M.Tech (IS)

Reasoning Systems Group Project

Intelligent Rapid Shuttle (IRS)



Team Members

SUDALAIANDI RAJA SUDALAIMUTHU

JAYARAMAN REVATHI

JAYASRI RAGHUNATHAN

SUNIL VARGHESE

# EXECUTIVE SUMMARY

In Singapore shuttle services to ferry students from their home to school and back is quite common. Probably every Singaporean would have travelled in a shuttle service during their student days.

Our team of 4 Singapore residents have also gone through the process of selecting shuttle services for our school going children and the biggest challenge we found was to find a service which takes the shortest distance to reach the school from our house at a reasonable cost.

We believe that many other Singapore parents would also be finding similar difficulties in selecting shuttle services for their children for a nominal cost. This provides a greater opportunity to address the gap in the market and thus we decided to embark on this project. In this project, we are assuming that we are a service provider having few depots operating shuttle buses at our disposal. We would be providing shuttle services optimized through the implementation of Reasoning Systems **Optimization** techniques using **KIE workbench**, **Drools** and **OptaPlanner**.

Key differentiator is the ability to service different schools by a shuttle service along the route for the students as part of the optimization outcome.

We felt that the scale and scope of the project was such that many more options for optimization could have been explored e.g. the amount of time taken for students to reach school taking traffic conditions in to consideration on top of travel distance optimized. However, given the time limitations we were happy with the solution we were able to achieve as part of this project.

Project Git Repository : <https://github.com/aivoyagers/IRS-RS-2019-03-09-IS1PT-GRP-aiVoyagers-irs-Intelligent-Rapid-Shuttle.git>

# BUSINESS PROBLEM BACKGROUND:

There is a demand for shuttle services which would ferry children from their home to the school and vice versa optimised to take the shortest distance in doing so. At the same time the prices charged for this service should be competitive compared to the current service providers. This can only be achieved by selecting optimized route along with correct distribution of students in to the available vehicles based on their starting locations and the schools.

# PROJECT OBJECTIVE

Having defined the business problem, our group’s aim was:

To develop an optimizer (Intelligent Rapid Shuttle - IRS) in which the route taken by a given shuttle should be the minimum distance for the students to reach their respective schools to minimize travel time and at the same time overall distance travelled should be less resulting in lower cost of operation for the service provider. This would make the services offered to the students for a lower shuttle service charge while servicing the students to minimize distance to travel to reach school. The optimizer would take the inputs as number of students, no. of Vehicles and no. of Schools to take the shortest route between a student location and their respective schools as well as the overall shortest route for a vehicle. This information will be pre-set from the dataset option selected by the users. Key feature supported would also include vehicle in operation can service multiple schools depending on students to be serviced. This would be a unique feature unlike other shuttle services which would typically operate for specific school. The routes would be shown visually on the google map to enable the service provider to navigate easily and follow the recommended route map by the IRS Optimizer.

# PROJECT SOLUTION

## BACKGROUND KNOWLEDGE AND USER INPUTS

For this project, below data/input are given/predefined:

User Input:

