

SLIPPERY ROCK UNIVERSITY

Dan Miller, Jordan Schiller, Zachar
Petrusch, Christopher Solomon

Technical Manual for Multi Modal Routing Problem

CONTENTS

System Requirements.....	3
Assumptions	3
Database	3
Database Tables	3
Bike	3
Cargo Ship.....	4
Carriers	4
Truck	5
Plane	5
Rail	5
Location	5
Segment.....	6
Shipment	6
Shipment History	7
Shipping Rates	8
Travel Types.....	8
Vehicle Travel Type Index.....	9
Database Selection	9
MMRP Core Classes	9
MMRP Routing Classes	10
Base Class and Database Interactions	11
API.....	12
UML Diagrams of MMRP	12
Use-Case Diagrams	12
Class Structure	14
Database Diagram	15
Glossary	16

MMRP Core	16
Base Class	16
Bike	16
Cargo.....	16
Carrier	16
Location	18
Plane	18
Rail	18
Segment.....	19
Shipment	20
ShipmentHistory	23
Shipper.....	23
ShippingRate.....	24
TravelType	25
Truck	26
Routing	26
RoutingAlgorithm	26
WeightedMetric	26

SYSTEM REQUIREMENTS

The following is a list of required hardware and software required to run the Multi Modal Routing Problem [MMRP] program.

- Intel® Pentium® IV or compatible
- Microsoft® Windows® XP or Microsoft® Windows® 7
- MySQL Workbench 6.0 CE or higher
- MySQL Community Server Version 5.6.12 or higher
- Eclipse IDE for Java EE Developers Kepler Edition Version 4.3.2 or higher

For instructions on how to run MMRP please see the User Manual.

Assumptions

It has been assumed that the hardware mentioned in the system requirements is in proper running condition. Further documentation will be supplied in subsequent documents¹ on the steps required to install the MySQL Community Server, MySQL Workbench and Eclipse JDK.

DATABASE

The project requires access to the Excalibur Server at Slippery Rock University at this time. That is where the database is located. In order to speed up access times if working a remotely a user may select to keep a local copy of the database. This manual does not cover that.

Database Tables

This section covers the different tables in the database as well as providing a description of each field. For information concerning the type of each field see the glossary. The tables are presented in alphabetical order as that is how they are stored in the database.

Bike

This is the Bike table. Bike is one of the mode of transportation that can be used to get a Shipment from one Location to the next. Bike types are not currently supported in the Graphical User Interface [GUI] but the support on the backend is ready should the project ever decide to incorporate Bikes as a mode of transportation for a shipment. Bike contains the following fields:

- Bike Id — This is the unique ID of the Bike
- Bike Name — This is the given name of the Bike
- Carrier — This is the ID of the Carrier that supplies the Bike
- Status — This is the current status of the Bike

¹ This information is located in the User Manual.

Cargo Ship

This is the Cargo table. Cargo is one of the mode of transportation that can be used to get a Shipment from one Location to the next. Cargo contains the following fields:

- Ship Id — This is the unique ID of the Cargo
- Ship Name — This is the given name of the Cargo
- Carrier — This is the ID of the Carrier that supplies the Cargo
- Status — This is the current status of the Cargo

Carriers

This is the Carriers table. The Carrier is the supplier for Vehicles that will be used for transporting Shipments. Carrier contains the following fields:

- Carrier Id — The unique ID of the Carrier
- Carrier Code — A code² for the Carrier
- Carrier Name — The name of the Carrier company
- Cost Modifier³ Truck — Modifier for shipping something by Truck with this company
- Cost Modifier Cargo Ship — Modifier for shipping something by Cargo with this company
- Cost Modifier Bike — Modifier for shipping something by Bike with this company
- Cost Modifier Rail — Modifier for shipping something by Rail with this company
- Cost Modifier Plane — Modifier for shipping something by Plane with this company
- Send By Fax — Boolean indicating if the Carrier can be reached by fax
- Send by Email — Boolean indicating if the Carrier can be reached by email
- Area Code — Area code of the Carrier
- Fax Number — Fax number of the Carrier
- Email Address —Email address of the Carrier
- Safety Rating — Safety rating value of the Carrier
- Safety Rate Date — Date that the Carrier received their safety rating
- Authorize — This is a field carried over from Zeus, see Zeus documentation

² An example: The Carrier Boyd Brothers has the code BODB.

