Seattle Transportation – UW Transit Project

Students: Xinyi Lyu, Janice Kim, Kelly Wang, Elaine Zhang, Fang Yu Lim (Fiona), Sue Boyd

Link to Visualization: Seattle_Transit (https://public.tableau.com/app/profile/kelly.wang5564/viz/Seattle_Transit/Landing?publish=yes)

Executive Summary

The Seattle Transportation project began with the motivation to take the bus on time. Most UW students who commute by bus regularly have experienced changing plans or missing the bus due to buses not arriving at scheduled times. To mitigate this issue, we decided to analyze the on-time performance data for buses that stop by the UW campus with various contextual data sets. The main goals of this project include understanding historical on-time performance for buses that stop near the University of Washington, helping users examine and compare historical on-time performance for overall and specific routes, and exploring the relationship between on-time performance and different variables such as football games, time and day, and weather conditions.

Some questions we wanted to answer include: Which buses are usually on time, early, or late? Is the bus less on time when it is raining outside? If one intends to take bus 31 in the morning on weekdays, how on time has the bus been historically during that time period? This project is not intended to show real-time locations of buses or predict real-time performance.

Concept Background

Since the project was initially aimed at understanding the on-time performance of Seattle public transportation, we started by including all the bus routes in Seattle, then narrowing it down to King County, and eventually focusing specifically on the routes and stops around the UW campus based on our target users - UW students. We also researched factors affecting the experience of bus riders in general, such as traffic conditions, weather, day of the week, time of the day, frequency of bus schedule, capacity of buses, crowdedness of passengers, etc. Due to the availability of data sources, we decided to focus on on-time performance and related features that UW students are most likely to be interested in, including bus frequency, time, day, weather, and UW football events.

We started our visual concepts by preparing the datasets. To better understand bus on-time performance and tendencies under different conditions, we needed historical data regarding a bus's arrival in comparison to its scheduled arrival time on a daily basis and the different conditions for each day. The final data we used to build the dashboard consists of filtered routes and bus stops, daily on-time performance, bus schedule, bus frequency, weather conditions, and UW football events schedule. (See Schema in Figure 1) The detailed data handling process is described in the rest of the section.