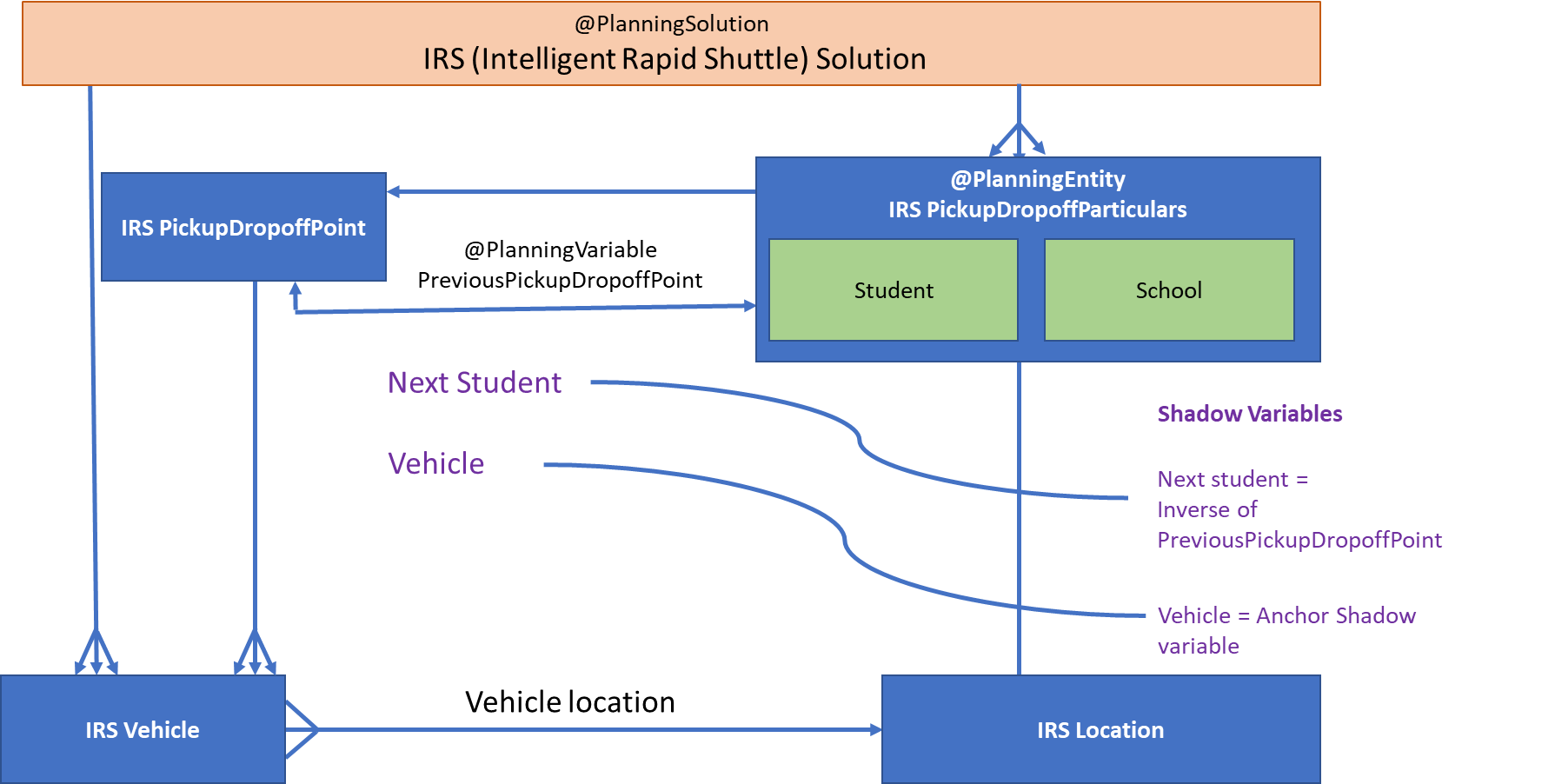
* The number of students
* The number of schools
* The number of vehicles

Data Collected from public sites using ParseHub web scrapping tool :

* A predefined list of schools and their locations (postal codes)
* A predefined list of student locations (postal codes)
* A predefined list of vehicle locations (postal codes)

## DOMAIN MODEL AND OPTIMIZATION VARIABLES

The UML class diagram is shown below. The planner concepts are already annotated.



* ***Planning Solution***: irsSolution Data Class contains the class that represents a data object and contains all planning entities. It includes list of Students, Schools and Vehicles as input to the Optimization Solver.

|  |  |
| --- | --- |
| **irsSolution** | **@PlanningSolution** |
| schoolList | List<irsPickupDropoffParticulars> |
| vehicleList | List<irsVehicle> |
| studentList | List<irsPickupDropoffParticulars> |
| score | HardSoftLongScore |

* ***Planning Entity & Planning Variable***: Data Class irsPickupDropoffParticulars is defined as Planning Entity. Students and School Objects are defined as irsPickupDropoffParticulars data class. List of attributes of this data class is shown below.

|  |  |
| --- | --- |
| **irsPickupDropoffParticulars** | ***@PlanninngEntity*** |
| dropOffLocationName | String |
| location | irsLocation |
| name | String |
| nextStudent | irsPickupDropoffParticulars ***(@Inverse Relation Shadow Variable)*** |
| vehicle | irsVehicle ***(@Anchor Shadow Variable)*** |
| prevPickupDropofffPoint | irsPickupDropoffPoint ***(@Planning Variable)*** |

irsPickupDropoffParticulars data class represents individual student or school information.

