Niner Transit Data

Team 14

Felipe Orrego, Thien Nguyen, Autumn Horrell, Django Yepidan, Jaren Robbins

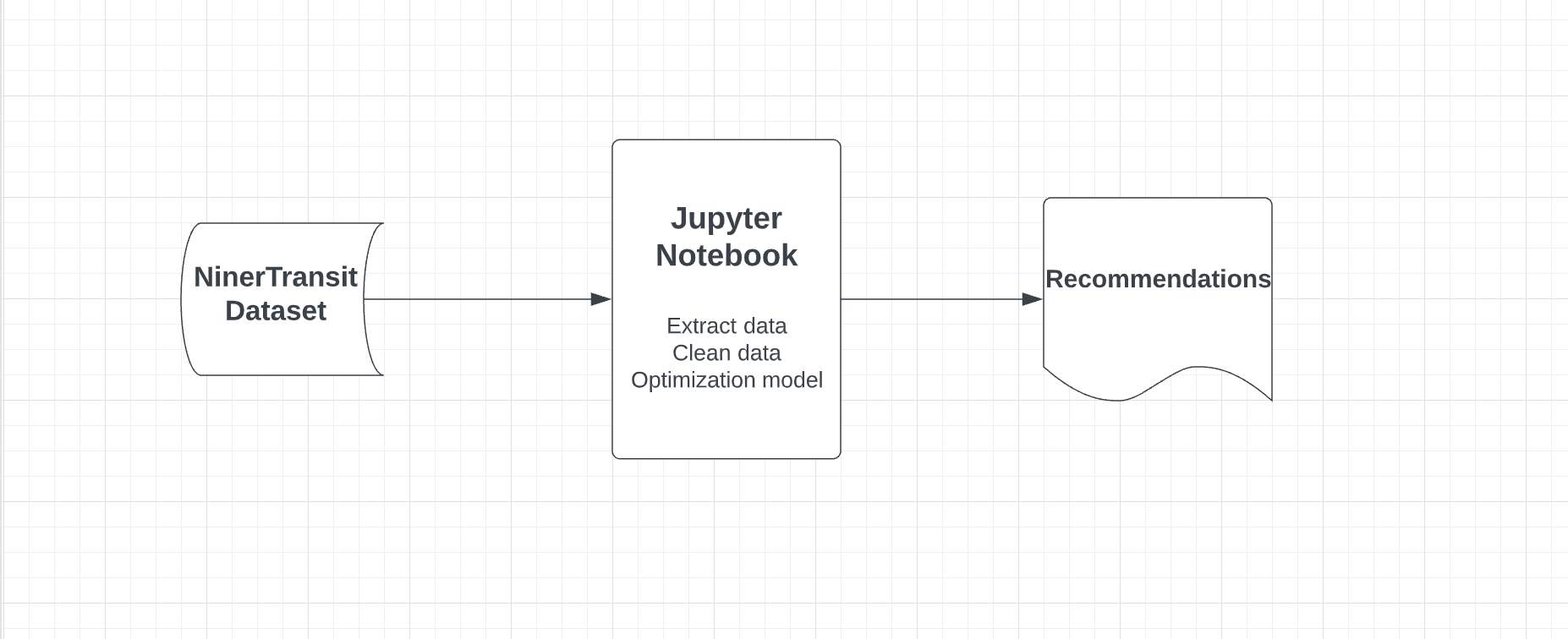
# Project Overview

The University of North Carolina at Charlotte constantly grows and inhabits new students each year. In correlation with the continual growth, services provided by the university tend to follow the same pattern. Unfortunately, one amenity that has failed to be improved is the transportation service. Students traveling to classes, dormitories, or other extracurricular activities rely on the Niner Transit system to get from point A to B. In addition, faculty members and university workers use the service as well. Therefore, the transportation service needs to adapt and increase its efficiency standards for the Niner Transit system to accommodate these growing demands. Our group has analyzed the provided data through visualization and extraction in support of this. These actions will help us make conclusions and recommendations regarding optimizing the transportation system.