³ Cost Modifiers are used because we do not have real data for the Carriers. We created cost modifiers such as Boyd Brothers costs 1.4 times more than standard for this particular type.

- Contract Date — This is a field carried over from Zeus, see Zeus documentation
- Ins End Date — This is a field carried over from Zeus, see Zeus documentation

Truck

This is the Truck table. Truck is one of the mode of transportation that can be used to get a Shipment from one Location to the next. Truck contains the following fields:

- Truck Id — This is the unique ID of the Truck
- Truck Name — This is the given name of the Truck
- Carrier — This is the ID of the Carrier that supplies the Truck
- Status — This is the current status of the Truck

Plane

This is the Plane table. Plane is one of the mode of transportation that can be used to get a Shipment from one Location to the next. Plane contains the following fields:

- Plane Id — This is the unique ID of the Plane
- Plane Name — This is the given name of the Plane
- Carrier — This is the ID of the Carrier that supplies the Plane
- Status — This is the current status of the Plane

Rail

This is the Rail table. Rail is one of the mode of transportation that can be used to get a Shipment from one Location to the next. Rail contains the following fields:

- Rail Id — This is the unique ID of the Rail
- Rail Name — This is the given name of the Rail
- Carrier — This is the ID of the Carrier that supplies the Rail
- Status — This is the current status of the Rail

Location

This is the Location table. A Location is a real world place that can be visited by the various modes of transportation available to the Shipments.

- Location Id — This is the unique ID of the Location
- Longitude — Longitude of the Location
- Latitude — Latitude of the Location
- Name — Name of the Location
- Travel Type 1 — Travel Type available at Location
- Travel Type 2 — Travel Type available at Location

- Travel Type 3 — Travel Type available at Location
- Travel Type 4 — Travel Type available at Location
- Travel Type 5 — Travel Type available at Location
- Travel Type 6 — Travel Type available at Location
- State — The state of the Location
- Country — The Country of the Location

Segment

This is the Segment table. The Segment is a single component that will make up the route that a Shipment will take to get from the start Location to the end Location. The Segment is linked to the Vehicle that is being used over this section of the route.

- Segment Id — This is the unique ID of the Segment
- From Location Id — Id of the starting Location
- To Location Id — Id of the ending Location
- Vehicle Id — Id of the Vehicle using Segment
- Mode Type — Type of vehicle using segment
- Time of Departure — Estimated time vehicle will leave starting Location
- Time of Arrival — Estimated time vehicle will arrive at Destination
- Lane — This is a field carried over from Zeus, see Zeus documentation
- Shipping Rate Id — Id of the Shipping Rate used on the Segment
- Earliest Arrival Time — Earliest time Vehicle can arrive at destination
- Latest Arrival Time — Latest time Vehicle can arrive at destination
- Earliest Departure Time — Earliest time Vehicle can depart start Location
- Latest Departure Time — Latest time Vehicle can depart start Location

Shipment

This is the Shipment table. The Shipment is the package that the needs to be transported from the start Location to end Location via Vehicles supplied by the Carriers, to get the Shipment from the Shipper to the Customer.

