

**NGEE ANN**  
POLYTECHNIC**School of InfoComm Technology**

## **Distributed Data Pipelines**

Diploma in Data Science (DS)  
October 2022 Semester

### **INDIVIDUAL ASSIGNMENT 2**

(40% of Distributed Data Pipelines Module)

#### **Deadline for Submission:**

**10<sup>th</sup> Feb 2023 (Friday), 2359 Hours**

Student Name	:	Ryan Liam
Student Number	:	S10222131E

#### **Penalty for late submission:**

10% of the marks will be deducted every day after the deadline.

**NO** submission will be accepted after **17<sup>th</sup> Feb 2023, 23:59**.

## Contents

1. Problem Statement Formulation .....	3
2. Preparation .....	4
3. API Call.....	5
4. Data Display .....	7
5. Enhancement (Optional) .....	9
5.1 Ideal Frontend Display – Adobe XD .....	9
5.2 Working Example – Alteryx Analytics App .....	12
6. Summary .....	13
6.1 Summarize Findings.....	13
6.2 Proof of Completion of Alteryx Foundation Micro-Credential and “Getting Started Learning Path” .....	14
7. Reflection.....	15
7.1 Possible Improvements .....	15
7.2 Reflection on skills learned.....	15
8. References .....	16

## 1. Problem Statement Formulation

In Singapore, many rely on public buses to commute to work, school, or other daily activities due to their accessibility and affordability. According to a survey conducted by the Public Transport Council (PTC), which seeks to understand commuters' needs and identify areas for improvement, it was found that customer satisfaction faced a dip in 2020 (Correspondent, 2022). More importantly, one of the main factors for this dip was that more commuters were unsatisfied with the waiting time of buses and their reliability. Furthermore, not knowing the wait times for buses could also result in commuters facing significant inconvenience and frustration, as it can lead to missed appointments, prolonged commutes, and wasted time waiting at bus stops which can reduce overall satisfaction with the public transport system.

To address this issue of not knowing the waiting time, we will be making use of a front-end and an analytics automation platform, Alteryx, to provide commuters with real-time information on when a bus will arrive at their selected bus stop. This will enable them to better plan their trips and avoid unnecessary wait times allowing their daily commutes to be smoother and more efficient. As for the data we will be using, we will be making use of one of the datasets about the real-time bus arrival information at designated bus stops provided by LTA on DataMall. We will then be using Alteryx Designer to load the real-time stream data and build a simple data pipeline to provide the real-time bus arrival time at any given bus stop.

For us to make use of the data in our workflow in Alteryx Designer, we will be making use of the Text Input tool which will contain the base URL of the API endpoint, my given Account key, and the selected bus stop code.

## 2. Preparation

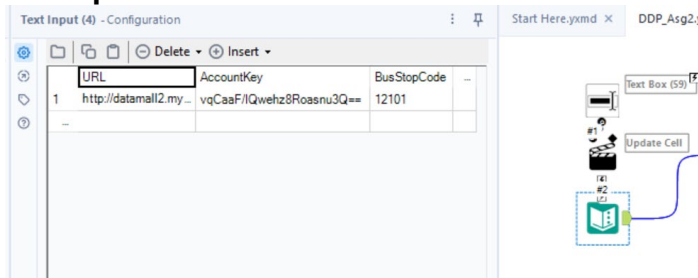


Fig 1: Input Tool and Text Box to update the Bus Stop Code for Analytics Application

It is necessary to have the base URL of the API and an Account key to access data via API (Application Programming Interface). The API URL acts as the endpoint for accessing a specific data source, and the Account key is used to authenticate access to that data. Thus, we will be including the API URL and Account key in the Text Input tool. Since we want the user to be able to input their bus stop code through a front-end website or Alteryx Analytic App, we will also need to add the bus stop code in the Text Input tool. The bus stop code in the Text Input tool will be updated once the user inputs their selected bus stop code.

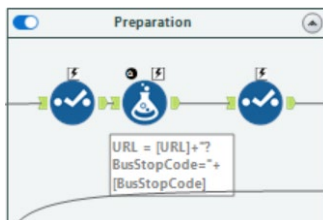


Fig 2: Preparing the new URL for API Calls

The Select tool is used as a preliminary step in data preparation to clean and prepare the data for further processing. To prepare the new URL for the API Calls, we will use the Select tool, a data manipulation tool that can include, exclude, and reorder the columns that pass through the workflow. By excluding irrelevant or unnecessary columns, it can help improve performance. The select tool can also be used to modify the type and size of the data, rename columns, add a description, and more. We will be making use of the select tool to ensure the API URL, Account Key, and Bus Stop Code are of a String data type. To use the Select tool, we first need to connect it to the Text Input tool, which contains the base URL, Account Key, and Bus Stop Code.

After we have prepared the data with the Select tool, we can move on to using the Formula tool. The formula tool can be used to apply conditional statements, convert numbers and strings, format dates, extract file paths, apply financial algorithms or mathematical calculations, find the minimum and maximum values, analyse spatial data, cleanse string data, or perform validation tests on data. The formula tool will be used to create the API Call URL where we are able to retrieve the real-time bus information of a particular bus stop. To do that, we will combine the base URL of the API with the “?” symbol to specify the parameters being passed to the API to allow it to receive and process the specified parameters. The parameter used to specify additional information for the API to return desired data is BusStopCode. The base URL for the API is “http://datamall2.mytransport.sg/ltaodataservice/BusArrivalv2”, and to get the bus information for a specific bus stop code, we must make an API call in the following format “http://datamall2.mytransport.sg/ltaodataservice/BusArrivalv2?BusStopCode=[BusStopCode]”.

We will be making use of the select tool again to exclude irrelevant or unnecessary columns to improve the workflow performance. Since we already have the URL that we will be using for the API call to get the real-time bus information of the selected bus stop code, we no longer need the bus stop code column. Using the select tool, we exclude the bus stop code column. After which, we are ready to make the API call.