As input to solver, below attributes to be passed on as part of the API post call.

* + ***name*** : Name of the Student or School object
  + ***location*** : This object contains name of the location, geo location points (latitude, longitude) information.
  + ***dropoffLocationName*** : Travel destination location name, name of the school if it represents a Student object. Value of this attribute should match the corresponding school object’s name attribute. Should have NULL if the object represents a School.

Optimizer would assign values to below attributes response to GET API call to solver’s ‘bestsolution’ method.

* + ***prevPickupDropoffPoint*** : Reference pointer to previous pickup or dropoff point which could either be a vehicle starting location, student or school object. This attribute is the @***PlanningVariable*** defined with *Chaining* graph type to form the route path. Its ValueRange would be list of students, vehicles and school objects.
  + ***nestStudent***: Reference pointer to next pickup or dropoff point which could either be a student or school object. It is a @***InverseRelationShadowVariable*** to prevPickupDropoffPoint attribute to form the chaining of the route path.
  + ***vehicle*** : Reference pointer to vehicle object assigned by the optimizer to provide shuttle service to the student. Defined as @***AnchorShadowVariable*** to assign as a starting point of the route path chain.

***Anchor***

**School 1**

Student 2

*(->School 2)*

Student 1

*(->School 1)*

**Vehicle 1**

Student 3

*(->School 2)*

**School 2**

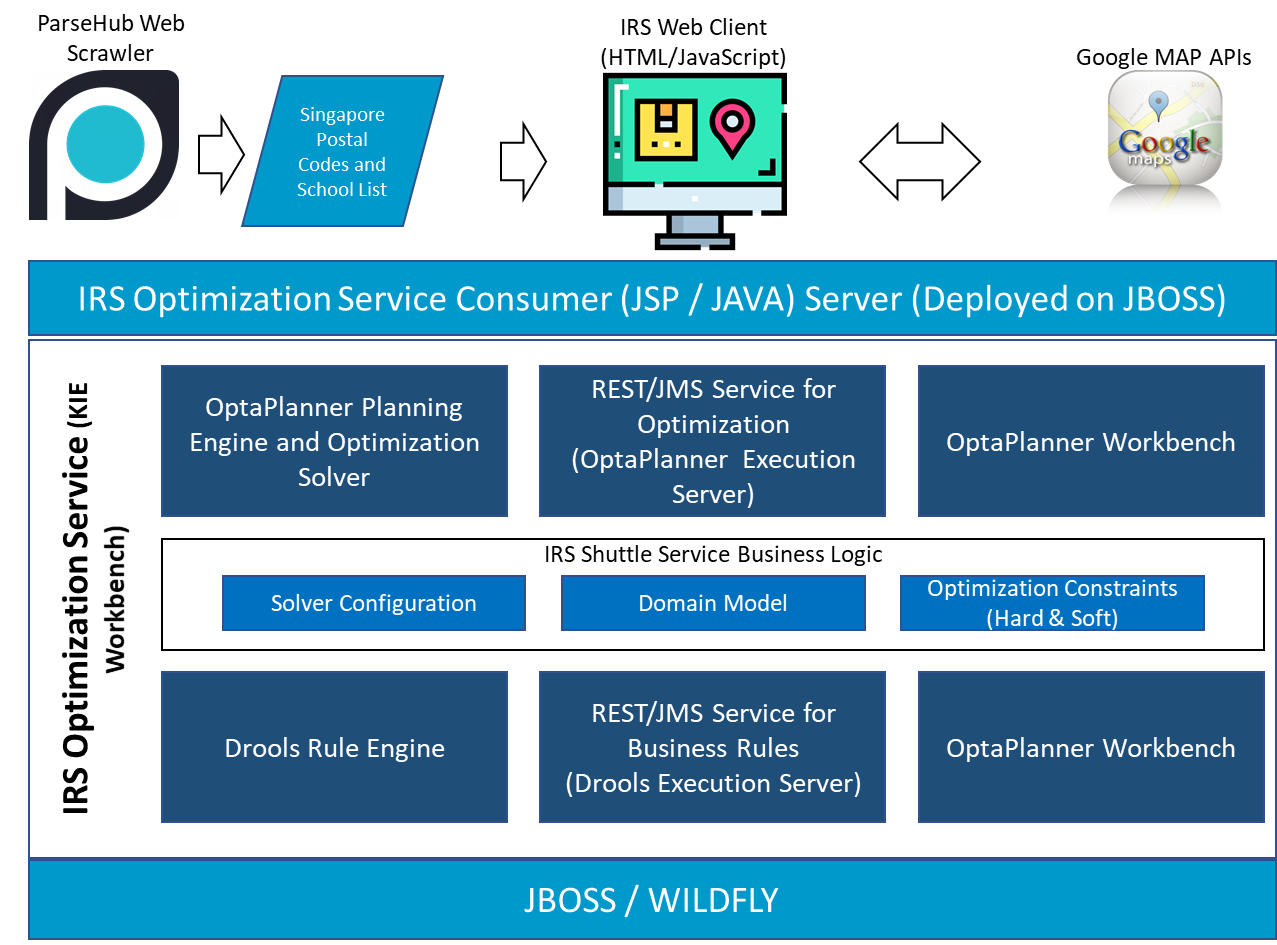
Student 4

*(->School 2)*

**Typical Route Path of IRS Shuttle Service**

## SYSTEM ARCHITECTURE

The system architecture diagram, illustrates how the application in the front-end has been interfaced with the Google MAP APIs, IRS Optimization Service Consumer and KIE Workbench Suite.



Below are the list of System components involved as part of this project.

* **ParseHub Web Scrawler** is used to collect sample list of locations with postal code information and list of Ministry of education (MOE) Schools in Singapore. This information is used to build input information to randomize location selection to assign pickup location and school location for student entity generated. There are around 20,000 postal codes and 173 school information collected from publicly available websites including MOE websites.
* **IRS Web Client** is built using HTML/JavaScript to interact with user to accept dataset selection to identify number of students, number of schools and number of vehicles. This input is used to form a simulated students list with their pickup location and drop off school location information along with number of vehicles to demonstrate the IRS Optimization solution.