- Shipment Id — This is the unique Id of the Shipment
- From Location Id — Id of the starting Location
- To Location Id — Id of the ending Location

- Priority — Priority of the Shipment, the higher the priority, the more urgent the Shipment.
- Earliest Departure From Start — Earliest time Shipment can depart start Location
- Latest Departure From Start — Latest time Shipment can depart start Location
- Earliest Departure Date — Earliest calendar date shipment can leave start Location
- Latest Departure— Latest calendar date shipment can leave start Location
- Earliest Arrival — Earliest time shipment can arrive at end Location
- Earliest Arrival Date — Earliest calendar date shipment can arrive at end Location
- Latest Arrival — Latest time Shipment can arrive at end Location
- Latest Arrival Date — Latest calendar date that shipment can arrive at end Location
- Size — Volumetric size of the Shipment
- Weight — Weight of Shipment
- Current Location — Id of Shipment’s current Location
- Loading Time — Average time it will take to load Shipment
- Unloading Time — Average time it will take to unload Shipment
- Loading Rate — Relative speed at which a Shipment can be loaded or unloaded
- Shipper — Id of the Shipper sending the Shipment
- Take Toll Roads — Boolean indicating Shipment can use toll roads
- Local Congestion By Pass — Boolean indicating if Shipment may attempt to bypass congested Segments
- Trailer Type - This is a field carried over from Zeus, see Zeus documentation
- Loading Type - This is a field carried over from Zeus, see Zeus documentation
- Unloading Type - This is a field carried over from Zeus, see Zeus documentation
- Hazmat Constraints — Considerations relating to hazardous material handling precautions that will be needed with the Shipment
- Pref Contractors — Carriers preferred for use on the Shipment
- Pref Lanes — Preferred Lanes that will be used by the Shipment
- Max Stops — Maximum number of loading and unloading stops the Shipment is

Shipment History

This is the Shipment History table. The Shipment History is a record of the path that the Shipment took to get to the current Location.

- Shipment History Id — This is the unique Id of the Shipment History
- Segment Id – Id of Segment that the Shipment has traveled
- Shipment Id – Id of Shipment that is related to Shipment History
- Node Number – Order segments were traversed for the Shipment

Shipping Rates

This is the Shipping Rates table. A Shipping Rate is the basic payment information needed to determine how much should be charged for transporting a Shipment over a given Segment using a certain Carrier.

- Shipping Rate Id – Unique Id of the Shipping Rate
- Carrier Id – Id of the Carrier using the Shipping Rate
- Start Location – Location Shipping Rate starts at
- End Location – Location Shipping Rate ends at
- Travel Type – Id of the Travel Type that applies to this Shipping Rate
- Weight 1 – Weight classification for which Rate 1 applies
- Rate 1 – Fee for transporting a Shipment of Weight 1 or less
- Weight 2 – Weight classification for which Rate 2 applies
- Rate 2 – Fee for transporting a Shipment of Weight 2 or less
- Weight 3 – Weight classification for which Rate 3 applies
- Rate 3 – Fee for transporting a Shipment of Weight 3 or less
- Mile Rate – Per mile rate
- Flat Rate – Flat Rate fee
- Rank - This is a field carried over from Zeus, see Zeus documentation

Travel Types

This is the Travel Types table. A Travel Type is a subtype of the separate Vehicle options to give more detailed information about transportation of the Shipment using the available Vehicles.

- Vehicle Type Id – Unique Id for the Travel Type
- Vehicle Mode – Mode of transportation
- Trailer 1 – Specifications of first trailer
- Trailer 2 – Specifications of second trailer
- Minimum Capacity – Minimum capacity that must be filled in order to allow Travel Type to leave current Location

- Maximum Capacity – Maximum capacity that travel type can handle
- Actual Capacity – Current available capacity of the travel type
- Service Type - This is a field carried over from Zeus, see Zeus documentation
- Radiation – Boolean indicating if Travel Type can handle radioactive Shipments
- Refrigeration – Boolean indicating if Travel Type can handle refrigerated Shipments
- Hazardous Material – Boolean indicating if Travel Type can Shipment containing hazardous materials
- Explosive Materials – Boolean indicating if Travel Type can handle explosive Shipments
- Tracking – Boolean indication if Travel Type can be tracked

Vehicle Travel Type Index

This is the Vehicle Travel Type Index table. A Vehicle Travel Type Index is a look up table that shows what Travel Types are available for which Vehicles.