### 3. API Call

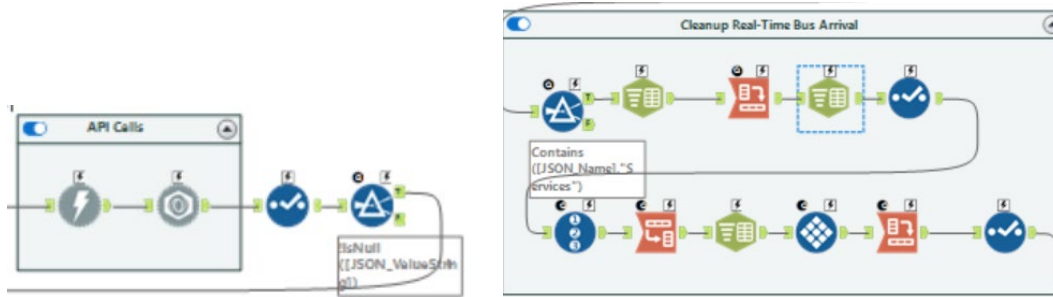


Fig 3: Download the output from API calls and transforming the data into appropriate format

For this portion, we made use of the Download tool to retrieve data from the URL we created. The data will appear in JSON format in a singular cell, which is why we will also be using the JSON Parse tool to turn it into a table schema. Now that we have our data, we no longer need the URL, Account key, and the Download Headers created using the Download tool. Therefore, I use the Select tool to exclude these columns, leaving the JSON name and value columns.

Record	JSON_Name	JSON_ValueString
1	Services.0.ServiceNo	151
2	Services.0.Operator	SBST
3	Services.0.NextBus.OriginCode	16009
4	Services.0.NextBus.DestinationCode	64009
5	Services.0.NextBus.EstimatedArrival	2023-02-05T21:24:37+08:00
6	Services.0.NextBus.Latitude	0
7	Services.0.NextBus.Longitude	0

Fig 4: After filtering for only rows with "Services" in JSON\_Name

To ensure there are no null values, I made use of the filter tool to select rows of data that do not have null values in the JSON value column. Rows of data that meet the condition are output to the True anchor, which we will be using. Since the data we require is only under the "Services" key, we will have to make use of the filter tool again to select only the rows of data that contain Services in the JSON name column.

Record	ServiceNo	EstimatedArrival	Feature	Load	Type	Latitude	Longitude
1	184	2023-02-05T22:30:49+08:00,2023-02-05T22:46:3...	WAB,WAB,WAB	SEA,SEA,SEA	SD,SD,SD	1.3180818333333333,1.3398765,1.3775545	103.77129033333333,103.779434,103.771666833...
2	61	2023-02-05T22:42:20+08:00,2023-02-05T22:52:3...	WAB,WAB,WAB	SEA,SEA,SEA	SD,SD,SD	1.3055088333333333,1.2747121666666666,1.267...	103.79474283333333,103.80195083333333,103.8...
3	151	2023-02-05T22:29:11+08:00,2023-02-05T22:42:3...	WAB,WAB,WAB	SEA,SEA,SEA	SD,SD,SD	1.3236628333333333,0,0	103.77205016666667,0,0
4	75	2023-02-05T22:34:22+08:00,2023-02-05T22:44:1...	WAB,WAB,WAB	SEA,SEA,SEA	SD,SD,SD	1.319288,1.3066426666666666,0	103.78231033333333,103.818889,0
5	154	2023-02-05T22:29:33+08:00,2023-02-05T22:41:4...	WAB,WAB,WAB	SEA,SEA,SEA	SD,SD,SD	1.3237088333333333,1.3231995,1.339310666666...	103.77209233333333,103.74797083333333,103.7...
6	74	2023-02-05T22:33:01+08:00,2023-02-05T22:40:0...	WAB,WAB,WAB	SEA,SEA,SEA	SD,DD,DD	1.3120818333333333,1.3001985,0	103.77256216666666,103.784593,0
7	52	2023-02-05T22:30:46+08:00,2023-02-05T22:46:4...	WAB,WAB,WAB	SEA,SEA,SEA	SD,SD,DD	1.3190136666666666,1.3351398333333333,0	103.7710375,103.74786116666667,0

Fig 5: Output after Cross Tab

We now want to extract the nested keys, such as "ServiceNo" from "Services.0.ServiceNo", to make our output more readable. We make use of the Text to Columns tool to split the JSON name into columns by "." delimiter. Then, we want to group our data by the unique identifier that represents each bus service number by using the Cross Tab tool. The values for the new columns will be from the JSON name columns and will be aggregated by concatenation. The headers for these values are taken from the column that stores them. As seen in Figure 5, the "\_Null\_" column shows values of the Service number and Operator, which should be in two different columns. Thus, I used the Text to Columns tool to split the "\_Null\_" into two columns by the comma delimiter. Then, we used the Select Tool to exclude "\_Null\_" and renamed the newly created columns to ServiceNo and Operator, respectively. The Select Tool is used to exclude the Operator, Visit Number, Feature, Origin Code, and Destination Code columns as they are not useful to the user. With the implementation of wheelchair accessibility on all public buses by 1 December 2020, the feature column is no longer needed (All Public Buses Are Wheelchair Accessible from 1 December 2020 | Land Transport Guru, 2020).

Record	ServiceNo	EstimatedArrival	Load	Type	Longitude	Latitude
1	184	2023-02-06T15:17:50+08:00	SEA	DD	103.77078	1.3217075
2	154	2023-02-06T15:55:10+08:00	SEA	DD	103.71111466666666	1.3411805
3	52	2023-02-06T15:42:57+08:00	SEA	DD	103.73880833333334	1.335776
4	184	2023-02-06T15:22:47+08:00	SDA	DD	103.76543516666666	1.315259
5	154	2023-02-06T15:19:21+08:00	SEA	DD	103.771288	1.3178095
6	75	2023-02-06T15:44:09+08:00	SEA	SD	103.83333216666666	1.28979
7	75	2023-02-06T15:27:25+08:00	SEA	SD	103.79617483333334	1.3121626666666666

Fig 6: Ideal Output

After the Cross Tab, our output for the Estimated Arrival, Load, Type, Longitude, and Latitude columns consists of information on the next three buses in each cell. The display is **extremely cluttered** and may be confusing to the users. Instead, we want to present the information in a clear and organized way to be more user-friendly by showing the arrival time for each bus in a separate row. Doing so allows for easy comparison of the arrival times.

