

**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:**

**10th Feb 2023 (Friday), 2359 Hours**

|  |  |  |
| --- | --- | --- |
| Student Name | : | Brian Low De Quan |
| Student Number | : | S10228107D |

**Penalty for late submission:**

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

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

# Problem Statement Formulation

To encourage collaboration and the co-creation of cutting-edge and inclusive transportation solutions, LTA makes a wide range of land transport-related datasets available on DataMall for businesses, independent developers, researchers, and other members of the public.

Real-time bus arrival information at a designated bus stop is one of the datasets that are available.

I must load real-time stream data from LTA DataMall into ASG2 for this assignment, then use Alteryx Designer to create a simple data pipeline that will display the real-time bus arrival time at any given bus stop.

It is the start of the year 2023, new badge of students that have just gotten their o levels result have chosen Ngee Ann Polytechnic as their choice of option for technical diploma course is coming to Ngee Ann Polytechnic this April to attend their first semester in school, and would like to know the options available to go to school, and would also like to know the estimated time of arrival. Hence, we will be extracting data that was stated above and create a pipeline of the data given which will be turn into an app where students and lecturers can just enter the bus code of bus stop and the pipeline will just run through and collect the data from the API which will be stored in an excel that can be supplied to another team that create user interfaces and web interfaces for further development. Which can ultimately be usable for public usage.

With this, students and lecturers will be able to schedule their time ahead and will not turn up late for their lessons/lectures in school.

# Preparation

A picture containing diagram

Description automatically generated

This project is done with 4 different stages, where the first stage will be the preprocessing and preparation steps, which I have to apply the first tool called text input which I have configured with the input, one is called URL, which contains the link to the API that contains the dataset of the nation wide bus stops, estimated timing and services. Another column will also be added that contains the unique bus stop code of Ngee Ann polytechnic that is defined as bus code “12101”. With the 2 features mentioned above, I will now move on to the next tool which is the select tools where I will be converting the data type of bus code from integer to variable string so that we can connect the URL and the bus code in the next tool. Which brought to us the 3rd tool we will be applying during the preparation part, called formula where I will be combining the URL and the bus code into a full URL with the formulation [URL] + '?BusStopCode=' + [BusCode] to access the full dataset of specific bus stop. Finally, after combining the URL and BusCode into one hyperlink we can now remove the bus code from the 2 features as we have already combined it during the formula tool. We will now move on to the API section to describe more on the process and the tools I have applied.

# API Call

A picture containing timeline

Description automatically generated

Now, I will be explaining the second section of the project which is the API call section where I will actually extract the website API data which the URL was provided from the earlier section from where we processed the URL and bus code to define specific bus stop for the data pipeline to extract data from. First tool we have implemented will be the download tool, with the user access key that we have registered for on the LTA website, which later on provided us with the access key to access their API data, also implementing the payload to access the URL with the usage of our access key in order to extract the data. Which is shown in the output where it stated HTTP/1.1 200 OK instead of 400 which usually meant for bad request/access.

A screenshot of a computer

Description automatically generated with medium confidenceA screenshot of a computer

Description automatically generated with medium confidence

Next by applying Text to column tool, I was able to split the downloaded data into rows uniquely with the delimiter usage of “\n” to help section it out nicely. Next would be implementing the select tool where I can tick the features, I want to implement which in this case I have remove the headers as it does not bring any significant value to the dataset other than the data which was categorized by it.

Graphical user interface, text, application

Description automatically generated

Next I have used the data cleaning tool to clean the feature “downloaded data” and replace values such as null and 0 or unwanted characters such as leading and trailing whitespace. Before using the last tool in API call section “JSON parse” to parse through the JSON content into individuals’ rows with quality content and values that is easy to understand and implement in the next section.

# Data Display

Chart

Description automatically generated

For the third section of the project, we will be applying the cleanse data from the API which was also parse through in the previous section into meaningful data, we will first filter the mass amount of data into specific data that was asked in the requirement.

A screenshot of a computer

Description automatically generated with medium confidence

Which in this case we will be containing values that have Estimated time which is the timing where the bus arrive to the specified bus stop, Service No which is the bus number that is in service during that specific timing which is also based off real time and lastly the destination code so that users will know where the bus they have entered will be stopping off at which interchange. By applying Filter tool, we can successfully obtain the 3 features within the wide pool of dataset the API provided.

A screenshot of a computer

Description automatically generated with medium confidenceA screenshot of a computer

Description automatically generated with medium confidence

Text to column was also applied afterwards to split JSON name into 3 columns to section the features with meaningful category that will be used as a column header later. Next, I will be applying the Select tools to uncheck columns that I will no longer be needing for my dataset which is the JSON name that has been split previously and JSON Name 1 as it has record that is too generalized for usage.

A screenshot of a group data

Description automatically generated with medium confidenceA screenshot of a computer

Description automatically generated with medium confidence

Next will be applying Cross Tab to group the values by JSON\_name2 which in this case are numbers that sections each different bus services and we will be selecting JSON\_Name3 as the header to categorized them with unique features with the method concatenate to aggregate the values.

We will now be ready to select features that we will be finalizing into the final dataset before implementation of additional tools to save the dataset and making it more automated for which user can choose their designated bus code. Below is the final dataset sample which consists of the features that are required in the project.