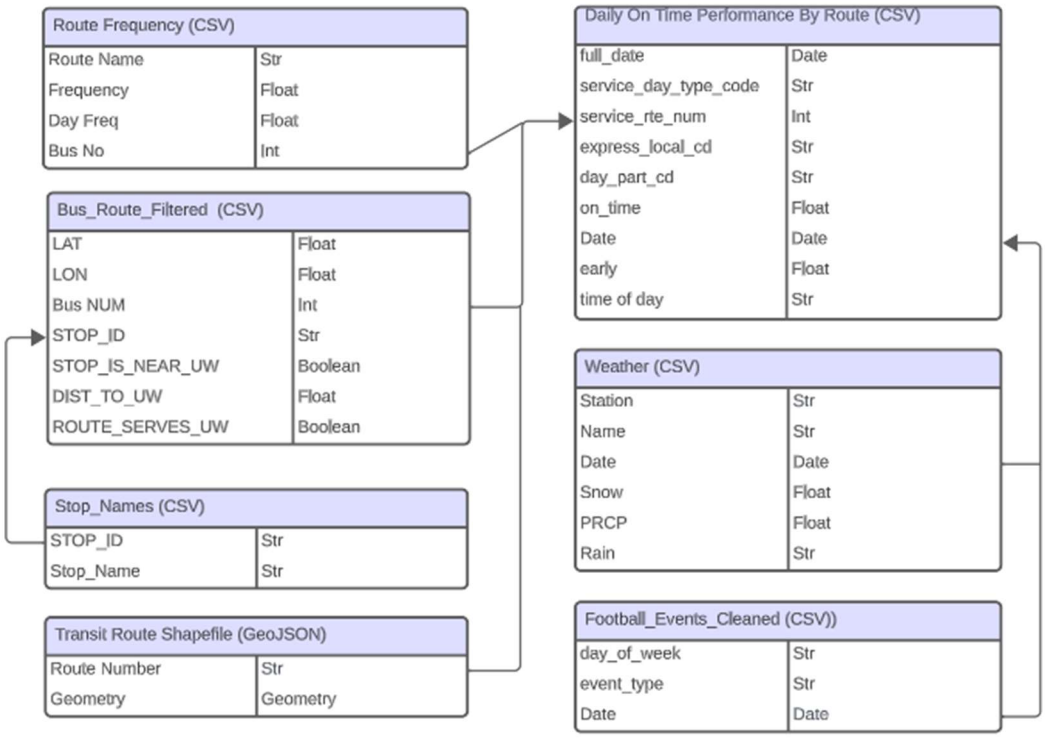

Figure 1: Final Data Schema In Tableau

First, we must determine which routes and stops we would focus on and filter accordingly. Thus, we created the data table 'Bus_Route_Filtered' from the raw data file 'Transit_Stops_for_King_County_Metro___transitstop_point.csv' obtained from the King County GIS Open Data portal. Data cleaning included removing duplicate stop values and splitting the original 'Route List' column into multiple columns when a single stop was associated with more than one bus route. We then melted down the bus number into a single column and extracted an intermediate data file 'Bus_Routes.csv' that contained only the columns 'X', 'Y', 'STOP_ID', and 'BUS_NUM'. We then further processed the data to select only those Bus_Routes that were relevant to UW students. We renamed the 'X' and 'Y' fields from the original dataset to 'LON' and 'LAT', respectively. We created a 0.4 by 0.4 mile square grid centered on UW and calculated whether each stop was within the square grid. The results from that comparison and the calculated distance from the stop to the Center of UW were recorded in the 'STOP_IS_NEAR_UW' and 'DIST_TO_UW', respectively. For each route, we then calculated whether at least one stop in that route was near UW, and filtered our dataset to include only the data from those bus routes. In addition to the longitude and latitude data, we also acquired the route geojson file from the King County GIS Open Data portal. The geojson file includes the route number and the geometry value used to plot the shape out in Tableau.

After the initial clean, we came back to the backbone of our dashboard, the on-time performance data provided by King County Metro. After an extensive search for historical bus arrival data, we found the King County Rider Dashboard which includes a basic report for on-time performance. We contacted the owner of the dashboard, Andrew Brick at King County Metro, to acquire the data powering their dashboard. To join the on-time data with factors like weather and events, we requested the data on a daily granular basis instead of the aggregated monthly view

shown in the King County Rider Dashboard. To clean up the data a bit, we used RegEx in Python to remove any letters in the route names to match the on-time performance data so we could join the datasets together in Tableau. We then used Python to obtain the list of routes in the 'Bus_Route_Filtered' table to filter the data in the On Time Performance table down to only the routes that had bus stops near UW. We also created a column that translated the time of day three-letter code to the actual time of day for use as a filter on the compare page. The on-time performance data is from January 1, 2019, through October 31, 2023.

However, later in the process, we realized we also wanted to display stop names in our map tooltip but had not captured that field in the Bus Route Filtered data table. Hence, we went back to the original data file "Transit_Stops_for_King - _County_Metro___transitstop_point.csv" and created a second table with both Stop Number and Stop Names so that this information could be joined in Tableau.

Other primary features we considered were the bus schedule and the corresponding arrival frequency of each route. The bus schedule data was combined from two open sources: King County Metro Develop Resource and One Bus Away data. The primary data source is the GFTS Feed from the King County Metro Developer Portal, which includes separate files describing bus routes, bus stops, stop times, and corresponding calendars. The first step is extracting the relevant columns from each table and joining them on common attributes such as trip_id, route_id, etc, to get a merged dataset with all required attributes in Python. To match the date data format with the other variables in the database, we applied an additional pivot step to melt down the days of the week to a single date column. Furthermore, we filtered the data based on the 'Bus_Route_Filtered' table to focus on our target routes only. However, the data is still incomplete, with specific routes or dates missing. Thus, we reached out to another common developer resource, One Bus Away, and requested API access as a supplemental resource. Another limitation of the primary data is that it only contains the regular schedule but lacks arrangements for holidays and changing conditions. We tried to solve the problem by extracting dynamic schedules with One Bus Away API, but the dataset overextended to over two million rows for a single route. Hence, we choose to use the processible version of the regular schedule only.