# Architectural Overview

Since our project isn't necessarily an actual product, we don't have one set deliverable like a website or an app. Instead, our goal for this project is to use our databases to visualize and generalize data about our Niner Transit System. Our leading platforms for performing tasks were MySQL and Jupyter. Both of these platforms allow us to take the given dataset and manipulate it in ways that can help us generate the results we want. MySQL allows for more straightforward queries that don't require programming, while Jupyter covers the more complicated stuff. Jupyter is a platform that allows users to write Python code and run it with ease. Python is the language of choice for us because it includes a few libraries that help us immensely: Pandas, NumPy, and Matplotlib. We use Pandas to transfer our databases into a single DataFrame that we can manipulate. NumPy provides us with comprehensive math functions and tools. Matplotlib gives us the tools to make graphs. Using these materials and collaborating with everyone on the team can help us develop many different visuals that can hopefully lead us to some conclusions on how to optimize the Niner Transit System.

## Subsystem Architecture



Using the NinerTransit data from the school, we will use the Jupyter Notebook to extract and clean the data to eliminate any noise such as null values or missing values using Numpy and Pandas packages in Python. Moreover, we will also drop the columns containing unnecessary data or too many missing values to make our data clean, neat, and easy to work with. After cleaning the data, we will create some visualizations to help us get more insights and a deeper understanding of our data in Jupyter Notebook by using the Matplotlib package in Python. Finally, after understanding our data, we will split our data into the training and testing data to build and train some predictive machine learning models to create a proper set of recommendations for the board to optimize the different lines at UNC Charlotte.

## Deployment Architecture

Our project aims to analyze the data provided and develop recommendations to optimize the different Lines (Silver, Gold, and Green). It is not software that will run on multiple systems.

## Persistent Data Storage

We stored our data in Excel worksheets. Concretely, we created three Excel worksheets to store our Niner transit data, one for Gold Line, one for Green Line, and one for Silver Line. Then we read these worksheets into SQL to create nested queries to get the desired answers and data in new tables. Using SQL, we created three tables. The first table is for Gold Line has three columns, including stop, averageOn, averageOff. Concretely, the stop column contains the name of all visits in the Gold Line. The averageOn column contains the value of the average number of passengers getting on at each stop in the Gold Line. The averageOff column contains the value of the average number of passengers getting off at each stop in the Gold Line. The other two tables for Silver Line and Green Line have the same format as the table created for Gold Line with three columns, including the stop column, averageOn column, and averageOff column. Each column contains the same corresponding value for Silver Line and Green Line. After creating these three tables, including necessary data for Gold Line, Silver Line, and Green Line, we import these three tables into Jupyter Notebook to use Python with different packages. These packages, such as Pandas, Numpy, and Matplotlib, help us dive deeply into the data, create visualizations and build reports for our final product. Besides importing into Jupyter Notebook to work with Python, we also import these tables into RStudio to use the Leaflet Package in R to create interactive maps. To create interactive maps, after importing these tables into RStudio, we need to attribute these tables by adding two new columns named Longitude and Latitude using the built-in function ‘impute’ in R.

