the project are –

# **♣** Project goals:

- Track current status of patient waiting list.
- Analyze historical monthly trend of waiting list in Inpatient & Outpatient categories.
- Detailed specialty level & age profile analysis.

# **Language** Data Scope:

-2018-2021

#### **Metrics Required:**

- Average & Median Waiting List
- Current Total Wait List

## **♣** Views Required:

- Summary Page
- Detailed Page for Granular Analysis

#### 2. Data Collection

Here, the data source has been determined, influencing how the dashboard will be updated post-deployment of the solution. In this project, a folder serves as the data connector. The dataset utilized is publicly available, and a link to it is provided at the conclusion of this document.

Prior to importing the dataset into Power BI, especially when dealing with multiple files, it was essential to verify that all files shared the same number of columns and had identical column headers. Having confirmed these criteria, the data was loaded into Power BI using the "Get Data" option. The 'folder' option was selected as the data connector, and by clicking 'combine and load,' the data was successfully imported into Power BI.

The dataset comprises two folders: 'Inpatient' and 'Outpatient.' The data loading process occurred sequentially, with data from the 'Inpatient' folder loaded first, followed by data from the 'Outpatient' folder.

# 3. Data Transformation & Modelling

# 3.1 Transformation of data: Power query tab in PowerBi.

After meticulously inspecting all the columns for irregularities to ensure accurate data loading, I identified a minor discrepancy in the 'Archive\_Date' column. Although designated as a date column, it was incorrectly specified as a text column. I rectified this issue by adjusting its data type using locale settings, a step that could also be achieved by modifying regional settings.

Now, the goal is to merge both the inpatient and outpatient datasets into a unified dataset for seamless visualization. For this to happen, both tables must share the same number of columns and identical column names to expedite the process.

Upon reviewing both the inpatient and outpatient tables, two disparities came to light. Firstly, the 'Inpatient' table contained an additional column named 'Case\_Type.' To facilitate the append process, I added a corresponding column to the 'Outpatient' table with the same name, 'Case\_Type,' populated with a consistent value for all records: 'Outpatient.'

The second difference involved the column naming convention for specialties. In the 'Inpatient' table, it was labeled 'Specialty\_Name,' while in the 'Outpatient' table, it was simply 'Specialty.' To maintain uniformity and streamline the subsequent append process, I renamed the 'Specialty' column in the 'Outpatient' table to 'Specialty Name.'

Using append queries, a new table was created, consolidating both datasets. The resulting table was aptly named 'All Data' for clarity and coherence in the visualization process.

## 3.2 Modelling of data: Modelling tab in PowerBi.

Within the Modeling tab, you'll find the display of all three tables—'Inpatient,' 'Outpatient,' and 'All\_Data.' Since the necessary data from both inpatient and outpatient tables is consolidated in the 'All\_Data' table, I aim to streamline the report view by hiding the 'Inpatient' and 'Outpatient' tables. This approach helps avoid any potential confusion during the dashboard design phase.

Moving forward, my next step involves importing the 'mapping' data file into Power BI. Upon successful import, upon returning to the Modeling tab, a dedicated mapping table is established, forming connections with both the 'All\_Data' and 'Inpatient' tables. To enhance clarity and simplify the structure, I intend to delete the connection specifically between the 'Mapping' table and the 'Inpatient' table. This leaves us with a refined connection solely between the 'Mapping' and 'All\_Data' tables.

# 4. Data Visualization Blueprint.

This is just a high-level idea of visuals might look like and this might change slightly over the period of development process.

## **Summary Page:**

- > Top left
  - Total Waitlist (Current Year vs Previous Year)
- ➤ Middle Section Average waiting list for multiple attributes
  - A pie chart showing a bifurcation between Outpatient, Inpatient and Day cases.
  - A column chart which will showcase the relationship between age profile and time band.
  - A grid which will show average waitlist by specialty.

**Note:** While going through the data I observed outliers in the specialty data. Depending solely on the average could potentially skew our analysis due to these outliers. To mitigate this issue, I decided to incorporate an additional metric. In this instance, I introduced the median as another key metric, supplementing the average wait list for all the charts in the middle section. This ensures that median values are also prominently featured in all three charts.