For bus frequency, since the data provided by the King County Rider Dashboard owner only contained weekday information and the expression was hard to interpret, we derived the bus frequency data from the bus schedule file by computing the time intervals between arrivals of each route at one stop using Python. The original plan was to divide one day into several periods where average time intervals are calculated based on the time of the day. Nevertheless, because the frequency distribution for each route varies greatly, the over classification of periods not only added difficulty when implementing filters but also broke the time trends of other factors. Thus, we dropped the fields of time of the day and presented the frequency based only on weekdays or weekends.

Next, to create the filter on weather conditions, we exported Seattle's daily weather data from the NOAA's website for January 1, 2019, through October 31, 2023. We explored the data in Excel and determined that the Seattle Sand Point Weather Forecast Office had the most complete set of precipitation data and was relatively close to the UW campus. We created a new column that categorized the numerical data into filterable, ordinal values using functions in Excel and rain guidelines to define the different levels of rainfall. For example, if a day recorded 0.1 inches of rain, the new column would denote this as 'light rain'. Due to limited data in other categories such as snow, fog, and hail, we decided to only use rain for our weather condition filter.

Lastly, we scraped football game data from the University of Washington Football Schedule. We inspected the HTML structure and displayed the data in a grid format to ensure the content was more structured for easier scraping. The URL was split into 2 parts: the base URL and the year number. This allowed us to iterate through each year in the range from 2019 to 2023. For each year, we pulled the HTML content using the 'request' library and parsed the HTML content with the 'BeautifulSoup' library. In 2019 and 2022, there is an additional column 'Tournament', so we skipped this column in these years. Subsequently, we wrote extracted data into a CSV file.

The data cleaning process for the football game data included dropping columns named opponents, location_state, TV, radio, result_2, and links, which were the columns we were not interested in. Initially, the column 'result_1' was included as it includes cancellation information. If a game was canceled, it wouldn't affect traffic on that particular day. However, after careful observations, the column 'result_1' was dropped as the data entry was inconsistent and would not provide beneficial insights. Since we are looking at traffic in Seattle, we only kept entries of home games and dropped the columns 'at' and 'location_city' after confirming. Next, we changed the time column to a 24-hours format, and split 'year_dayofweek' into 'year' and 'day_of_week'. The 'date' column was modified to be combined with the 'year' column into a month-year-day format. We added a column called 'event type' and labeled these entries as 'football game'. Lastly, we generated a dataset of all days from January 1, 2019, to October 31, 2023, and their corresponding day of the week. We joined the two datasets, and for days when no football game occurred, we labeled it as 'no football game'.

Initially, we also wanted to scrape the UW Event Calendar website to obtain a whole list of events that are happening on campus. We found that the information on the UW Event Calendar website is loaded dynamically within an iframe after user actions, such as clicking the back button or filtering based on event type. Consequently, scraping the event data we are interested in became more challenging. After manual inspection, we also realized that the website only dates back to September 2022. At the same time, as it encompassed every event, many of them were too minor to have a significant impact on traffic and bus on-time performance. Hence, we chose to discard the calendar website data since there were limited advantages in doing so.

We also generated a list of dates between January 1, 2019, and October 31, 2023, with the day of the week for each date to create the day-of-week filter. Since the on-time and frequency data only denoted weekend or weekday, we used this list to provide users with an option for a more detailed analysis.

Process Description

Initial Plan

The initial plan was to create this dashboard for the greater Seattle area. After obtaining the data, we found many different potential use cases we could pursue - students, office workers, and even city planners. Each of these use cases would require a different view of the same data. To create a more streamlined and targeted dashboard, we decided to limit our scope to students at the University of Washington. We also decided to make the distinction that our tool would not provide any predictive insights for the users, rather it should provide historical information to aid decision-making. In this way, we came up with the idea of having a page with a map of the routes and another page allowing for comparison between routes, assuming that users have different routes they usually choose between.

Initial Designs and Usability Tests

Starting with our design sketches and later our low-fidelity dashboard, we created our initial prototype of the two-page tool. On the left side of the first dashboard, we planned to plot the main map where users can view the routes that go through the UW campus and click on a particular route to filter. On the right side, we planned to locate plots for overall statistics of on-time performance that could potentially be filtered based on routes or time. On the second dashboard, we planned to have line graphs of two routes with times, events, and weather conditions as a filter. The tooltip that indicates the percentage of on-time performance was also included in the plan to display an aggregated on-time metric at a glance. Our design sketches can be found in the appendix.