- Vehicle Travel Type Index Id – Unique Id for the Vehicle Travel Type Index
- Vehicle Id – Id of the Vehicle
- Travel Type Id – Id of the Travel Type
- Travel Mode – Mode of transportation

Database Selection

Optionally, another database can be selected to be used as the source of information for the MMRP project. This selected database must match the tables and fields above for the MMRP to perform correctly without modifications to the code. This selection can be made via the databaseConfiguration.txt file.

MMRP CORE CLASSES

These are the classes that make up the core package of MMRP

- **BaseClass.java** – This acts as a wrapper parent class for all the subclasses that need access to the database.
- **Bike.java** – This is one of the Vehicle subclasses that can be used to transport a Shipment – analogous to the database table.
- **Cargo.java** – This is one of the Vehicle subclasses that can be used to transport a Shipment – analogous to the database table.
- **Carrier.java** – This is the owner of the Vehicles that transports Shipments, and the party that sets particular ShippingRates – analogous to the database table.

- **FormatChecker.java** – This class handles the setting of new data into the classes. It acts as a valid input checker, formatter and range checking software to ensure no bad data get into the classes or database.
- **Location.java** – This is the class that holds data about real world locations that a Shipment can reach – analogous to the database table
- **Plane.java** – This is one of the Vehicle subclasses that can be used to transport a Shipment – analogous to the database table.
- **Rail.java** – This is one of the Vehicle subclasses that can be used to transport a Shipment – analogous to the database table.
- **Segment.java** – This is the basic component of a solution. One or more Segments provide the all the information needed to trace the route a Shipment will take from its start Location to the end Location – analogous to the database table.
- **Shipment.java** – This is the package that the Shipper wants to get to the Customer – analogous to the database table.
- **ShipmentHistory.java** – This a a Segment/Shipment pairing that indicates the Segment that a Shipment had to travel along in order to reach its destination – analogous to the database table.
- **Shipper.java** – This is the party that sends the Shipment via Carriers to the Customer – analogous to the database table.
- **ShippingRate.java** – This is the way a Carrier charges for various Shipment weights and sizes over a particular Segment that the Shipment will travel along – analogous to the database table.
- **TravelType.java** – This is the specifications of the Vehicle that is traveling along a Segment. This object contains the important details about the capacity and regulations for the individual Vehicle using a Segment – analogous to the database table.
- **Truck.java** – This is one of the Vehicle subclasses that can be used to transport a Shipment – analogous to the database table.
- **Vehicle.java** – This is the abstract base class of all the modes of transportation that the Shipment can take, or are offered by the various Carriers.

MMRP ROUTING CLASSES

- **AStarAlg.java** – Classic A* Algorithm applied to this problem set. Varied that it takes a weight when traversing the search space, using normalized values.
 - **AStarNode.java** – This object servers as the nodes used by the A Star Algorithm.

- **BestFirstFind.java** – This algorithm does the following :
 - If direct path is available choose the best one
 - If direct path not available choose the lowest weighted cost Segment from this Location.
 - Repeat until the end is reached or time is up
- **NextAvailableVehicle.java** – This algorithm does the following:
 - If direct path is available choose the next vehicle leaving and take it
 - If direct path is not available choose the next vehicle leaving from all the available Segments at this Location.
 - Repeat until end is reached or time is up
- **NodeCrawler.java** – This algorithm does the following:
 - If a random number is greater than the percent chance to find a direct route, then try to find a direct route from this Location to the end. If there are Segments available then randomly choose one.
 - If no direct route is found, or we do not randomly attempt a direct path, then we will randomly choose from all the available Segments to travel along.
 - Repeat until end is reached or time is up.
- **TravelByType.java** – This algorithm does the following:
 - If a random number is greater than the percent chance to find a direct route, then try to find a direct route using the given mode of travel. Choose the lowest weighted cost Segment to travel to the end.
 - If there is no direct route, or the direct route is not randomly chosen, then choose the lowest weighted cost Segment that uses the given mode of travel.
 - Repeat until the end is reach of time is up.
- **WeightedMetric.java** – This is the way that traveling along a particular Segment is measured. The user defines the importance of minimizing time, cost and distance related to each other and the result is a weighted cost function to determine the lowest cost Segments the algorithms will use to find a solution.