* **Google Map APIs** is utilized by IRS Web Client to visually represent location detail of Students, Schools and Vehicles on Google Map using Google Directions APIs. Once IRS Optimization solution output obtained, Google APIs are again called to visually show the route path.
* **IRS Optimization Service Consumer** is a server component built using JSP / JAVA to integrate with IRS Shuttle Service Optimizer using REST API. This component registers IRS Solution Solver and invokes POST Solver method with information generated by IRS Web Client. Upon successful invocation of post solver method, recursively GET best solution outcome and pass it to Web Client to plot the recommended route path and the distribution of students in the vehicles as recommended by the IRS Solver.
* **IRS Optimization Service** is the core module implementing Hybrid Reasoning Systems techniques using KIE Workbench. Sub-components of this core service is described below.
* **OptaPlanner Planning engine and Optimization Solver** has been configured using OptaPlanner Workbench with Data Class defined for Planning Solution, Planning Entitiy, Planning Variable and their corresponding value range providers, etc., Domain Data Model in the previous section covers details on planning configurations.
* **Drools Rule Engine** is used to construct Hard and Soft Constraints required to meet IRS Shuttle service optimization solution. Below are the list of constraints defined using Drools Workbench.

**Hard Constraints:**

* irsVehicleCapacity constraint is defined using guided drools rule asset. This constraint ensures that the number of students on-boarded in the vehicle does not exceed the capacity of the vehicle.
* irsDropoffAfterPickup is a hard constraint defined using guided rule asset as well to construct the logic to enforce that the drop off point for every student is after the pickup point in the same vehicle.

**Soft Constraints:**

* distanceToSchoolSoftConstraint drools rule invokes a method to calculate Euclidean disttance for each student to reach their school via the route recommended by the Solver. This would enable the optimization engine to identify the optimal route for the student to reach their school destination.
* measureEuclideanDistance drools rule invokes a method to calculate Euclidean distance which is summarized across all route path. As the Optimization engine iterates planning variable ‘prevPickupDropoffpoint’ with assignment of value range providers ‘studentsRange’, ‘schoolRange’ and ‘vehicleRange’.
* distanceFromLastDropoffPoint DRL rule measures the Euclidean distance from the last drop off point to the vehicle starting location which is typically Shuttle Service main or branch office location.
* **JBOSS / Wildfly Application Server** is used to deploy KIE execution server and IRS Optimization Service Consumer components. JBOSS/Wildfly is leveraged as part of KIE Workbench.