We conducted usability tests with six users, three of whom were students in Data 511, using our low-fidelity mockups. Users were generally able to navigate through the dashboard with ease after a short description of the project. Users provided some suggestions for improvement, which we implemented in our final dashboard. On the 'Main' page, we put instructions and a 'How Do I Use This Dashboard' tutorial page based on user requests for more instructions. On the 'Visualize' page, bus frequency information was added into the stop tooltip based on the user feedback saying that frequency data would be helpful to get more context. On the 'Compare' page, we added a 'Reset Filters' button to make it more intuitive to reset different filters. Drop-down menus were also implemented for users to select the bus numbers for the routes they wanted to compare. Our initial design asked users to enter the route numbers in a text box, which some users found confusing and could be difficult if a user is not familiar with the existing routes.

There were some aspects of feedback that we were not able to include due to time and data constraints. Users suggested that information related to such as average delay time, route safety, and walk time references, would be helpful. While we agreed that many of these data points might be useful, including them was beyond the scope of the project given the time and data available.

Issues and Breakthroughs

One major issue we encountered during the data preparation step was the sheer size of the data. For each bus route, we had at least four to five rows per day, one for each time period of the day. When joined to the data with the stops for each route, the data expanded again. In order to improve performance, we trimmed down the on-time data to just the routes near UW and filtered the stops to the ones that were within the box we defined as near the UW campus.

We then started to build our dashboard in Tableau, where we encountered a problem with plotting the routes. We obtained the longitude and latitude of each stop in each route we wanted to plot, but Tableau would connect dots from left to right, top to bottom, and not in the shape the bus would take. Connecting the dots between stops also raised the problem of being unable to select a whole route, since clicking on a line would result in selecting one dot and not the whole shape. After some more exploration in the King County GIS Open Data portal, we found a geojson file with each route's shape. Loading these shapes into Tableau gave us a clean, accurate representation of the bus routes and fixed the issue of not being able to select a whole route. We also decided to include both routes and stops near UW so that a user can compare performance for different routes that pass through the same stop.

Another major issue after we started implementation was that we had to repeatedly go back to the data processing step after noticing issues with the joins in Tableau. While the on-time data had separated bus numbers and route type, the bus route data from the King County GIS Open Data portal had concatenated the bus numbers and route type into one route number. For example, route 988 was denoted as 988 in the on-time data and 988E in the route data. To facilitate the join and retain the data accurately, we confirmed that routes with letters in the number, like 988E, did not have another variation without the letter and could be matched with the letterless version in the on-time data. We cleaned the letters using regular expressions in Python. We also had to modify or create new fields or tables to accommodate functions like filters, as documented in the Concept Background section.

We also changed our plan of representation based on the effects of experimental visualizations. Initially, we had planned to show the on-time percentage on the compare page, with a different line for each year. However, after testing it with the other components of the dashboard, we found that it was confusing and not the most informative representation of the data. It was also rather similar to the King County Rider Dashboard we originally obtained the data. We then changed the line graphs to display the average on-time, late, and early percentages for each month over the selected time period. Doing so allows the user to see all three metrics and make more informed decisions since a bus being early versus late can result in different actions being taken.

Another modification involved the representation of bus schedules and frequency. We tried to include an additional bar chart describing bus frequency on the compare page. However, since the overcrowded graphs tend to distract the users' attention, we decided to focus mainly on on-time performance and moved bus frequency to the tooltip as supplemental information. Moreover, to make the tooltip view condensed and easy to comprehend, we added an external link to the trip planner website operated by the King County Metro instead of showing tables of schedules as we planned.

After receiving feedback from our peers in the presentation, we made a few final adjustments to our dashboard. First, we changed our yearly comparison donut charts to bar charts for a more intuitive design and visibility. We then changed the tooltip graph for each route to a donut chart since the tooltip for the routes only shows one route at a time and is not used for comparison between routes. We also added another page for more detailed instructions on how the dashboard is designed to be used and data definitions. We also noted that certain rain categories did not have quite enough data to make informed decisions, so we consolidated a few categories, going from seven categories to five. We tried adding a drop-down to select a route from the map but ran into an issue with the selection not properly filtering the graphs on the right. The bars would change with the drop-down but the title did not accurately reflect the selected route. Due to time constraints, we decided to leave it off to prevent confusion. We also added a ‘Clear Filters’ button to the compare dashboard to clear all filters at once.