BASE CLASS AND DATABASE INTERACTIONS

Any classes that derive from the Base Class will have Booleans indicating if the object instantiated is new and will need to be inserted into the database or dirty, i.e. changed data, and needs to be updated in the database.

The children of Base Class will have to implement a Delete and Upload function that will interact with the database and map correctly to the analogous class table in the database.

API

The application developers chose to put a normal API into the javadoc ability. There are java docs for each function. That resources should be the one referenced if needed.

UML DIAGRAMS OF MMRP

The following UML diagrams will help better explain the functionality of MMRP System as well as the interactions between the various agents.

Use-Case Diagrams

There are two use cases associated with MMRP. Figure 1 models the low-level interaction between the GUI and the database.

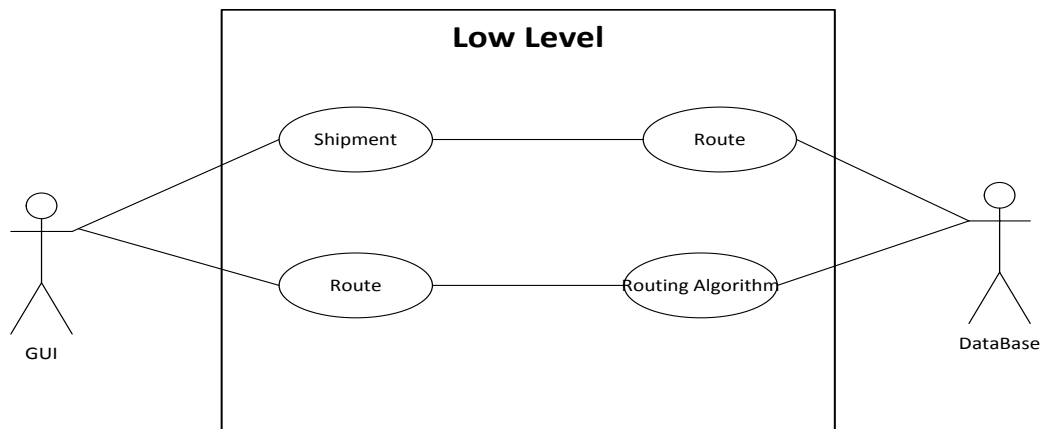


Figure 1 – Low Level UML

Figure 2 represents interaction between the user and the low level side of the program. Mainly showing the intended functionality of the GUI.

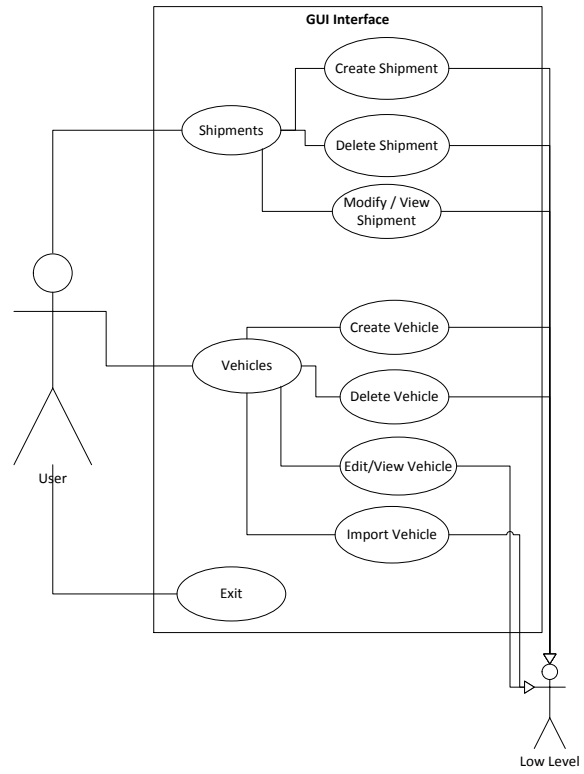


Figure 2

Figure 3 models the class structure of the program. This only models the object found in the core of the application.



