Figure 1:

Inputs:
Student addresses
School address
Number of available buses
Bus capacity
Bus route time-slot
Region bounds

System:

First the application creates bus stops by organizing students into clusters based on student addresses.

Then the application uses search algorithms to create bus routes based on the bus stops and other inputs.

Outputs:
Gives a comprehensive list of bus routes that covers all student locations while minimizing distance/time traveled by all buses

Figure 2:

Input

Input Modifier:
Identifies the type of the input and passes it to the data transfer system.

If the input type happens to be an address the modifier will take the address and convert it over to a longitudinal and latitudinal coordinate and then pass it over to the data transfer system.

Data Transfer:
Takes in data to be added to the database and handles the data based on the type (i.e. routes, addresses, geocode, etc.) and pulls data from data base and formats data to pass to algorithm

Algorithm:
Takes location data and calculates shortest routes based on given number of routes

Display(Output):
Allows for user input for addresses, bus numbers, route generation, route updates, etc.
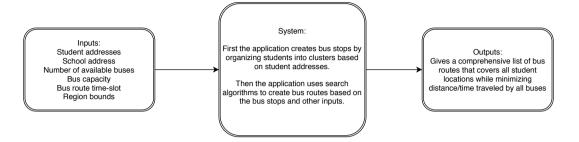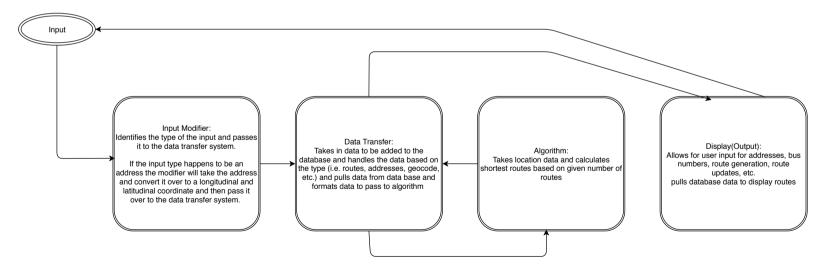pulls database data to display routes

Figure 1:

Data Format and Transfer:

The user individual student addresses will be collected at the beginning of the school year, these data points will then be fed into the system which will then format and transfer the data.

The address data will need to be formatted using into latitudinal and longitudinal points using the geocoding software ArcGis.

The formatted data will then be transferred over to a database which will map the geocoded coordinates for the student addresses to the students themselves for referencing

The database that will house the different data points will need a table that will store the individual students and addresses, a connected table that maps those students to their geocoded address, a table for each of the bus routes, and a table that outlines the assets of the school and their resources, such as buses and bus capacity

Algorithm

The individual student addresses will be collected at the beginning of the school year, these data points will then be fed into the system which will then format and transfer the data.

The address data will need to be formatted using into latitudinal and longitudinal points using the geocoding software ArcGis.

The formatted data will then be transferred over to a database which will map the geocoded coordinates for the student addresses to the students themselves for referencing

The data will be fed organized in a binary tree where each node will represent a single stop, each stop consisting of a geolocation as well as a number indicating how many students will be getting on at the stop.

Once the nodes have been built out each node will be associated with the various road nodes that will be supplied by the ArcGis data.

The distance between each node will be calculated boundaries will be built based on the size of the district and the number of nodes.

Each region will be assigned a specific number of buses based on the number of students in a region.

The distance between each node in a region will be calculated and any nodes within a certain radius will be combined into a single node where the bus will stop once to pick up all kids encompassed by that node.

Front End:

The main user of this system would be the transportation supervisor for a school.

The bus drivers would also have a limited access account that would allow them to see the routes assigned to them and also a list of students on their route.

The main functionality for a supervisor would be to enter in the number of buses, a leave time for the buses, a report time, the address of the school, add new students to the routes, and add/subtract buses from the pre-created routes.

All users would get a map of the district, the bus drivers would see the map with their specific route highlighted, and the supervisor would have a color coded map of all of the bus driver's routes.