Record	RecordID	ServiceNo	Name	Value
1	1	184	EstimatedArrival	2023-02-06T15:17:50+08:00,2023-02-06T15:22:4...
2	1	184	Load	SEA,SDA,SDA
3	1	184	Type	DD,DD,BD
4	1	184	Latitude	1.3217075,1.315259,1.32636
5	1	184	Longitude	103.77078,103.76543516666666,103.7756913333...
6	2	61	EstimatedArrival	2023-02-06T15:33:54+08:00,2023-02-06T15:46:3...
7	2	61	Load	SEA,SEA,

Fig 7: Output after Transposing

To get our ideal output, we first need to give each row a unique ID using the Record ID tool. Doing this will be very useful later when we are working with the Tile tool. Next, we used the transpose tool with the key columns RecordID and ServiceNo and the other columns being the data columns. Now that we have all the cluttered values in one column, we can clean it much easier. Using the Text to Columns tool, we separate the cluttered values into rows by the comma delimiter.

Record	RecordID	ServiceNo	Name	Value	Tile_Num	Tile_SequenceNum
1	4	75	EstimatedArrival	2023-02-06T10:59:22+08:00	-3	1
2	4	75	EstimatedArrival	2023-02-06T11:23:43+08:00	-3	2
3	4	75	EstimatedArrival	2023-02-06T11:34:57+08:00	-3	3
4	5	154	Longitude	103.77165733333334	-3	1
5	5	154	Longitude	103.76285466666667	-3	2
6	5	154	Longitude	103.72434133333333	-3	3

Fig 8: Output after Tile tool

Next, we are going to use the Tile tool to assign a sequence number based on the Record ID and the Name column. In the Tile tool, we configure the Tile method as Smart Tile, the Numeric column as RecordID, and Group by the RecordID and Name column. Doing this allows us to uniquely identify which values belong to which bus for each service number. Then, we want to group our data by the RecordID, ServiceNo and Tile\_SequenceNum using the Cross Tab tool. The new column headers are based on the Name column and its values will be from the Value column.

Record	RecordID	ServiceNo	Tile_SequenceNum	EstimatedArrival	Longitude	Latitude	Load	Type
1	1	184	1	2023-02-06T15:17:50+08:00	103.77078	1.3217075	SEA	DD
2	5	154	3	2023-02-06T15:55:10+08:00	103.71111466666666	1.3411805	SEA	DD
3	7	52	3	2023-02-06T15:42:57+08:00	103.73880833333334	1.335776	SEA	DD
4	1	184	2	2023-02-06T15:22:47+08:00	103.76543516666666	1.315259	SDA	DD
5	5	154	1	2023-02-06T15:19:21+08:00	103.771288	1.3178095	SEA	DD
6	4	75	3	2023-02-06T15:44:09+08:00	103.83333216666666	1.28979	SEA	SD
7	4	75	2	2023-02-06T15:27:25+08:00	103.79617483333334	1.3121626666666666	SEA	SD

Fig 9: Output after Cross Tab

Since we no longer require the RecordID and Tile\_SequenceNum column, we exclude them using the Select tool.



#### 4. Data Display

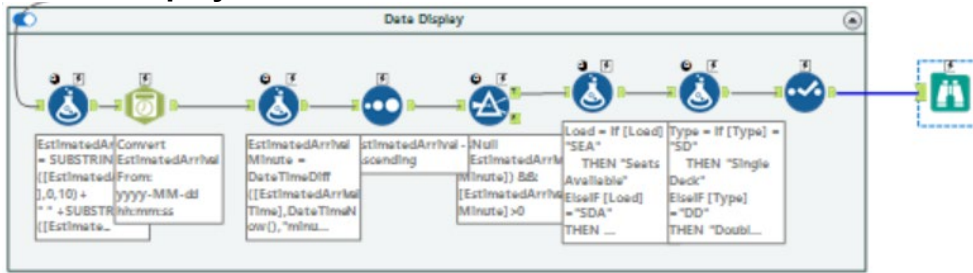


Fig 10: Filter and cleanse the data to display the bus service number and estimate arrival time

After transforming the data into a cleaner format, we can now start filtering and cleaning the data to display the bus service number, estimate arrival time, and other important details about the bus. First, we must clean the EstimatedArrival column using the Formula tool, where the substring function is used to get the values in the yyyy-MM-dd hh:mm:ss format. We then use the DateTime tool to convert the EstimatedArrival column into a new column called EstimatedArrivalTime of the DateTime data type. Then we used the Formula tool with the DateTimeDiff and DateTimeNow functions to get how long the bus will arrive at a particular bus stop in minutes. The value created will be stored under the Estimated Arrival Minute column. I did this to make it easier for the user to see how long it will take for their bus to come. Furthermore, Singapore is a small city-state, and it is unusual for the bus to arrive in an hour's time, so I put it in minutes.