Final Visualization & Critical Evaluation

Our final visualization consists of a landing page and three dashboards, which is fairly similar to our initial design. The first dashboard provides insight into how users can navigate through and use the dashboard. The route map page consists of two bar charts portraying the yearly and quarterly performance of all buses, which can be filtered to reflect the on-time performance for a specific route by selecting the route on the map view on the left. A donut chart showing the

on-time performance and the time interval between bus arrivals on weekdays and weekends for a specific route will be shown if users hover over the route. In addition, users can hover over a stop near UW to view the on-time performance for all buses that stop at the particular bus stop in the form of horizontal bar charts. The compare page consists of two line graphs that users can use to compare two bus routes with filters based on year, rain condition, event, day of week, and time of day. At the bottom of the graphs, we included the on-time percentage and the scheduled bus arrival frequency on weekdays and weekends.

For variable encodings, we applied lessons from Mackinlay's hierarchy of encoding methods [1]in determining how to encode the different variables displayed in our dashboard. As performance data (on time, early, and late) was our most important variable, we encoded these quantitative variables using color, position, length, or area, each of which ranks highly on Mackinlay's hierarchy for encoding quantitative data. To distinguish between categories of performance types, we encoded it as a hue and connected it with conventional color perceptions, with green being positivity, thus on-time, and red being negativity, thus late. For early, we decided to encode it with yellow, as it is also a relatively negative color, and the bus being early is not necessarily a good thing.

In the 'Visualize Performance' page, we used the length and the size of the bar chart to encode performance data for the main graphs. For the graphs in the tooltips that appear on hovering over a route, we encoded performance data using area. Regarding routes and stops, we used a map view to display the location as map views are particularly well suited to displaying geographical data. We also double-encoded each route using different hues to enable users to distinguish different routes using both geographical location and color labels. On the 'Compare' page, we mainly used positions for the line charts within two-dimensional coordinates. The y-axis encoded the performance percentage as a ratio while the x-axis encoded time as an interval variable.

Our dashboard also incorporated a number of practices recommended by Heer and Shneiderman in *Interactive Dynamics for Visual Analysis* (hereafter "H&S"). [2]On both our main visualization page and 'Compare' page, we paired the ability for users to filter the data with real-time updates to the corresponding visualizations, enabling users to probe areas of particular to them, as H&S recommended. Likewise, we allowed users to pan and filter in the map view. By combining linked maps and bar charts on our main visualization page, we employed the strategy of 'linked visualization' and 'multiple coordinated views' to allow users to simultaneously view the data along multiple dimensions including geography, time, and bus route/stop number. Finally, our decision to filter data to include only routes and stops near UW for performance reasons took into account the guidance from H&S that 'slow response times' can inhibit use and satisfaction.

Moreover, we incorporated best practices from Tufte, *Graphical Integrity*[3], including providing data labels and explanations. We ensured that our graph axes for performance data always started at zero and ended at one, which is the full range of the dataset, to have consistency, avoid data distortion, and enable comparison between graphs.

Through this visualization, we found that of the routes that stop near UW, bus 43 is the least on-time, with an overall on-time percentage of 61%. An interesting point for bus 43 is that

---

[1] Mannheimer, *Week 2 DATA 511 Graphical Excellence and Integrity.pptx*,pp.15-16, citing Mackinlay (1986), *Automating the Design of Graphical Presentations of Relational Information*

[2] Heer, J., & Shneiderman, B. (2012). Interactive Dynamics for visual analysis. *Communications of the ACM*, *55*(4), 45–54. https://doi.org/10.1145/2133806.2133821

[3] Tufte, E. R. (2004). *The visual display of quantitative information*. Graphics Press.

though the route tends to have more late arrivals than early arrivals, the percentage of early arrivals was almost two times higher than late arrivals in 2020. This difference may be due to service changes in response to the COVID-19 pandemic that year. We also found that bus 73, unlike other routes, has become more on-time over the past few years. Bus 73 started at 80.72% on time in 2019 and has increased to 91.21% as of October 31, 2023.