Database Diagram

Figure 4 models the database where the information is stored. This database was described in full in earlier sections.

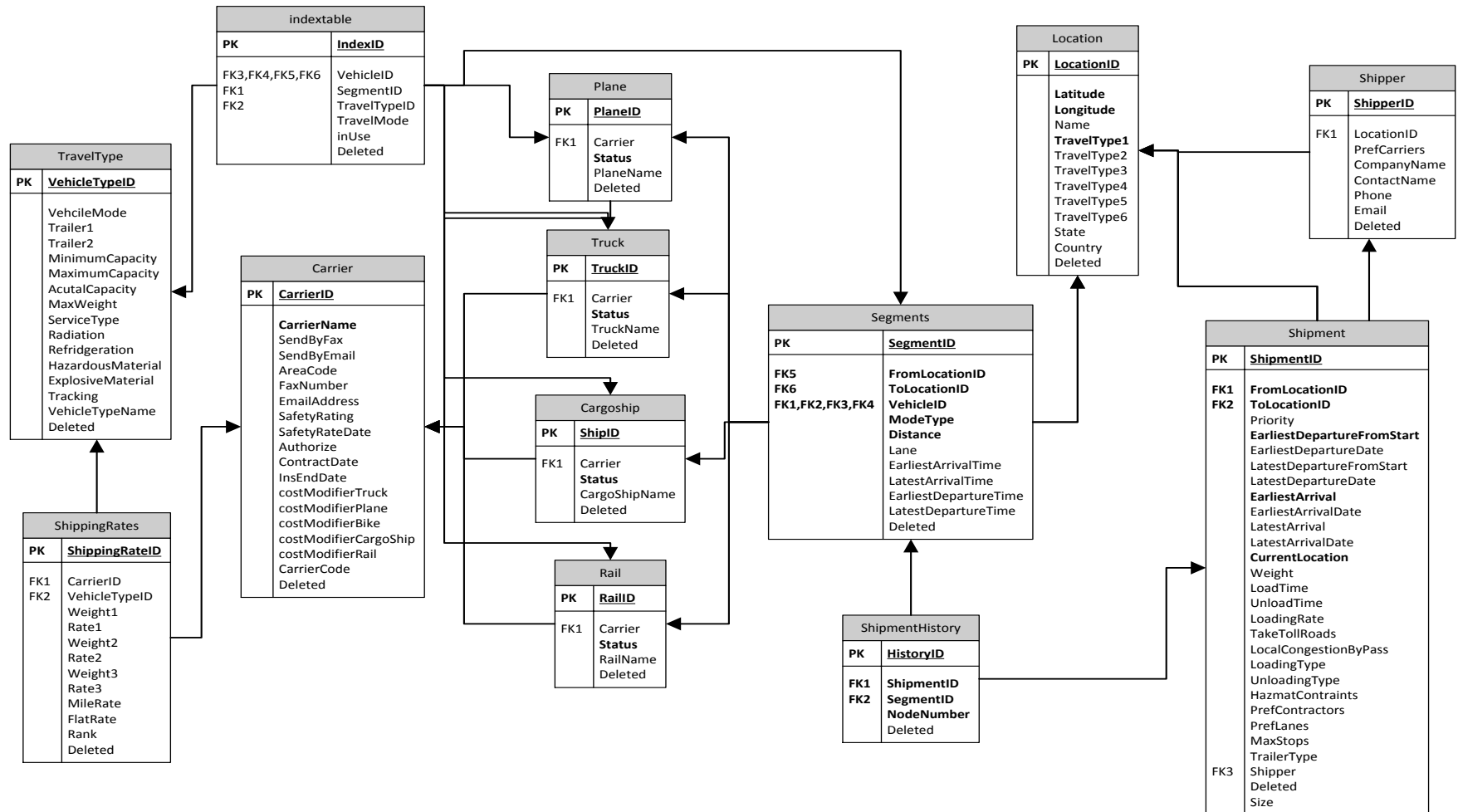


Figure 4

GLOSSARY

The purpose of the Glossary is to explain the attributes of objects.