Graphical user interface, text, table

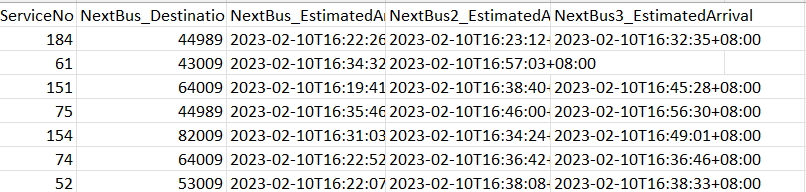
Description automatically generated

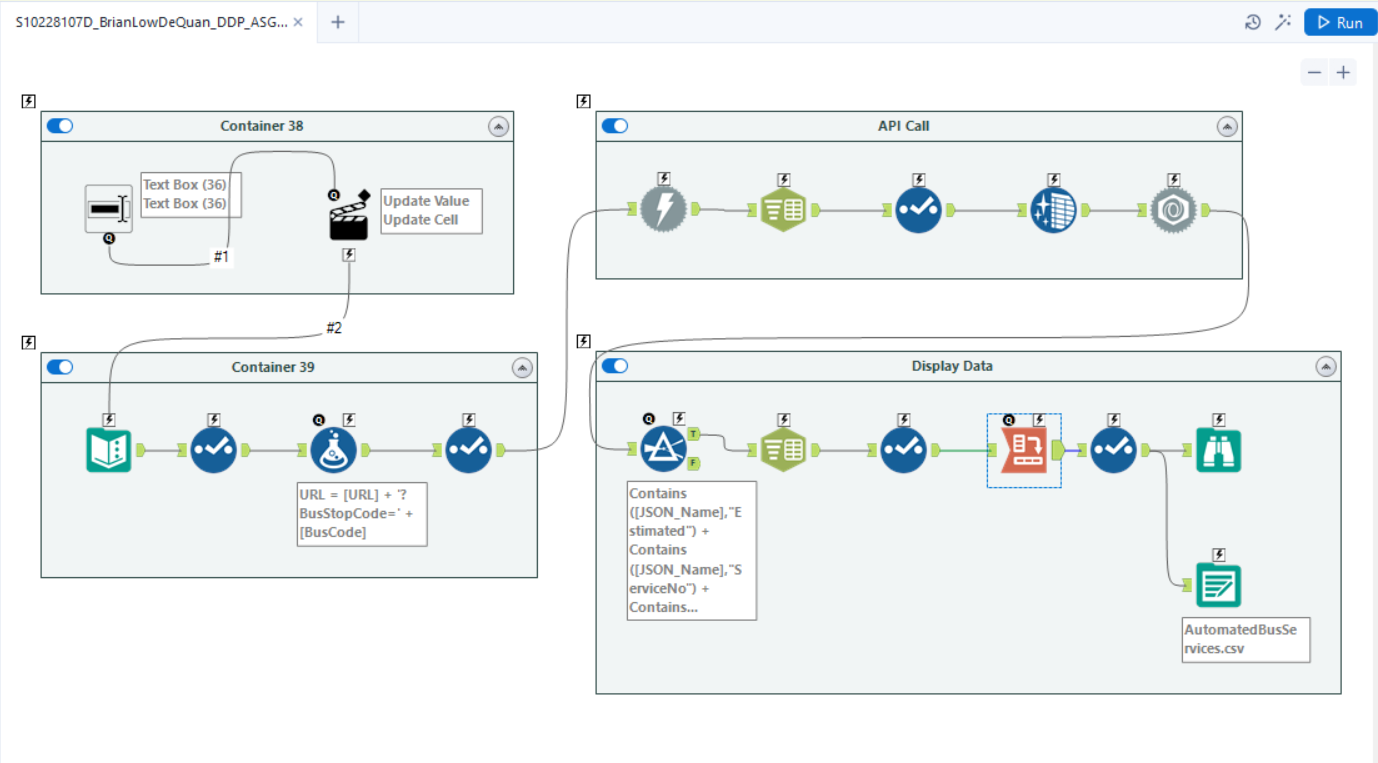
# Enhancement

Diagram

Description automatically generated

For the additional features I have applied text box, and Output Data that was set up with the connection to my file directory excel sheet called “AutomatedBusService.csv”. How this tools work is when user runs the workflow, it will prompt out a text box to ask user to enter their bus code, which the workflow will pass through the information by replacing the current bus code to the bus code defined by the user which will rewrite the information to the excel sheet. Which was shown below. Must click **the magic wand** beside the run button for the workflow to work with the additional implementation. For example, instead of running the predetermined bus code provided “12101”, users get to input the desired bus code to have access towards the upcoming bus arrival service timing.





# Summary

While working on the project, there are some limitations regarding the usage of magic wands.

One would be running the magic wand as shown as analytical app which does not show the output on the Output Log after running the entire workflow which may not be great for checking as user would not be able to check whether the dataset obtain from the API is as accurate as possible and would have to check if the information obtain for the LTA API is accurate in the excel that it is save on. Which requires developers to spend more time on the checking process.

A screenshot of a computer

Description automatically generated

As compared to running the workflow as usual as shown in the “Data Display” section which clearly shows the output in the browse tool for user to check whether they have successfully extracted the information from the API.

Graphical user interface, text, table

Description automatically generated

# Reflection

I would hope that in the future when I get to work on this project once more, I would want to implement a database to keep the information obtained and apply it into Business Intelligence tool for further implementation that could help users or company understand more about their system. Next would be allowing further implementation as an actual app as shown in real life example such as “Moovit” which they extract the same information and display it on the app for users to view as they type the destination bus code as the place they would like to go with the current location as their starting point.

When working on the project, I have learnt how easily it is to implement API call using Alteryx and implement it into the data pipeline with ease that is also easy to understand, some stuff I can probably implement it better would be the way I display my dataset as there is much better and concise way of displaying the data given to me, however I was not able to find the tools to help me segment the data more comprehensively which result in lesser column of dataset being applied in this LTA Bus Stop Service project.