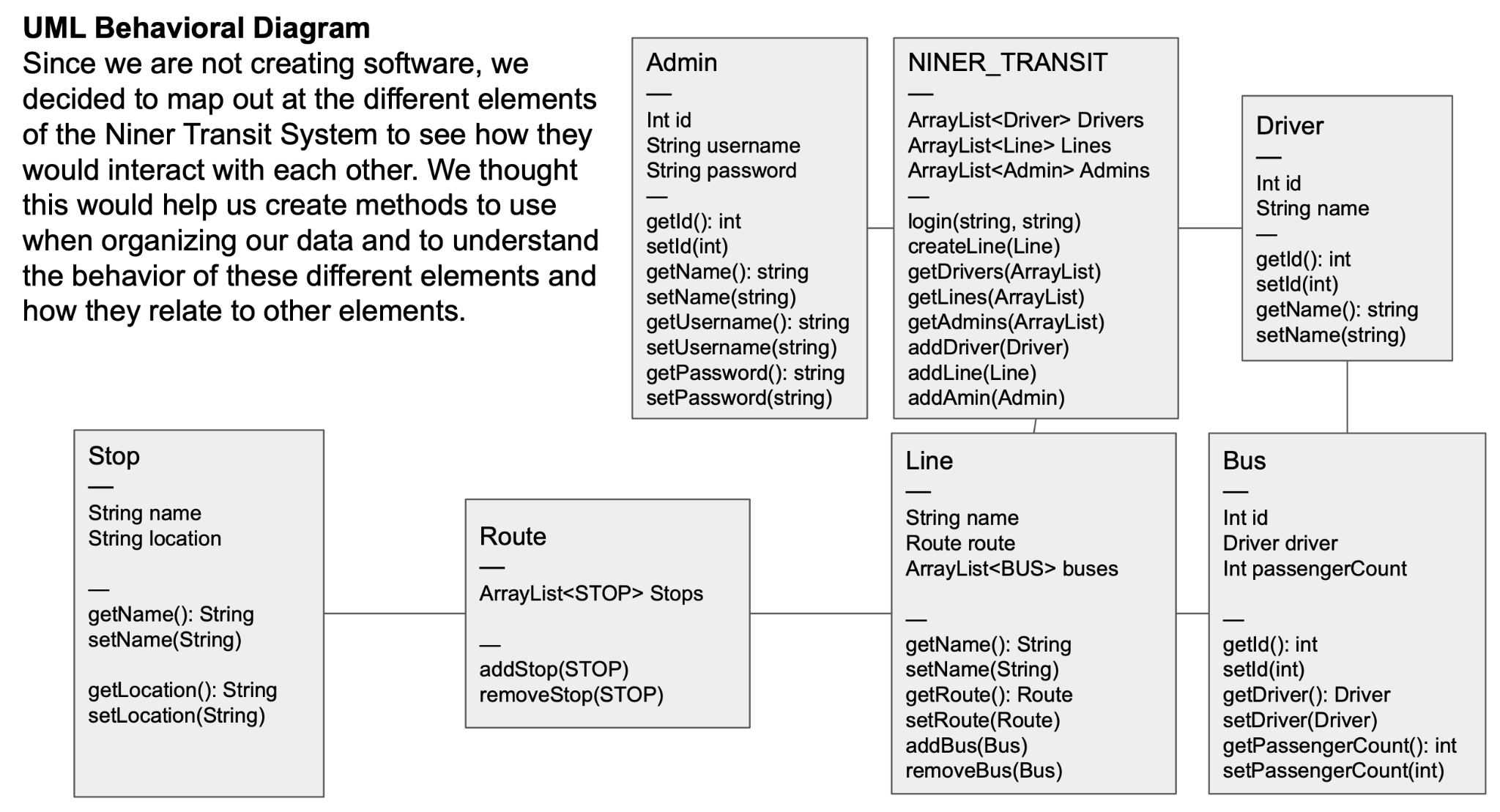
## Global Control Flow

Since we are not building any application and we are creating a proposal hosted on Streamlit so all the models of control flow, such as procedural or event-driven, time dependency, concurrency, will not be applied to our project. Concretely, we will host our proposal on Streamlit. Streamlit is an open-source app framework, so that we will present our proposal in the form of an interactive app framework. Still, it is not an application, so it does not apply any global control flow mentioned above. However, our proposal’s execution will be controlled entirely by the users. Therefore, we would say our proposal will have a ‘user-driven’ control flow since we can hold it in any order or way our users would like. Our Streamlit app framework will have a side panel containing options such as a searching bar, sliding bar, and checkbox, allowing users to interact with the visualizations. The users will be able to use these tools to interact with the visualizations in specific and the whole proposal in general in any order the users would like. That means the users do not have to use these tools to interact with anything. For example, the users do not have to use the searching bar first, then use the checkbox to select the timeframe of the visualization. One of the main goals of our project is to be able to give the users the freedom to explore and interact with our visualizations and our proposal as much as they would like.

# Detailed System Design

Since our project isn't an application, the system was working with the Niner Transit System. The lines go around the main campus in opposite directions, while the Silver line goes to the engineering side of campus. Each route includes a specific number of buses with drivers to take students to different parts of campus through bus stops. Below is a static and dynamic view of the system.

## Static view (UML Class Diagram)



## Dynamic view (Sequence Diagram to Create a New Line)

