#### MTS Class Diagram - UML 1.4 Bus busId: int SimulationEngine name : String currentStopIndex: int coefficientSpeed: double nextStopIndex: int coefficientCapacity: double arrivalTime: int coefficientWaiting: double riderCount: int coefficientBuses: double passengerCapacity: int coefficientCombined: double speed : int eventHistoryItemList: List<String> fuel: int fuelCapacity: int changes: Map<String, String> addBus(bus : Bus) addStop(stop: Stop) getRoute(): Route addRoute(route: Route) applySpeedChanges() getBus(busId: int): Bus applyCapacityChanges() getStop(stopId: int): Stop getRouteChanges() : string getRoute(routeld : int) : Route addChange(changeType: string, changeDetail: string) moveBus() clearChanges() calcDistance(stop1: Stop, stop2: Stop): double calcTravelTime(stopFrom : Stop, stopTo : Stop, speed : int) calcSystemEfficiency(): double travel along rewindEvents(count : int) initFromFile(filePath : String) Route changeBusSpeed(busId : int, speed : int) changeBusCapacity(busId: int, capacity: int) routeld: int changeBusRoute(busId: int, routeId: int, stopIndex: int) number : int name: String addStop(stopId: int) getStopAtIndex(index : int):Stop getNextStopIndex(currentStopIndex : int) : int has Event EventQueue Stop time: int type : EventType stopId: int addEvent(event : Event) objectId: int name: String popEventWithLowestTime() : Event latitude : double updateEvent(objld: int, time: int) longitude : double waitingCount: int ridersArriveHigh: int ridersArriveLow: int ridersOffHigh: int ridersOffLow: int ridersOnHigh: int <<enumeration>> ridersOnLow: int EventType ridersDepartHigh: int ridersDepartLow: int move\_bus calculateRidersArrive(): int calculateRidersOff(): int calculateRidersOn(): int calculateRidersDepart(): int

### Jubin Jose | October 29, 2018

# Comments / Assumptions

- a. Access modifiers, getters, setters omitted.
- b. Constructors assumed.

### **Changes from implementation of Assignment 5**

- Removed Passenger from class diagram as we are only concerned with their count and not individual objects.
- Removed displayInfo() from Bus and Stop since it is out of scope for current requirement. Removed sortEvents and popEvent methods in EventQueue since not needed.

### Changs due to new requirements

# Stop class

-----

- a. Added low & high boundaries for passenger exchange distribution values.
- b. Added methods to calculate Rider count for Arrive, Off, On and Depart based on above boundaries.

## SimulationEngine class

- a. Added moveBus and calculateSystemEfficiency methods to handle client commands.
- b. Added properties for efficiency coefficients.
- c. Added initFromFile to initialize engine from input file.
- e. Added eventHistoryItemList to store a string of comma separated values representing state of bus and stop before an event is processed.
- f. Added rewindEvents(n) to handle client request for rewind of n events using the information from eventHistoryItemList
- g. Added changeBusSpeed, changeBusCapacity and changeBusRoute methods to accept user input for these actions

# Bus Class

a. Added riderCount and changeRoute.

- b. Removed updateDestination sicne stopIndex setters suffice
- c. Added a HashMap to store pending Bus Changes. Stored as <changeType, changeDetail> where changeType is speed, capacity or route and detail is a string with associated details of change
- d. Added addChange and clearChanges methods to add and clear above HashMap

#### EventQueue

-----

a. Added updateEvent to update tiem on ane xisting event. This is
needed because a rewind will need to rewrite time on existing event