Record	ServiceNo	EstimatedArrival	Load	Feature	Type	Longitude	Latitude	EstimatedArrivalTime	EstimatedArrivalMinute
1	74	2023-02-06 14:31:31	SEA	WAB	DD	103.7777255	1.3320910000000001	2023-02-06 14:31:31	-1
2	151	2023-02-06 14:33:38	SEA	WAB	SD	103.772352	1.3241435	2023-02-06 14:33:38	0
3	154	2023-02-06 14:34:02	SEA	WAB	SD	103.77252833333334	1.3240643333333333	2023-02-06 14:34:02	0
4	184	2023-02-06 14:34:36	SEA	WAB	DD	103.77215216666667	1.3236315	2023-02-06 14:34:36	1
5	75	2023-02-06 14:34:43	SEA	WAB	SD	103.77207016666667	1.323684	2023-02-06 14:34:43	1
6	52	2023-02-06 14:36:27	SEA	WAB	DD	103.7712505	1.3180923333333334	2023-02-06 14:36:27	2
7	61	2023-02-06 14:36:44	SEA	WAB	SD	103.77165416666666	1.319155	2023-02-06 14:36:44	3
8	74	2023-02-06 14:38:09	SEA	WAB	DD	103.77238766666666	1.3120773333333333	2023-02-06 14:38:09	4
9	154	2023-02-06 14:40:25	SEA	WAB	SD	103.7676335	1.3123466666666666	2023-02-06 14:40:25	6
10	74	2023-02-06 14:44:59	SEA	WAB	DD	103.78069183333334	1.3055331666666667	2023-02-06 14:44:59	11
11	151	2023-02-06 14:47:48	SEA	WAB	SD	0	0	2023-02-06 14:47:48	14
12	61	2023-02-06 14:49:01	SEA	WAB	SD	103.795124	1.305158	2023-02-06 14:49:01	15
13	75	2023-02-06 14:50:11	SEA	WAB	SD	103.80926433333333	1.3086	2023-02-06 14:50:11	16

Fig 11: Output after Sort Tool

Now that we have the arrival time of the bus in minutes, we can sort it by ascending order. With this, our users can quickly see which bus will arrive at the stop in the shortest amount of time. If they can take more than one bus with different service numbers, they can also use it to plan which bus they should take. Either by determining which is faster or whether they can catch it in time. Doing so allows them to choose the bus that is most convenient for them, improving the customer experience overall.

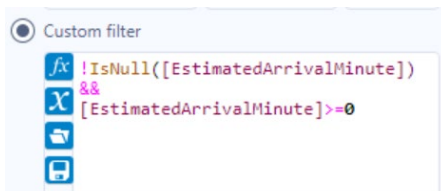


Fig 12: Custom filter code in Filter tool

Next, we will be using the Filter tool to ensure there are no missing values and that the Estimate arrival time of the bus in minutes is more than 0. Rows of data that meet the condition are output to the True anchor, which we will be using. We are doing this because the API may sometimes provide outdated information on the bus's estimated arrival time, leading to a negative value for the arrival time of the bus in minutes. One reason that this happens could be that the API may still be providing information on buses that have already missed their estimated arrival time or is currently at that bus stop.

Record	ServiceNo	EstimatedArrival	Load	Feature	Type	Longitude	Latitude	EstimatedArrivalTime	EstimatedArrivalMinute
1	151	2023-02-06 14:49:58	SEA	WAB	SD	103.77574516666667	1.3269068333333334	2023-02-06 14:49:58	0
2	154	2023-02-06 14:52:19	SEA	WAB	DD	103.770985	1.3219593333333333	2023-02-06 14:52:19	2
3	52	2023-02-06 14:55:31	SEA	WAB	DD	103.7698995	1.3120171666666667	2023-02-06 14:55:31	6
4	184	2023-02-06 14:56:00	SEA	WAB	DD	103.768063	1.3125943333333332	2023-02-06 14:56:00	6
5	61	2023-02-06 14:56:09	SEA	WAB	SD	103.7857535	1.3180151666666666	2023-02-06 14:56:09	6
6	74	2023-02-06 15:00:21	SEA	WAB	DD	103.78291533333334	1.3085388333333334	2023-02-06 15:00:21	10
7	151	2023-02-06 15:03:25	SEA	WAB	DD	103.77260966666667	1.2970388333333334	2023-02-06 15:03:25	13

Fig 13: Output after Filter Tool

Even though we are only tasked with obtaining the bus service number and its estimated arrival time, I believe that the Load, Type, Longitude, and Latitude columns are also useful. By having visibility into the bus capacity and its type, whether double-decker, single-decker, or bendy, users can make informed decisions on whether to board that bus based on their comfort preferences. In my ideal bus application or website, users will be able to track where their next bus is through a map or other visualizations with the use of the longitude and latitude. Doing so allows my users to estimate when the bus will arrive based on the bus's exact location.

Output Column	Data Preview	Output Column	Data Preview
Load	Limited Standing	Type	Double Deck
<pre> If [Load] = "SEA"   THEN "Seats Available" ElseIf [Load]="SDA"   THEN "Standing Available" ElseIf [Load]="LSD"   THEN "Limited Standing" ELSE   [Load] ENDIF </pre>		<pre> If [Type] = "SD"   THEN "Single Deck" ElseIf [Type]="DD"   THEN "Double Deck" ElseIf [Type]="BD"   THEN "Bendy" ELSE   [Type] ENDIF </pre>	
Data type: V_WString	Size: 2048	Data type: V_WString	Size: 2048

Fig 14: Formula Tool code to change values in Load and Type to its Full form

To make the Load and Type columns more readable to the user, I used the Formula tool to show the expanded form of the values in these columns. However, if there is not enough space in the Front-end to display this information, there is no need for this step. Finally, we used the Select tool to exclude EstimatedArrival and EstimatedArrivalTime since we already have the EstimatedArrivalMinute column, which shows the arrival time of the bus in minutes. The Select tool will also be used to reorder the table.

