INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY

BANGALORE

Design Process and algorithm

Bus bunching problem

Version 1.0 approved

Prepared by:

Sanyam K Jain(MT2016122)

Raju Raghuwanshi(MT2016116)

Aman Bansal(MT2016007)

Team Guide:

Dr. S. Rajagopalan

Problem Statement :-

Over any length of time, buses serving a single route are likely to end up getting directly behind each other. Identical driving or constant customer demand and traffic congestion causes bunching problem. Due to bunching problem the vehicle departing at equal time intervals end up getting close to each other. The purpose of this project is to identify the bunching problem on certain route at a specific time, So that necessary measures could be taken to resolve it.

Algorithm for Bunching calculation :-

- User will enter the route number
- Return the list of bus stops with latitude, longitude and direction which belongs to mentioned route number
- Now user will select the direction and bus stop which internally contains latitude, longitude and route number and mention the date.
- **Input received from user :-**
    1. Route number
    2. Date
    3. Latitude and Longitude of bus stop in which we are checking bunching
    4. Direction in which we want to check bunching

- Use haversine formula to determine the bus records which are nearer to the specified bus stop with in the specified distance i.e 70-80 meter.

- **Calculate direction for fetched records :**

    1. Calculate the euclidean distance of bus with source and destination for the mentioned route.

    2. Check the calculated distance which is nearer to source or destination.

    3. Assign direction of bus based on the distance i.e calculated from first record (if the bus is nearer to source it means direction of the bus is from source to destination otherwise destination to source).

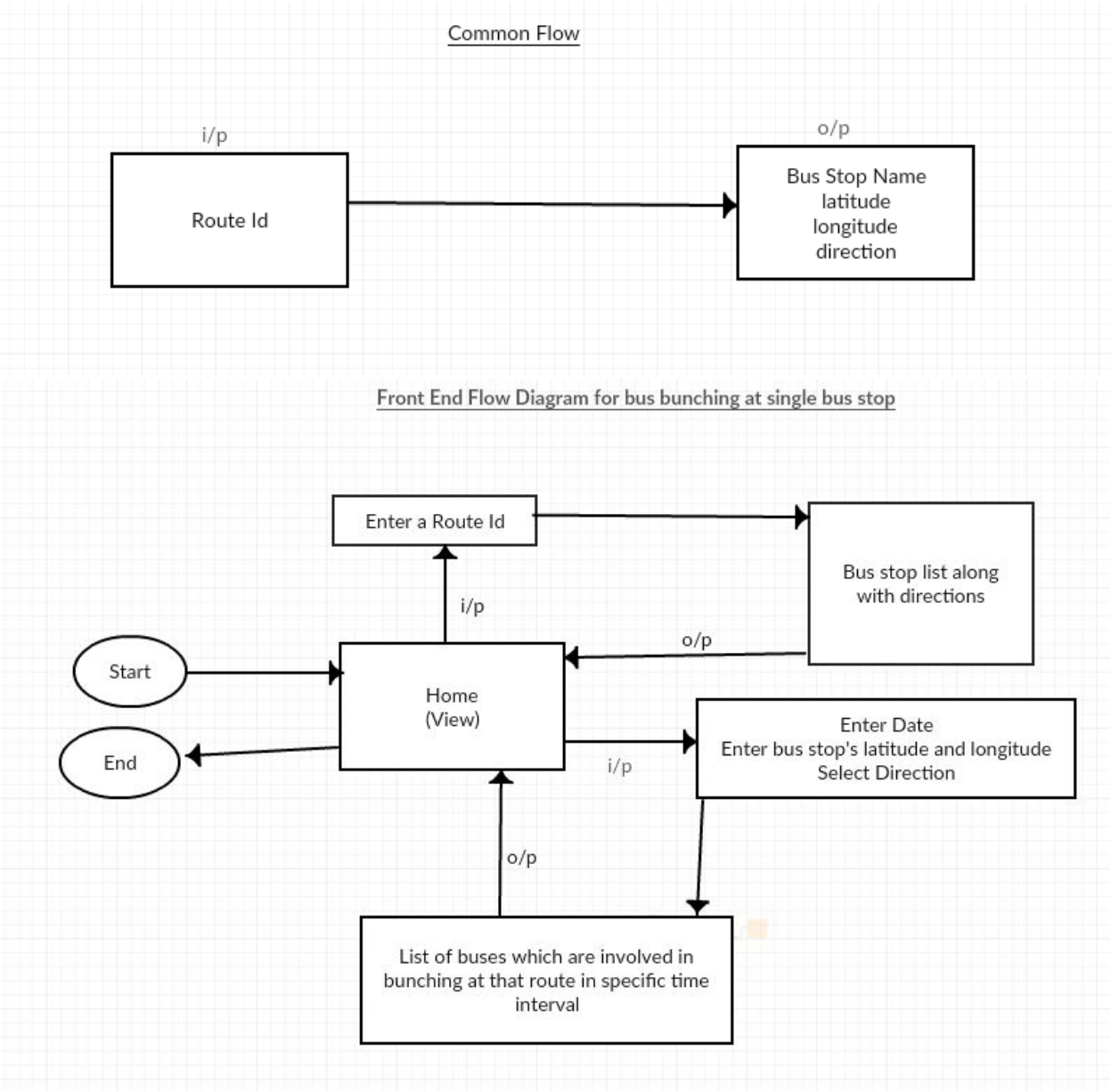
    4. Reverse the direction if bus reaches its destination i.e distance between the bus and destination becomes zero.