#### **Bottom Section**

- A line chart of monthly trend analysis on the basis of inpatient, day cases and Outpatient. This will showcase the total waitlist for a particular category.

#### Top Right

- A Date Filter
- A Case Type Filter
- A Specialty Filter

### **Detailed Page:**

- > Right
- A grid which can show the data in the most granular form.
- > Left
  - A few filters to help user to interact with the grid data.

#### 5. Dashboard Layout & Design

Prior to initiating the design process, I prefer to activate the Gridlines and Snap to Grid options. These features enhance the alignment of objects and charts on the canvas, facilitating a smoother design experience.

#### 5.1 Summary Page

- I. Top Left: Total Waitlist (Current Year vs Previous Year)
  - Here, there are two figures presented: the current month's wait list versus the wait list from the same month in the previous year.
  - This comparison is facilitated through a card visual, utilizing the total wait list column for the numerical values. However, the total column cannot be directly employed; hence, a new measure is devised for this specific card. The logic behind this measure involves the summation of the total wait list column, applying a filter based on the archive date column to retrieve the maximum date present in the dataset. This measure is appropriately named "Latest Month Wait List."

```
Latest Month Wait List = CALCULATE(SUM(All_Data[Total]), All_Data[Archive_Date] = MAX(All_Data[Archive_Date])
```

- Similarly, a fresh measure has been crafted to furnish the wait list for the same month in the preceding year, designated as "PY Latest Month Wait List." The formula remains largely unchanged, with a slight modification involving the incorporation of the 'EDATE' function alongside MAX(All\_Data[Archive\_Date]). This adjustment facilitates seamless navigation from the current month to the corresponding month in the previous year, a perfect fit for our specific requirements. The second parameter, denoted as '-12,' is instrumental in achieving this temporal alignment.

```
PY Latest Month Wait List = CALCULATE(SUM(All_Data[Total]), All_Data[Archive_Date] = EDATE(MAX(All_Data[Archive_Date]), -12).
```

# II. Top right: Toggle button for Average and Median

- To establish a toggle button, a placeholder table named 'Calculation Method' is generated.
- This table comprises only two rows: one for 'Average' and the other for 'Median.
- Subsequently, a slicer is crafted, and 'Calculation Method' is integrated as a field. By adjusting the slicer style to 'Tile,' a toggle button is effectively generated.

### III. Measure for Average and Median.

- Initially, two key measures are crafted: 'Average Wait List,' which computes the average of the 'Total' column in the 'All\_Data' table, and 'Median Wait List,' which calculates the median of the 'Total Wait List' column.

```
Average Wait List = Average(All_Data[Total])
Median Wait List = Median(All_Data[Total])
```

- To enable seamless toggling between these two metrics using a button, an additional measure is essential. This measure, named 'Avg/Med Wait List,' employs the SWITCH

function. It dynamically selects between the two measures based on the user's interaction with the button labeled 'Calculation Method.'

**Avg/Med Wait List** = SWITCH(VALUES('Calculation Method'[Calc Method]), "Average", [Average Wait List], "Median", [Median Wait List])

- This measure essentially acts as a switch, allowing users to effortlessly alternate between viewing the average and median wait list values.

## **Explanation:**

- The SWITCH function serves as a pivotal tool for toggling between 'Average' and 'Median.' The segment VALUES('Calculation Method'[Calc Method]) extracts values from the 'Calculation Method' table, which exclusively holds two values: 'Average' and 'Median.'
- When put together with the help of SWITCH function as SWITCH(VALUES('Calculation Method'[Calc Method]), "Average", [Average Wait List], "Median", [Median Wait List])
- This construct dynamically selects the appropriate measure based on the current value. In simpler terms, if the extracted value is 'Average,' the [Average Wait List] measure is employed, and if it is 'Median,' the [Median Wait List] measure takes precedence. This mechanism allows seamless switching between the two metrics as dictated by the user's selection.