Record	ServiceNo	EstimatedArrivalMinute	Load	Type	Longitude	Latitude
1	74	0	Limited Standing	Double Deck	103.77770233333334	1.3324065
2	184	3	Seats Available	Double Deck	103.77078	1.3217075
3	75	4	Seats Available	Single Deck	103.77239366666667	1.3192763333333333
4	74	5	Seats Available	Double Deck	103.771245	1.3179238333333334
5	154	5	Seats Available	Double Deck	103.771288	1.3178095
6	151	6	Seats Available	Single Deck	103.7721055	1.3120206666666667
7	184	8	Standing Available	Double Deck	103.76543516666666	1.315259

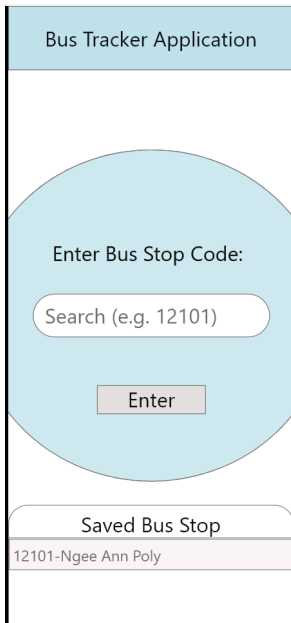
Fig 15: Final Output

To display the final output, we use the Browse tool, which only has one anchor. The Browse tool is a very useful tool as not only does it show the output as a table, but it also allows us to visualize each column in a count plot or histogram. Overall, the Browse tool can be used to view information on data type, number of records, data quality, and a variety of statistics.



## 5. Enhancement (Optional)

### 5.1 Ideal Frontend Display – Adobe XD



Bus Tracker Application

Enter Bus Stop Code:

Search (e.g. 12101)

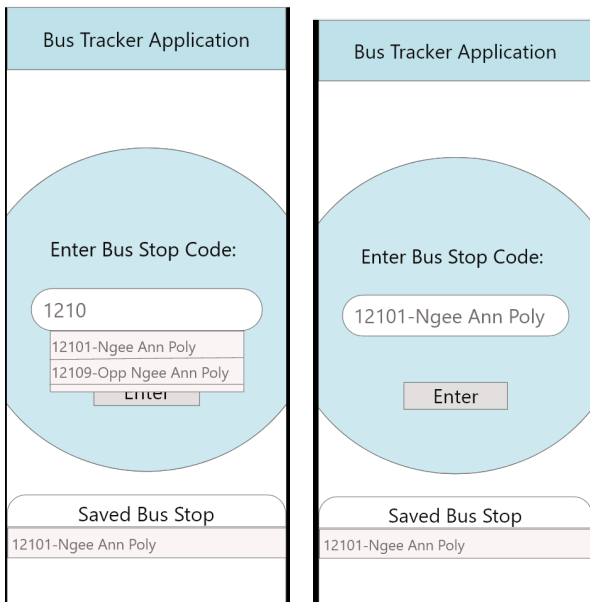
Enter

Saved Bus Stop

12101-Ngee Ann Poly

Fig 16: Bus application home page

To showcase my desired display, I will be using Adobe XD. The home page of the bus application shows the name of the Bus Tracker application at the top, and in the middle, there is a search bar. The user can get the desired bus schedule of the selected bus stop by inputting the bus stop code in the search bar. Once they have typed or selected their desired bus stop code, they can click the enter button to move on to the Bus Stop information page. At the bottom of the page is a section where users can see their saved bus stops. Clicking on one of their saved bus stops will lead them directly to the bus information page of the selected bus stop. This function allows my users to access the bus information they want faster.



Bus Tracker Application

Enter Bus Stop Code:

1210

12101-Ngee Ann Poly

12109-Opp Ngee Ann Poly

Enter

Saved Bus Stop

12101-Ngee Ann Poly

Fig 17: Bus application search function

As the user is typing out their bus stop code in the search bar, it will also show the possible options they can choose. This function is to allow my user to have an easier and faster experience searching for their desired bus stop.

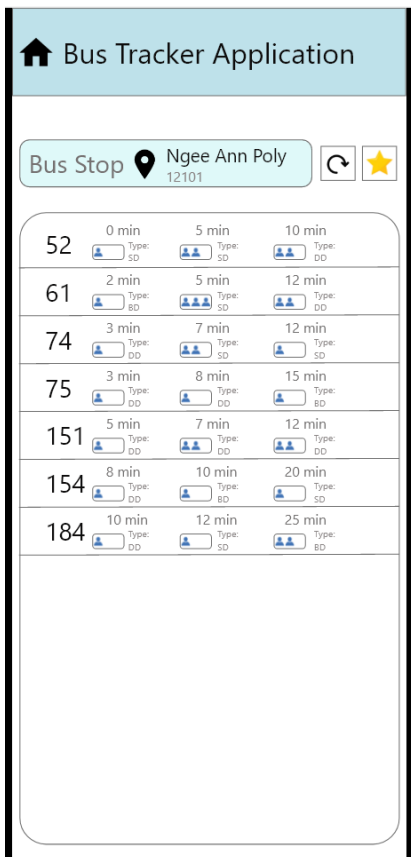


Fig 18: Bus application Bus Stop information page

After selecting their desired bus stop, it will lead the user to this page which shows information about the bus of their chosen bus stop. At the top of the page, the user can click on the home icon to get back to the home page. Next to the container that contains what bus stop we are looking at, there is a refresh button and a save button. In my desired Bus Tracker Application, clicking on the refresh button will re-run the workflow to show the updated bus information of their chosen bus stop. When the user clicks on the save button, it will appear on the home page under the Saved Bus Stop section. The section below these buttons shows all the buses that go to the selected bus stop and their respective waiting times for the next three buses. Below the waiting time is a container with human logos to show how packed the bus is. The maximum is three showing that there is limited standing space. When there are two, it means that there is standing space available. Lastly, when there is one, there are seats available. Next to the container shows what type of bus it is, which could be Single Decker, Double Decker, or Bendy. For the Bus Application, the type of the bus will be in acronym format as there is limited space. When the user clicks on any bus service number, it will bring them to the Bus Tracking page.