- To prevent the repetition of records when bus enters the range of bus stop we are using hash table.
- Make clusters of records subsequently which fall in 5-8 minute time frame.(suppose one bus reaches the bus stop at 8:07:05, we will check whether is there any record which fall between 8:07:05 to 8:14:05 if so then these group of buses result in bunching).

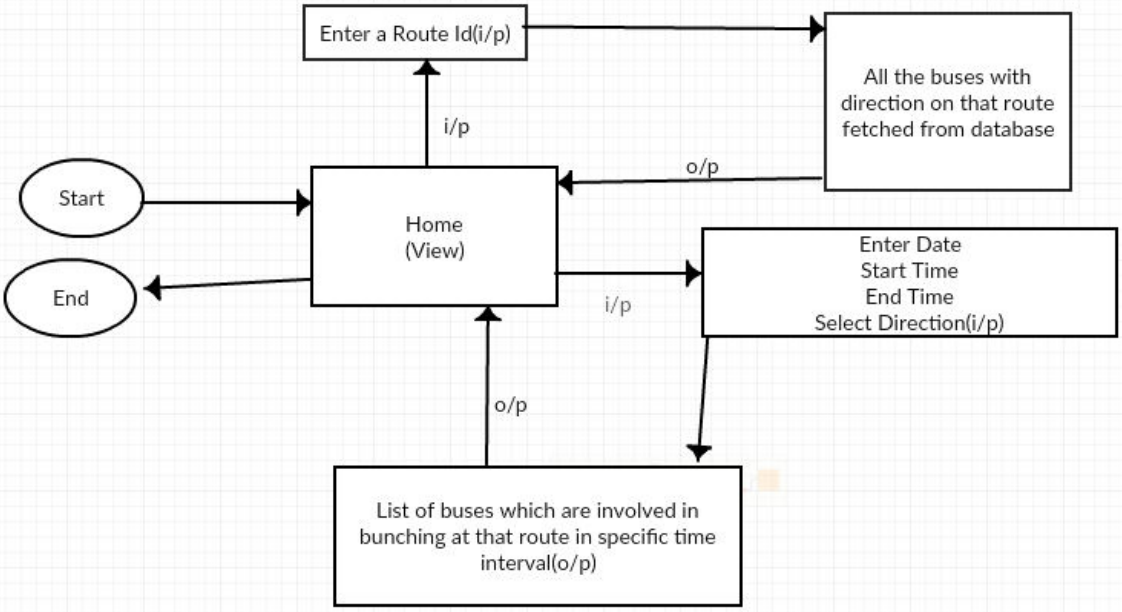
- Return the clusters of bus which holds the device_id, direction, latitude, longitude and time.

- All the obtained records will be shown on map to the user.

Flow Diagram:-



Front End Flow Diagram for entire route



Back End Flow Diagram