# PROJECT SCOPE

Project scope is to develop, integrate and demonstrate at least two Hybrid Reasoning Systems techniques by delivering a standalone bespoke application. Objective is to demonstrate understanding of the concepts learnt in Hybrid Reasoning Systems module as part of M.Tech (Intelligent Systems) graduate programme.

IRS Intelligent Rapid Shuttle project scope covers following minimum viable product features.

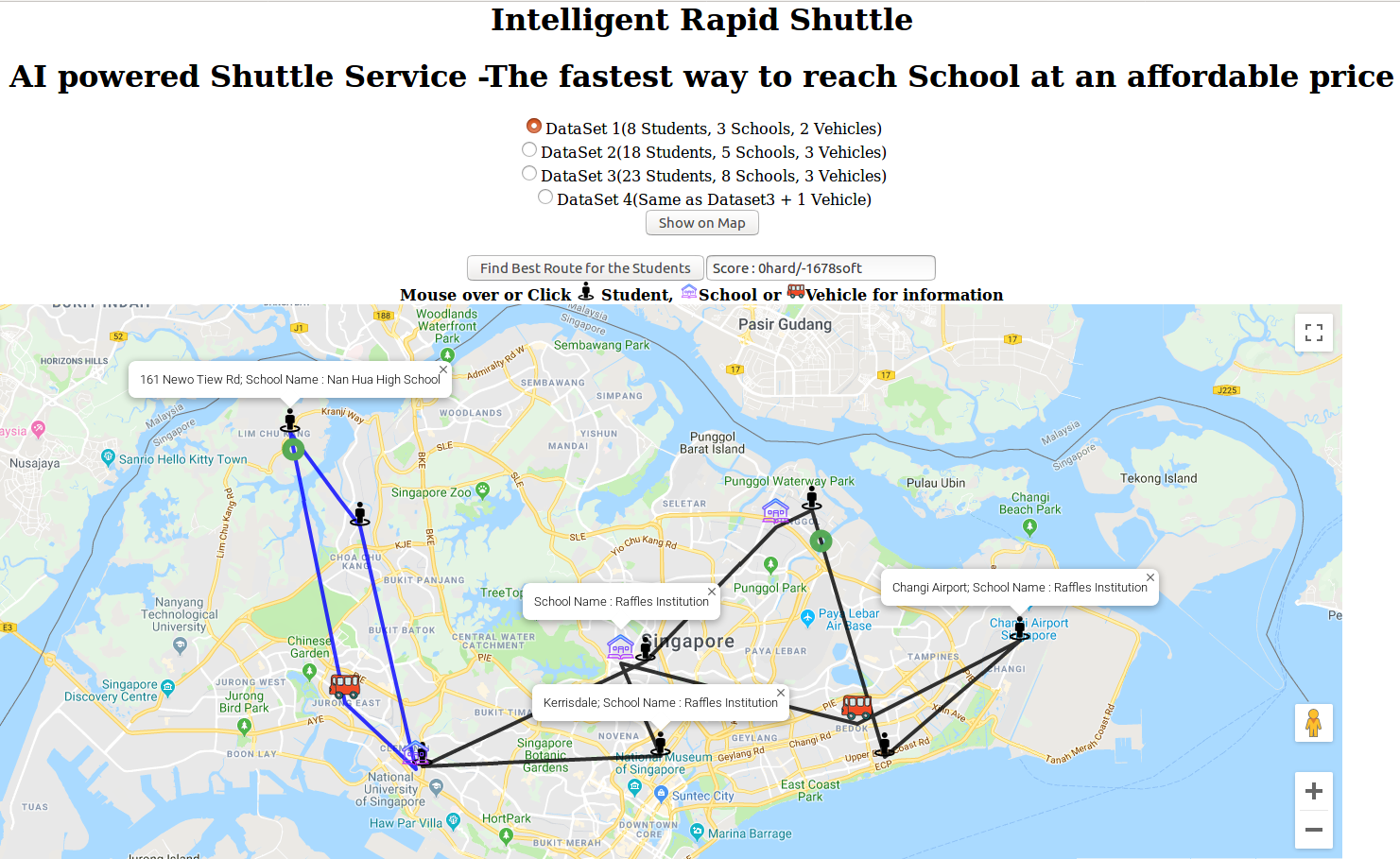
* Develop a web user interface to accept user input to identify number of students, number of schools and number of vehicles to be part of Optimization Solution.
* Visual representation of location information for the students, schools and vehicles location on google map using google directions APIs.
* Integrate with IRS Optimization Service to initiate the optimization solver, post solution input to the solver and get the optimized route path and distribution of students among available vehicles to achieve optimization goal.
* Develop IRS optimization service to consutruct domain models representing the solution details and implement hard and soft constraints to achieve business logic required.
* Configure to achieve optimimal solution output to meet the problem statement stated in this project.
* Visually represent optimization results using Google maps.