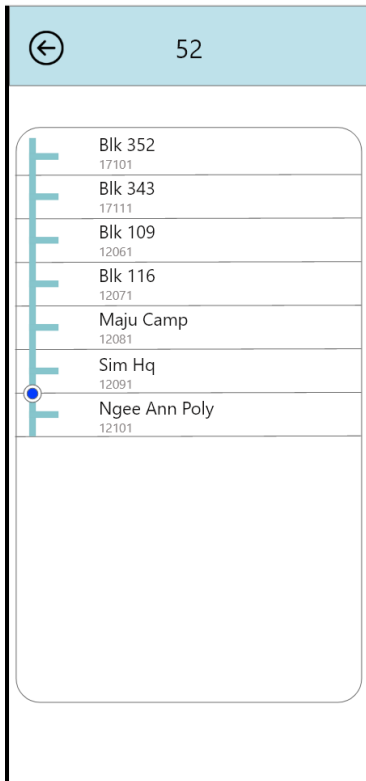


Fig 19: Bus application Bus Tracking page

In our desired Bus application, we will use the latitude and longitude to show whether the bus is at a particular bus stop or is on its way to the user's chosen bus stop. At the top of the page, it shows what bus service number we are looking at, and there is also a button that will bring the user back to the Bus Stop information page.

**If you do not have Adobe XD, you can access the prototype with this link:**

<https://xd.adobe.com/view/5f705f28-a995-443c-b547-0a9a9a9c87e2-0a23/?fullscreen>

## 5.2 Working Example – Alteryx Analytics App

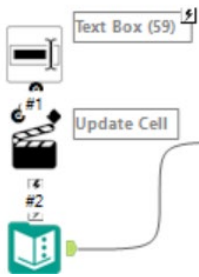


Fig 20: Text Box tool to update the Bus Stop Code in Text Input Tool in Workflow



Fig 21: Wand button to run the Analytics App

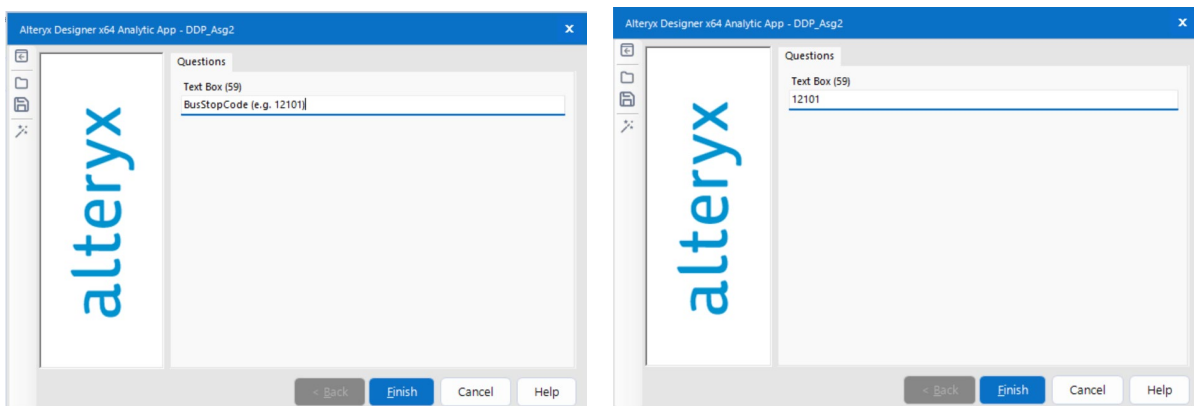
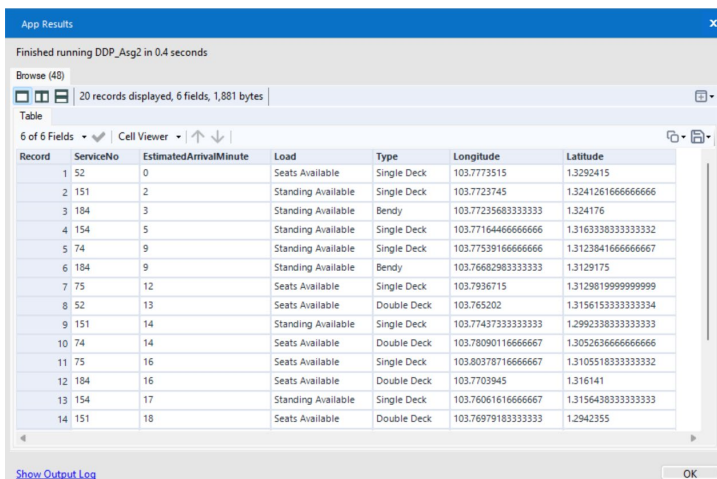


Fig 22: Alteryx Analytics App UI

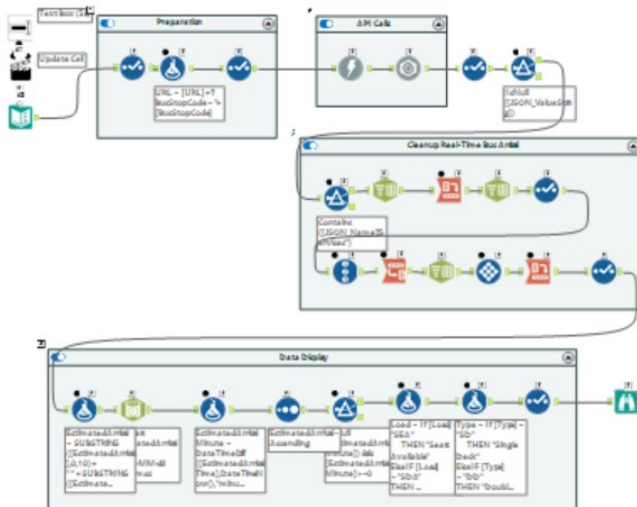


The screenshot shows the 'App Results' window in Alteryx. It displays a table with 20 records, 6 fields, and 1,881 bytes. The table is titled 'Table' and shows the following data:

