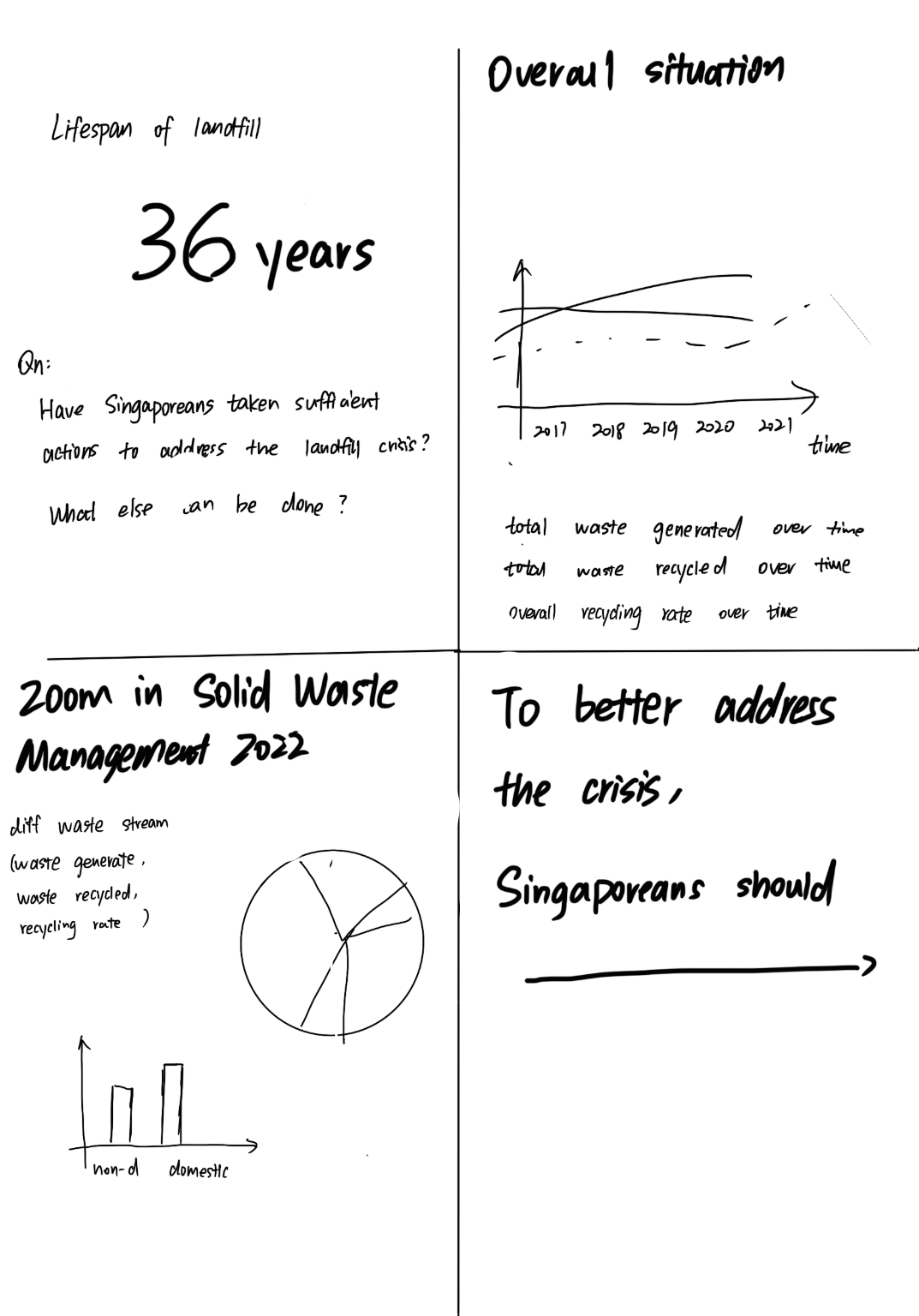
Project Documentation For NM2207

Draft 1:­­­­



Draft 2:

­After consulting the professor, I make some changes and here is the new draft:

Diagram

Description automatically generated

Main Sources of Data:

Key Environmental Statistics 2020

(https://www.mse.gov.sg/resources/key-environmental-statistics.pdf)

Key Environmental Statistics 2021

(https://www.mse.gov.sg/files/resources/Key-Environmental-Statistics-2021-Publication.pdf)

Waste and Statistics from 1990 to 2020

(https://www.nea.gov.sg/docs/default-source/default-document-library/waste-and-recycling-statistics-2017-to-2020.pdf)

Also I refer to domestic recycling rate in Finland:

https://www.hsy.fi/en/waste-and-recycling/waste-statistics/

­

After two weeks of hard work, I manage to create something like this:

Graphical user interface, application

Description automatically generatedChart, waterfall chart

Description automatically generated

Chart, scatter chart

Description automatically generated

You can check the page and play with the interactive element in my NM2207 homepage or Github archive.

Following is a break down of code, where I will explain how I apply the knowledge I learned in this course.

1. Function:

I use the concept of function to create a label for my chart to extract data from my data object and customize the message being hovered over in the tooltip. (see dfChart.js line 43 – 56).

I define a function called label inside the callbacks object in the tooltips property. This function takes two arguments: tooltipItem and data.The tooltipItem argument contains information about the tooltip, such as its index and dataset index. The data argument contains the chart data.Within the label function, I define a variable called datasetLabel to store the label of the current dataset being hovered over in the tooltip. In this way, I create a custom label to display "Waste Recycled", "Waste Disposed" and "Waste Generated. "

By using a function to define the tooltip label, I have made my code more reusable and modular. Specifically speaking ,I modify the same block of code and apply them to other charts to create tooltip label. (see dfChart.js line 301 -307 )

1. Document Object Mode(DOM) and Canvas API

In this line of code (see dfChart.js line 82), *document.getElementById("chart1")* is used to retrieve a reference to a canvas element in the HTML document with the id "chart1". The *getContext('2d')* method is then called on this canvas element to return a 2D rendering context for the canvas, which can be used to draw various shapes and graphics. The Chart.js library is then used to create a new chart object on this canvas, passing in the data and options for the chart as parameters.

1. Data Structure: Array and Object

I create *arrays totalWasteGenerated, totalWasteRecycled* and *totalWasteDisposed* (see dfChart.js line 7-10), to store data retrived from data source.

Then I created object *data1* (see dfChart.js line 12-30), which contains two keys, labels and datasets. The labels key has a value of the year array. The datasets key has a value of an array of two objects, each representing a dataset in the chart. For the styling purpose, each object has keys for the label, data, backgroundColor, borderColor, and borderWidth of the dataset. The *data1* object is then used as data source to create chart1.

1. Document Object Mode(DOM) and Cascading Style Sheets(CSS)

In this code, I use the canvas element, which is a DOM element that allows us to draw graphics on the web page. The canvas elements are assigned an id attribute, which can be used to reference it in JavaScript code. They are also styled using the .canvas-container class, which has the various CSS properties for styling. (see index.html line 22-32).

As this is only the second round of prototyping, for the upcoming three weeks, I will:

1. Add in event and event listener to my chart, so when the mouse is hovered, the explantion and insights will jump.
2. Apply HTML grid system to place charts and headlines in a more harmonious way.
3. Choose a nice color palette and apply it in css styling.

Maybe there will be more changes. Let’s look forward to that!