Team Members:

- Ansh Pandey
- Cameron Morrow

Initial Decisions:

For the language we decided to use Python. It is a language that we both have a decent amount of experience in and it has some pretty powerful data manipulation methods.

Notes on Internal Architecture:

For the internal architecture we decided to use four different dictionaries each with different key-value pairs that go as follows:

1. { student last name, [student list)}
2. { teacher last name, [student list]}
3. { bus route number, [student list]}
4. { grade level, [student list]}

Even though four separate dictionaries take up much more space in memory, we decided to use this system because all of the operations that we need to support for this lab are based on finding specific data and not really focused on a scalable data table. Internally, Python dictionaries are just hash tables so that makes the look up operation O(1), so once the data file is processed, finding information is super quick. There is a decent bit of preprocessing to sort the data into these dictionaries but once that is done the actual operations are as efficient as possible.

Task Log:

1. ~ 2 hrs for the feature implementation

    a. Framework: Ansh
    b. Student: Ansh
    c. Teacher: Ansh
    d. Bus: Ansh
    e. Grade: Cameron
    f. Average: Cameron
    g. Info: Ansh
    h. Quit: Ansh
2. Manual testing found a couple bugs
    a. i.e. changed the way that the dictionaries was set up

Testing Notes:

Outside of the manual testing we did while creating our implementation of the features, Cameron wrote most of the tests with Ansh's input on what to include. While testing the exception edge cases, we realized that we had not properly handled invalid characters for `Bus`, `Grade`, and `Average`. These fixes were very quick to implement, and only extended the time to complete testing by ~10 minutes, for a full time of an hour and a half.

Final Comments:

We purposefully decided not to check the names of students and teachers for digits as, in the current landscape of our schooling system, it is entirely plausible that some students would want to go by another name and like to see it reflected in the system. We felt it was not our place to implement this restriction if this were to really be used as a database, so we reflected that in allowing numbers to be

parts of names. A few humorous examples when this would matter are if 50 Cent's kid, 25 Cent, were to enroll or any of the "boyz" from 'Boyz 2 Men' were to enroll.

Additionally, we decided not to clear the terminal each time an output is displayed in case the user needs to see the data of multiple queries at once. Instead, we provided them the option to clear the terminal as they needed.