Record	ServiceNo	EstimatedArrivalMinute	Load	Type	Longitude	Latitude
1	52	0	Seats Available	Single Deck	103.7773515	1.3292415
2	151	2	Standing Available	Single Deck	103.7723745	1.3241261666666666
3	184	3	Standing Available	Bendy	103.77235683333333	1.324176
4	154	5	Standing Available	Single Deck	103.77164466666666	1.3163338333333332
5	74	9	Standing Available	Single Deck	103.77539166666666	1.3123841666666667
6	184	9	Standing Available	Bendy	103.76682983333333	1.3129175
7	75	12	Seats Available	Single Deck	103.7936715	1.3129819999999999
8	52	13	Seats Available	Double Deck	103.765202	1.3156153333333334
9	151	14	Standing Available	Single Deck	103.77437333333333	1.2992338333333333
10	74	14	Seats Available	Double Deck	103.78090116666667	1.3052636666666666
11	75	16	Seats Available	Single Deck	103.80378716666667	1.3105518333333332
12	184	16	Seats Available	Double Deck	103.7703945	1.316141
13	154	17	Standing Available	Single Deck	103.76061616666667	1.3156438333333333
14	151	18	Seats Available	Double Deck	103.76979183333333	1.2942355

Fig 23: Alteryx Analytics App Result

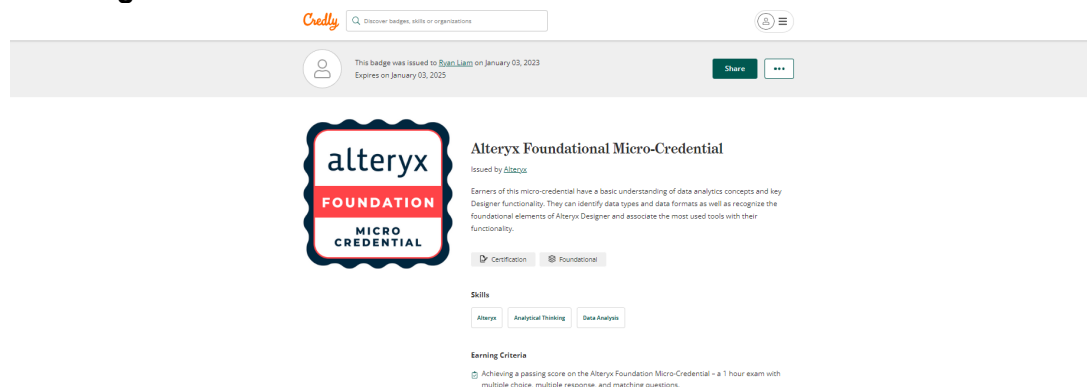
In Alteryx, our created workflow can be integrated into a user interface by using interface tools like Text Box. When the Wand button is clicked, as seen in Fig 21, it will run the workflow through the Analytics App. As shown in Fig 22, a user interface will appear with the text box where the user can type out their chosen Bus Stop Code and click Finish. In the workflow, the bus stop code entered in the text box will be updated in the bus stop code column in the Text Input Tool. The user is then brought to the App Results page, which shows the output of the workflow in Tabular format, as shown in Figure 23.



Under the Enhancement section, I used Adobe XD to show how the Front-end will look. I have also used the Alteryx Analytics App to display a working example of running the workflow through a User Interface.



## 6.2 Proof of Completion of Alteryx Foundation Micro-Credential and “Getting Started Learning Path”



**Alteryx Foundational Micro-Credential**  
 Issued by [Alteryx](#)

Earners of this micro-credential have a basic understanding of data analytics concepts and key Designer functionality. They can identify data types and data formats as well as recognize the foundational elements of Alteryx Designer and associate the most used tools with their functionality.

**Skills**

- Alteryx
- Analytical Thinking
- Data Analysis

**Earning Criteria**

Achieving a passing score on the Alteryx Foundation Micro-Credential - a 1 hour exam with multiple choice, multiple response, and matching questions.

**Alteryx Certification Exam Results**

Results for: Ryan Liam

Test: **Alteryx** Foundation Micro-Credential

Percentage: 100%

Duration: 00:43:34

Date started: Tue 3 Jan 2023 05:05

Date finished: Tue 3 Jan 2023 05:49

**Feedback:**

**Congratulations** - you passed the **Alteryx** Foundation Micro-Credential exam!

Next, accept your digital badge! You will be receiving an email from [admin@credly.com](mailto:admin@credly.com) **within 48 hours**. Once you receive the email, click on the link and create your Credly account. Accept your badge and start sharing! Learn more about digital badges [here](#).

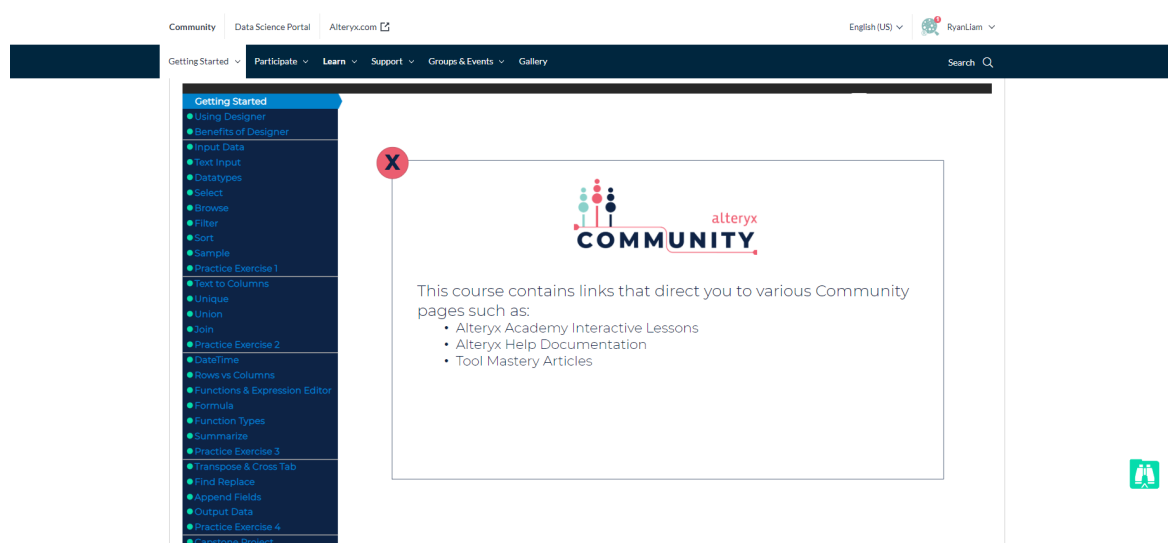
If you haven't seen your email from Credly, be sure it wasn't caught in your junk folder.