#### IV. Middle Section

- On the left, a pie chart showcases the distribution of case types, accompanied by a dynamic measure, Avg/Med Wait List, enabling users to seamlessly toggle between the average and median of the total wait list.
- Positioned at the center, a column chart takes center stage. Here, Avg/Med Wait List is represented on the Y-axis, Time\_Bands on the X-axis, and Age\_Profile is depicted in the Legend section. After loading the chart, there were couple duplicate values that can be seen in the X-axis labels, which are 18+ months and 18 months +.
  - With the power query tab, I have done three changes to the data.
    - ✓ Replaced 18 monmths + to 18+ months in Time Bands column.
    - ✓ Performed trim operation on both Age profile and Time Bands column.
    - ✓ Replaced blanks with 'No Inputs' in both columns.
  - After making these changes, everything looked good However, in the chart, the 'No inputs' field didn't seem important, so I decided to remove it from the chart and legend for a clearer view.
- Positioned on the right, a multi-row card exhibits the top 5 specialty groups. This display is based on 'Specialty\_Name' and 'Avg/Med Wait List.' In the filtering section, the type is adjusted to 'Top N,' with the specified number set to 5. The 'Avg/Med Wait List' is assigned in the value section for accurate representation.

#### V. Bottom Section

- On the left, there's a line chart displaying inpatient and day case patients. It uses 'Archive\_Date' on the horizontal axis, 'Total' on the vertical axis, and 'Case Type' as the legend. The chart is filtered to show only inpatient and day case cases.
- On the right, another line chart shows inpatient and day case patients. Similar to the left chart, it uses 'Archive\_Date' on the X-axis, 'Total' on the Y-axis, and 'Case Type' as the legend. However, this chart is filtered to display only outpatient cases.

# VI. Top Right: Filters

- A slicer with 'Archive Date' as field.
- A second slicer with 'Case Type' as field, with a dropdown style.
- A third slicer with 'Specialty Name' as field, with a dropdown style.

# VII. Title for Toggle: Average and Median

- While utilizing the Average and Median toggle button, it may not be immediately apparent to the user which metrics are currently being shown. To address this, I intend to incorporate a title explicitly indicating whether 'Average' or 'Median' is selected. This will be achieved through the creation of a dynamic measure named 'Dynamic Title.'
- The 'Dynamic Title' measure is formulated as follows

**Dynamic Title** = SWITCH(VALUES('Calculation Method'[Calc Method]), "Average", "Key Indicators – Patient Wait List (Average)", "Median", "Key Indicators – Patient Wait List (Median)")

**Explanation:** The logic mirrors that of the 'Avg/Med Wait List,' but in this case, when the value is 'Average,' the title displays as "Key Indicators – Patient Wait List (Average)," and when the value is 'Median,' it shows as "Key Indicators – Patient Wait List (Median)."

- A card is employed to showcase the title, utilizing the 'Dynamic Title' measure.

The summary page has been crafted, featuring a designated text box labeled 'Summary'.

### 5.2 Detailed Page for Granular Analysis

I made a new page called 'Detail.' I copied over the textbox and filters from the summary page and synchronized them on the detail page.

## I. Left Section:

- All the three filters from the summary page are arranged in a top-down approach to the left.
- I added another filter, similar to the case\_type and Specialty\_Name slicers, but this one uses 'Time Bands' as the field.

# II. Right Section:

- A matrix view, that will give us an option to summarize our data and maybe create some other type of transformation within this view, that will be beneficial for the end user for granular level analysis.
- In this grid, the rows include 'Archive\_Date', 'Specialty\_Name', 'Age\_Profile', and 'Time\_Bands'. 'Case\_Type' is used for columns, and the 'sum of Total' is shown in the values section. Now the detail page is also created.

# **Enhancing Design:**

To give our dashboard a polished and appealing look, I decided to create a theme. I found a design on Adobe Stock that I liked and combined it with our dashboard using Microsoft PowerPoint. To get the color palette from the chosen design, I used another website called color.adobe.com to extract the colors from the image. This way, I've added a stylish theme to our dashboard.