# SYSTEM’S FEATURES

Despite the limitations, the team went through an in-depth thought process to implement significant features in the Intelligent Rapid Shuttle system which can substantially add value to potential value to offer an AI powered optimized route path to achieve better service to customers at the same time operating at a lower cost with minimum possible total distance.

* Key differentiator is the ability to pick up students from various locations and service multiple schools at the same time. Same vehicle can service different schools optimizing vehicle capacity while achieving optimal travel distance for the students.
* The system optimizes the routes considering the student’s location to their respective destination (school) across all the students to be serviced and the available vehicles at disposal. This helps the students to reach school faster.
* The overall distance travelled by all the vehicles operating for the service provider is optimized such that the least distance is travelled. This would help the service provider to minimize the cost and hence price the services competitively.
* The soft score computed includes:
* Distance factor for each student from their starting location to to their respective school
* Distance factor for overall distance travelled by all vehicles available for service
* The capacity of the vehicles is optimized dynamically, such that at any given point of time capacity is maximized without crossing the capacity limit. This in turn helps in maximising the profit for the service provider.
* The system is designed for scalability, i.e. it caters for multiple vehicles with multiple locations as well as multiple student locations. The system is catered to have these inputs scaled up.

## IRS Solution – Sample Screen for the Data Input provided in Section 6.2 Sample Input



## IRS Solution – Sample Input

Input data consists of List of Vehicles, List of Schools and List of Students with their starting location and destination (school) details. Sample input is provided below.

|  |  |  |  |
| --- | --- | --- | --- |
| **Vehicle Name** | **Starting Location-Latitude** | **Starting Location-Longitude** | **Vehicle Capacity** |
| Vehicle 1 from Bedok Mall | 1.324944 | 103.929392 | 10 |
| Vehicle 2 from Jurong JEM Mall | 1.332787 | 103.743121 | 4 |

|  |  |  |
| --- | --- | --- |
| **Name of the School** | **School Location - Latitude** | **School Location - Longitude** |
| Raffles Institution | 1.346979 | 103.843417 |
| Nan Hua High School | 1.308489 | 103.768901 |
| Compassvale Secondary School | 1.396588 | 103.899726 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Student Name** | **Location Name** | **Location - Latitude** | **Location Longitude** | **Dropoff School Name** |
| Student of Bishan Loft | Bishan Loft Block 31 | 1.347593 | 103.852434 | Raffles Institution |
| Student of Yew Tee | Block 632 Yew Tee | 1.396750 | 103.748564 | Nan Hua High School |
| Student of Changi Airport | Changi Airport | 1.355246 | 103.988475 | Raffles Institution |
| Student of Neo Tiew Rd | 161 Newo Tiew Rd | 1.430672 | 103.723253 | Nan Hua High School |
| Student of Oassis Terraces | Oasis Terraces | 1.402721 | 103.912795 | Compassvale Secondary School |
| Student of Bayshore | 34 Bayshore Road | 1.313012 | 103.939276 | Raffles Institution |
| Student of Clementi Ave 1 | 425 Clementi Ave 1 | 1.309749 | 103.771281 | Raffles Institution |
| Student of Karissdale | Kerrisdale | 1.313492 | 103.857975 | Raffles Institution |