Use your **Alteryx** powers to help others by answering questions on the Discussion Boards or by joining a User Group!

Now that you have conquered the Foundation micro-credential, start your preparations for Core exam with the [Core Certification Exam Prep Guide](#), the [Getting Started Learning Path](#) and the [Core Certification Prep Videos Parts I - IV](#).

Category	Percentage
1.1 Data Formats	100%
1.2 File Extensions	100%
1.3 Data Characteristics	100%
1.4 Data Fields	100%
1.5 Data Types	100%
1.6 Metadata	100%
1.7 Statistical Measures	100%
2.1 UI Elements and Functions	100%
2.2 Help Resources	100%
2.3 Workflow Components	100%
2.4 Workflow Management	100%
2.5 Designer File Types	100%
2.6 Designer Tools	100%
2.7 UI Familiarity	100%

Fig 25: Proof of completion certificate document of the Alteryx Foundation Micro-Credential



Community | Data Science Portal | [Alteryx.com](#)

English (US) | [RyanLiam](#)

Getting Started | Participate | Learn | Support | Groups & Events | Gallery

**Getting Started**

- Using Designer
- Benefits of Designer
- Input Data
- Text Input
- Datatypes
- Select
- Browse
- Filter
- Sort
- Sample
- Practice Exercise 1
- Text to Columns
- Unique
- Union
- Practice Exercise 2
- DateTime
- Rows vs Columns
- Functions & Expression Editor
- Formula
- Function Types
- Summarize
- Practice Exercise 3
- Transpose & Cross Tab
- Find Replace
- Append Fields
- Output Data
- Practice Exercise 4
- Capstone Project

**alteryx COMMUNITY**

This course contains links that direct you to various Community pages such as:

- Alteryx Academy Interactive Lessons
- Alteryx Help Documentation
- Tool Mastery Articles

Fig 26: Proof of completion of all of the "Getting Started Learning Path".

## 7. Reflection

### 7.1 Possible Improvements

To optimize the current workflow, we can utilize the Select tool more efficiently by removing fields that are not necessary as early as possible in the data stream. Doing so can help improve the performance of the workflow by reducing the amount of data being processed.

Another way we can improve workflow efficiency is by optimizing record size. By reducing record size, we can increase data loading speed. The record size is determined by field size, which tends to have a default size that is larger than it should be. String fields with a big size can be costly, slowing down the workflow. Therefore, we can optimize the workflow by setting the field to the smallest possible size and most efficient field type. To do so, we can use the Auto Field tool to scan the data and find the smallest possible string type and length which will contain the data. This reduces resource consumption and can result in significant speed improvements for the workflow.

As mentioned in the Data Display section, I turned the Load and Type values into an expanded form to make it more readable to the user. When we have finalized a working frontend display where we can run the Alteryx workflow, we can determine if there is enough space or a need to display this information. If there is not enough space, we can remove the Formula tools used to improve workflow performance.

To execute workflows on a Front-end and display the results on it, we require an Alteryx Server license. If we want to output our data in Power BI, we can make use of the Power BI output tool. However, we require admin approval to do so.

### 7.2 Reflection on skills learned

After completing this assignment, I have experienced the benefits of Alteryx especially in the data pipelining process. Alteryx provides us a variety of tools to perform different stages of the data pipelining process. This includes data ingestion, data preparation, data transformation and data analysis. Furthermore, Alteryx provides us tools to perform machine learning as well.

In Alteryx, we can get data using the Data input tool by connecting to a file or database. We can also retrieve data from APIs by using the Text input tool. Furthermore, Alteryx allows us to easily use data from APIs by providing tools like Download and JSON parse. As for data preparation, we have tools such as Select, Data Cleansing, and Filter. This allows us to get the necessary data, convert values to the correct datatypes and remove or replace null values. For data transformation, we have tools such as Formula, Text to columns, Transpose, and Cross Tab, which allow us to manipulate and shape our data in a way that is useful for analysis. We can make use of DateTime values by using the DateTime tool. If we want to sort the data by a column, we could use the Sort Tool. Lastly, we can display our data in a Tabular format using the Browse tool. It can also be used for data analysis as it allows us to visualize each of the features and view information on a variety of statistics.

As for the skills that could have been learned better concerning the module learning objectives stated in the week one slide, we have not touched on Microsoft Azure and used it in any practical. I feel that it could be interesting to learn its applications and also use them in this assignment.

Overall, Alteryx provides us tools to retrieve, prepare, and clean data simply and transparently with no code at all. Sometimes using code for data pipelines could make it difficult to track what was done with the data. By using Alteryx, which shows the workflow visually, we can easily track what happened with the data. Therefore, I believe that Alteryx is a very useful tool for Data Science.

## 8. References

Correspondent, K. Y. (2022, February 10). Dissatisfaction with S'pore public transport rises: Survey. *The Straits Times*. <https://www.straitstimes.com/singapore/dissatisfaction-with-spore-public-transport-rises-in-2021-survey>

All public buses are Wheelchair Accessible from 1 December 2020 | Land Transport Guru. (2020, December 12). <https://landtransportguru.net/all-public-buses-wheelchair-accessible/>