## **Final Changes:**

- First, in the dropdown of Specialty\_Name filter, with the help of formatting, 'select all' option is added.
- Second, I included a number in the pie chart to display the overall average or median for the chosen filter criteria. This was done by selecting a card and adding the Avg/Med Wait List as a field.
- Third, I have changed the decimal places of Avg/Med Wait List measure to zero.

# 6. Interactivity & Navigation

#### **6.1 Interactivity:**

- First, when I changed the date to 2018 in the Date filter, the PY Latest Month Wait List showed Blank. To address this, I added zero to the measure formula of PY Latest Month Wait List. Now, it displays zero whenever there is no data. I applied the same adjustment to the Latest Month Wait List.
- Second, selecting a specific case type might cause one of the two bottom charts to disappear. However, since the bottom charts already showcase daycase, inpatient, and outpatient wait list data, I disabled interactions between the Case Type filter and the bottom two charts.

  Third, when the Specialty Name filter is set to 'Accident & Emergency,' the bottom left chart.
  - Third, when the Specialty\_Name filter is set to 'Accident & Emergency,' the bottom left chart disappears because there is no data to display. Instead of it disappearing, I introduced a new measure called 'NoDataLeft' to show as 'No Data for selected criteria.' I replicated the same approach for the bottom right chart by creating a new measure called 'NoDataRight.'

NoDataLeft = IF(ISBLANK(CALCULATE(SUM(All\_Data[Total]),All\_Data[Case\_Type] <> "Outpatient")), "No data for selected criteria", "")

NoDataLeft = IF(ISBLANK(CALCULATE(SUM(All\_Data[Total]),All\_Data[Case\_Type]="Outpatient")), "No data for selected criteria", "")

# **Explanation:**

- CALCULATE(SUM(All\_Data[Total]),All\_Data[Case\_Type] "Outpatient"). Here calculate function, selects the sum of Total Column only for outpatient entries.
- IF(ISBLANK ()), here, if selected criteria are blank, then it displays "No data for selected criteria" else it displays blank ("").
- All\_Data[Case\_Type] \rightarrow "Outpatient", this selects case types that are not Outpatients. As the left chart displays only Inpatient and day cases.
- All\_Data[Case\_Type]="Outpatient", this selects case types that are Outpatients. As the right chart displays only Outpatient data.

#### **6.2 Navigation:**

- To Navigate through the two pages, I have added a information button at the bottom section and added navigation to the detail page.
- Similarly, a back button is added in the detail section to go back to the summary page.

# 6.3 Custom ToolTip:

- Instead of the regular tooltip, I made a custom tooltip for the bottom charts. This custom tooltip shows the breakdown of specialties for that month when you hover over the charts.
- I made a new page called 'Drilldown,' but I've hidden it because I only need it for the tooltip feature.
- On this hidden page, I created a card displaying the total number of wait lists for that specific month.
- Using the Specialty\_Group from the Mapping\_Specialty table and selecting the 'Total' column from the All Data table, I created a stacked bar chart.
- This 'Drilldown' page is added to the tooltip settings for both charts.

## 7. Testing

During the testing phase, I focused on verifying the basic functionality of the Power BI dashboard. I specifically tested whether all buttons were working as intended, if all the data was being displayed correctly, and assessed the responsiveness of the interactions. This testing aimed to ensure that the fundamental components of the dashboard were operational and user-friendly.

## 8. Sharing

I've successfully published my dashboard project on Power BI Services. Since my account isn't a premium one, I couldn't publish it as a PBI app. However, I can share it through a link or export it as an HTML link. I opted for the HTML link and integrated it into my portfolio website for easy access.

#### 9. Maintenance and Refresh.

To update the data in the Power BI dashboard, I simply need to refresh the data files within the designated folder. Using the Power BI desktop app, I click the refresh button, initiating a connection to the folder for the latest data. Once updated, clicking the publish button ensures the changes are reflected not only in the Power BI desktop app but also on Power BI Services.

### References

- a. Dashboard Images Browse 426,103 Stock Photos, Vectors, and Video | Adobe Stock
- b. Color theme from image | Adobe Color