## IRS Solution – Sample Output

Output from Optimization solution engine provided below for the sample input shared in the previous section. Outcome has distributed the students among the 2 available vehicles specified and route path to pickup students and drop them off in their schools. After dropping all the students to their destination, vehicle goes back to the starting location.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Optimized Route** | **Pickup / Dropoff** | **Student Name** | **Location - Latitude** | **Location Longitude** | **Dropoff Detail** |
| **1** |  | **Vehicle 1 from Bedok Mall** | **1.324944** | **103.929392** | **Capacity 10** |
| **2** | Pickup Student | Student of Bayshore | 1.313012 | 103.939276 | Raffles Institution |
| **3** | Pickup Student | Student of Changi Airport | 1.355246 | 103.988475 | Raffles Institution |
| **4** | Pickup Student | Student of Oassis Terraces | 1.402721 | 103.912795 | Compassvale Secondary School |
| **5** | Dropoff at School | Compassvale Secondary School | 1.396588 | 103.899726 | **Student of Oassis Terraces** |
| **6** | Pickup Student | Student of Bishan Loft | 1.347593 | 103.852434 | Raffles Institution |
| **7** | Pickup Student | Student of Clementi Ave 1 | 1.309749 | 103.771281 | Raffles Institution |
| **8** | Pickup Student | Student of Karissdale | 1.313492 | 103.857975 | Raffles Institution |
| **9** | Dropoff at School | Raffles Institution | 1.346979 | 103.843417 | **Student of Changi Airport, Student of Bishan Loft, Student of Clementi Ave 1, Student of Karissdale** |
| **10** |  | **Vehicle 1 from Bedok Mall** | 1.324944 | 103.929392 |  |
|  |  |  |  |  |  |
| **1** |  | **Vehicle 2 from Jurong JEM Mal** | **1.332787** | **103.743121** | **Capacity 4** |
| **2** | Pickup Student | Student of Neo Tiew Rd | 1.430672 | 103.723253 | Nan Hua High School |
| **3** | Pickup Student | Student of Yew Tee | 1.396750 | 103.748564 | Nan Hua High School |
| **4** | Dropoff at School | Nan Hua High School | 1.308489 | 103.768901 | **Student of Bishan Loft, Student of Karissdale** |
| **5** |  | **Vehicle 2 from Jurong JEM Mal** | 1.332787 | 103.743121 |  |

# LIMITATIONS

System consolidates all the students going to a particular school in a single vehicle though they can be distributed across multiple vehicles to achieve better route path and travel distance. This limitation is due to lack of value range specification feature in OptaPlanner for repeatable optional school value range. This can be solved by providing multiple school entity if the number of students exceed certain value or their locations are distributed beyond certain distance among them.

# CONCLUSION

Our team had a great time working on this project, and we definitely picked up some useful skills along the way. Understanding the configuration of OptaPlanner, constraint development using Drools and class diagram design of the solution with integration of web interface using JavaScript / JSP/ Java was part of the entire process. Without a sound knowledge base taught in the lectures, we wouldn’t have been able to build on system based on all the different rules. Building the system itself presented a whole new set of learning points. We got to apply practical knowledge of the KIE Server, OptaPlanner, as well as tap on our team’s existing expertise in Java and extensive experience in design and delivering enterprise application systems. Working on the exercise together allowed everyone to learn some or the other skills from one another.

# IMPROVEMENTS

Following are the improvements noted while executing this project.

On top of overall distance travelled as an optimization goal to achieve, below can be added to provide a better service to the customers by the Shuttle Service provider.

* Consider traffic conditions (travel time at student level) across the route path chosen to achieve better travel time. This would also improve trust that Service Provider takes in to account of ensuring students reaching their school on time. While this feature would provide better service, there could be an impact to overall travel distance travelled for the service provider affecting operating expenses. This can be balanced by assigning appropriate weights while calculating soft score.
* Address the limitation highlighted in Section 7 LIMITATIONS

Annex1

Class Data Model(excel)



Sample Postman Collection file attached for API interface to IRS Optimization Service running on KIE Workbench Execution Server.



Sample Input to IRS Optimization Service (OptaPlannner Execution Server) (API/XML example)



Sample Output of IRS Optimization Service (OptaPlannner Execution Server) (API/XML example)