MMRP Core

Base Class

dirty : Boolean – This Boolean marks if changes have been made to a derived object that will require an update to the database to insure the cohesiveness of data across the system.

newObject : Boolean - This Boolean marks that a derived object is new and needs to be added to the database.

DRIVER : String - This is the string that locates the driver that is needed for the connection to the database.

URL : String - This is the address that is used to connect to the database.

PSWD : String - This is the password that is needed to connect to the database.

connectionCounter : int -This is a count of the number of concurrent connections opened to the database, used mostly for debugging and exception handling.

Bike

DEFAULT_BIKE_NAME : String -This is the default bike name, if no other name is set, this is the name that will be chosen.

Cargo

DEFAULT_CARGO_NAME : String -This is the default cargo name, if no other name is set, this is the name that will be chosen.

Carrier

id : int - This is the unique id of the Carrier object given to the object from the database.

carrierCode : String -This is the code for the Carrier, for faster identification.

carrierName : String -This is the name of the Carrier.

sendByFax : Boolean – This Boolean indicates if the Carrier can be reached by fax.

sendByEmail : Boolean – This Boolean indicates if the Carrier can be reached by email.

areaCode : String – This is the area code of the Carrier's fax number.

faxNumber : String – This is the fax number the Carrier can be reached at.

emailAddress : String – This is the email address that the Carrier can be reached at.

safetyRating : int – This is the safety rating value for the Carrier.

safetyRateDate : String – This is the date of the last safety rating for the Carrier.

authorize : int – This is a field carried over from Zeus, see Zeus documentation

contractDate : String – This is the date that the Carrier received their contract

insEndDate : String – This is a field carried over from Zeus, see Zeus documentation

costModifierTruck : int – This is the modifier to the cost for this Carrier to supply a truck.

costModifierBike : int – This is the modifier to the cost for this Carrier to supply a bike.

costModifierCargoShip : int – This is the modifier to the cost for this Carrier to supply a cargo ship.

costModifierRail : int – This is the modifier to the cost for this Carrier to supply a train.

costModifierPlane :: int – This is the modifier to the cost for this Carrier to supply a plane.

MAX_SAFETY_RATING : int : 100 –This is the maximum possible value for the safety rating of the Carrier.

MIN_SAFETY_RATING : int : 0 - This is the minimum possible value for the safety rating of the Carrier.

MAX_AUTHORIZE : int : 5 -This is the maximum possible value for the authorize value for the Carrier.

MIN_AUTHORIZE : int : 0 -This is the minimum possible value for the authorize value for the Carrier.

MIN_MODIFIER : int : 0 -This is the minimum possible value for the modifier values for the Carrier.

MAX_MODIFIER : int : 100 -This is the maximum possible value for the modifier values for the Carrier.

DEFAULT_AREA_CODE : String : “0” - This is the default for Carriers.

DEFAULT_AUTHORIZE : int : 0 – This is the default authorize value for Carriers.

DEFAULT_CARRIER_CODE : String : “defaultCarrierCode” - This is the default carrier code value for the Carriers.

DEFAULT_CARRIER_NAME : String : “defaultCarrierName” - This is the default carrier name for the Carriers.

DEFAULT_CONTRACT_DATE : String : “January 1 1970” - This is the default contract date for the Carriers.

DEFAULT_COST_MOD_BIKE : int : 1 – This is the default cost modifier for the Carrier to supply a bike.

DEFAULT_COST_MOD_CARGO : int : 1 – This is the default cost modifier for the Carrier to supply a cargo ship.

DEFAULT_COST_MOD_PLANE : int : 1 – This is the default cost modifier for the Carrier to supply a plane.