We also noted that service changes in 2020 due to the pandemic may have impacted the data obtained in the last few years. For example, route 31 reduced its service from six days a week to five from 2020 through September 2021 before returning to operating every day. Route 988 also stopped service from 2020 April to 2021 September. Because the majority of 2021 does not have the service data, route 988 had 58.98% late performance in that year. The reduced frequency may have also impacted the amount of data gathered from 2020 to late 2021 when service returned to normal levels. Similarly, we have less football game data due to reduced games during the pandemic.

With this visualization, we successfully addressed the questions about the on-time performance of each route and stop within the University of Washington. Besides overall performance, the visualization demonstrates yearly and quarterly time trends of each route. Using the overall statistics as a default in the form of a pie chart, users can naturally narrow down from yearly to quarterly statistics as they are virtually located. Users can also easily pin down different routes from the average overall performance by hovering and clicking around the main map. We also answered the questions with specific context such as, “Would route 31 generally be late on a rainy day?”, by utilizing the comparison page. Through the comparison page, users can filter and see how two buses perform based on year, time of the day, day of the week, football games, and weather conditions. This provides contextual information to users, which helps amplify cognition besides numeric performance. Adding text that indicates the on-time performance rate, the information that most users look for, and information that mentions the absence of data, could increase data integrity and prevent misleading from the visualization.

Potential improvements to our visualization include geographical expansion, an increase in a number of route comparisons, and an exploration of specific relationships between on-time performance and contextual variables. Due to the size of the data, we limited our geographical area to the UW campus. Expanding to the greater Seattle area would expand our target users and demonstrate richer information. We also obtained other data from the King County Metro Rider Dashboard regarding ridership and capacity but ultimately did not include it in our dashboard because of performance issues and time constraints. To prevent the visualization from harming the visibility of users and becoming too overwhelming, we limited our comparison options to two routes only. However, allowing comparison based on all buses that stop at a certain bus stop would be helpful for users who are considering different route options at a particular stop. Lastly, even if a user can individually filter the contextual variables, it is difficult to see the relationship between different routes and weather conditions. Like the suggestion from the presentation, adding rainy days, football games, and other special occasions as a highlighter instead of the filter can help view the relationship between particular weather conditions or events and on-time performance. Although we can answer if route 31 usually arrives late on rainy days, we do not know if rainy days are more correlated with the higher late rate in general. Investigation of this general correlation relationship can be an interesting future direction.

Reference

Heer, J., & Shneiderman, B. (2012). Interactive Dynamics for visual analysis. *Communications of the ACM*, *55*(4), 45–54. https://doi.org/10.1145/2133806.2133821

Mackinlay, J. (1986). Automating the design of graphical presentations of relational information. *ACM Transactions on Graphics*, *5*(2), 110–141. https://doi.org/10.1145/22949.22950

Tufte, E. R. (2004). *The visual display of quantitative information*. Graphics Press.

Appendix

Design Sketches

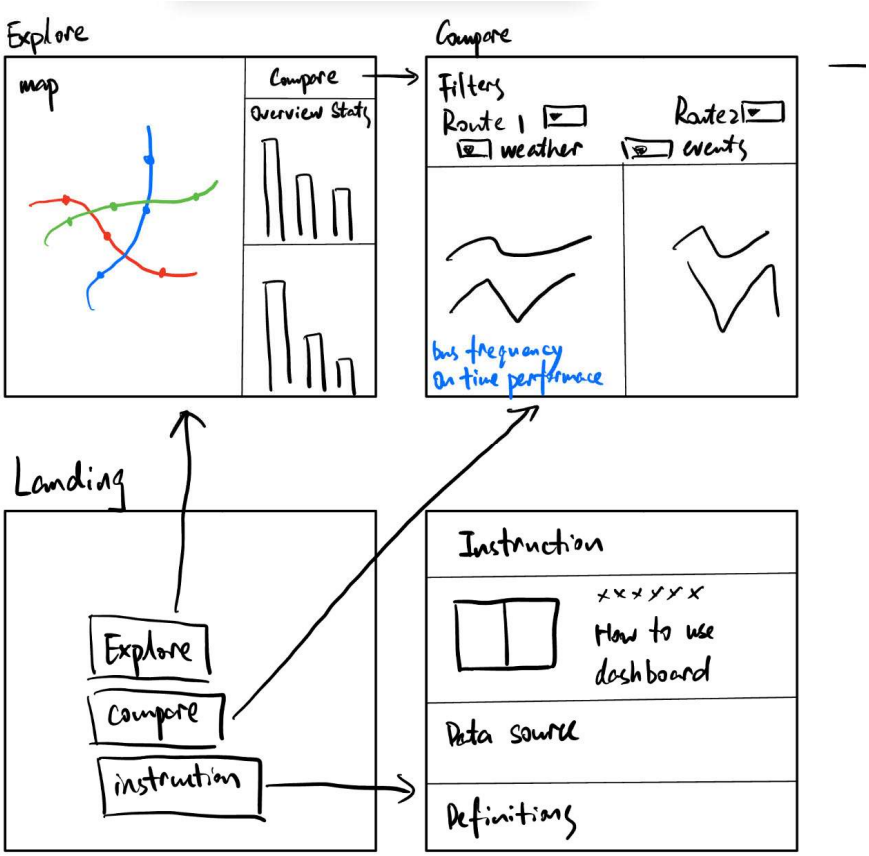

Figure 2: Final Design Prototype

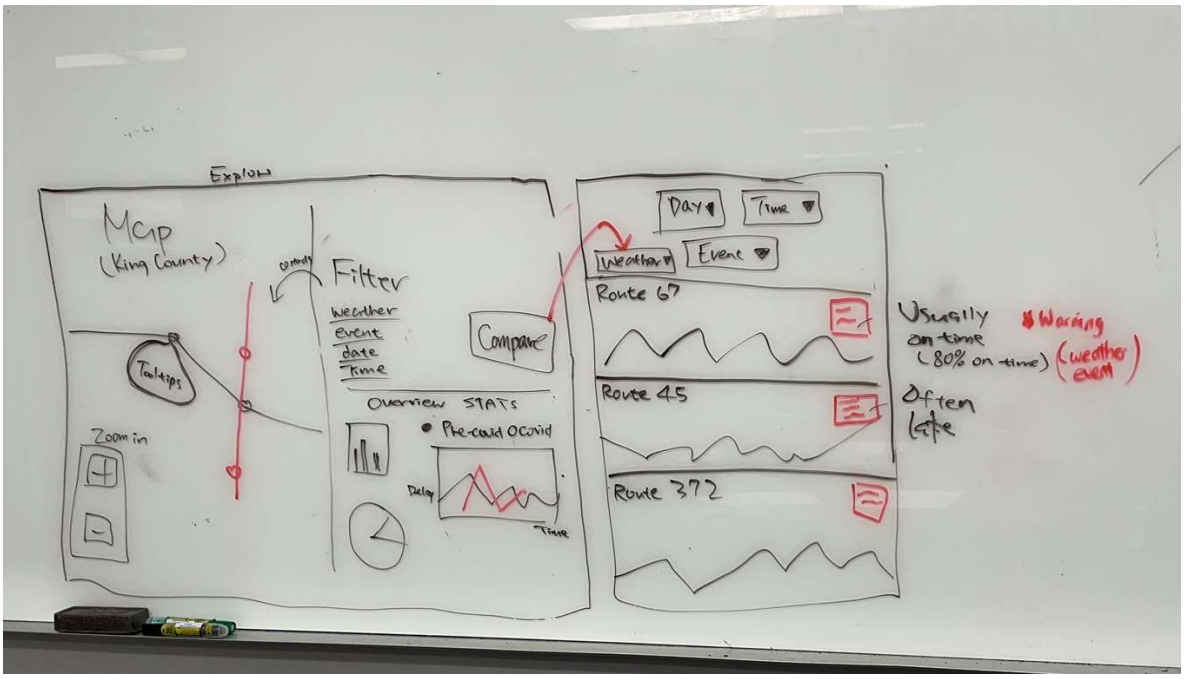

Figure 3: Initial Design Prototype

Supplemental Data Information

Rain guidelines https://weatherins.com/rain-guidelines/

King County Rider Dashboard
https://kingcounty.gov/en/legacy/depts/transportation/metro/about/accountability-center/rider-dashboard

Weather NOAA https://ncdc.noaa.gov/cdo-web/