DEFAULT_COST_MOD_TRUCK : int : 1 – This is the default cost modifier for the Carrier to supply a truck.

DEFAULT_COST_MOD_RAIL : int : 1 – This is the default cost modifier for the Carrier to supply a train.

DEFAULT_EMAIL_ADDRESS : String : “default@default.com” - This is the default email address for Carriers.

DEFAULT_FAX_NUMBER : String : “123-456-7890” - This is the default fax number for Carriers.

DEFAULT_INS_END_DATE : String : “January 1 1970” - This is the default ins date for the Carriers.

DEFAULT_SAFETY_RATE_DATE : String : “ January 1 1970” - This is the default safety rate date for the Carriers.

DEFAULT_SAFETY_RATING : int : 1 – This is the default safety rating for the Carriers.

Location

travelModes : ArrayList<Vehicles.TravelModes> - These are the method of travels that can be used to reach this Location.

id : int – This is the unique Location id that is set by the database.

latitude : double – This is the global latitude of the Location.

longitude : double – This is the global longitude of the Location.

name : String – This is the name of the Location.

state : String – This is the state the Location is in.

country : String – This is the country the Location is in.

vehiclesAtLocation : ArrayList<Vehicle> : This is an array list of all the Vehicles that are currently at this Location.

Plane

DEFAULT_PLANE_NAME : String -This is the default plane name, if no other name is set, this is the name that will be chosen.

Rail

DEFAULT_RAIL_NAME : String -This is the default rail name, if no other name is set, this is the name that will be chosen.

Segment

id : int – This is the unique Segment id assigned to the Segment from the database.

toID : int – This is the id of the Location that the Segment ends at.

fromID : int – This is the id of the Location that the Segment begins at.

vehicle : Vehicle – This is the vehicle that uses this Segment.

mode : String – This is the mode of travel that the Segment uses.

Distance : double – This is the distance between the starting and ending Locations of the Segment.

DepartureTime : int – This is the estimated departure time that the vehicle will leave the starting Location of the Segment.

ArrivalTime : int – This is the estimated arrival time that the vehicle will reach the ending Location of the Segment.

EarliestArrivalTime : int – This is the earliest time that the vehicle will reach the ending Location.

LatestArrivalTime : int – This is the latest time that the vehicle will reach the ending Location.

EarliestDepartureTime : int – This is the earliest departure time that the vehicle will leave the starting Location.

LatestDepartureTime : int – This is the latest departure time that the vehicle will leave the starting Location.

ShippingRate : ShippingRate – This is the pricing that will be used by this vehicle along this Segment.

TravelType : TravelType – This is the vehicle specifications of the vehicle being using over this Segment.

Lanes : String – This is a field carried over from Zeus, see Zeus documentation

onBoard : ArrayList<Shipment> - This is the list of Shipments that are on board this vehicle instance.

DEFAULT_ARRIVAL_TIME : int : 50 – This is the default estimated arrival time for the vehicle at the end Location

DEFAULT_DEPARTURE_TIME : int : 0 – This is the default estimated departure time for the vehicle at the start Location.

DEFAULT_DISTANCE : double : 100 – This is the default distance between the starting and ending Locations.

DEFAULT_EARLIEST_ARRIVAL_TIME : int : 50 – This is the default earliest arrival time for the vehicle to arrive at the end Location.

DEFAULT_EARLIEST_DEPARTURE_TIME : int : 0 – This is the default earliest departure time for the vehicle to leave the start Location.

DEFAULT_START_LOCATION_ID : int : 1 – This is the default Location id for the start Location.

DEFAULT_LANES : String : “defaultLane” - This is the default value for the lanes for the Segment.

DEFAULT_LATEST_ARRIVAL_TIME : int : 50 – This is the default latest time that the vehicle will arrive at the end Location.

DEFAULT_LATEST_DEPARTURE_TIME : int : 0 – This is the default latest time that the vehicle will leave the start Location.

DEFAULT_MODE : String : “defaultMode” - This is the default mode that the Segment will set for the vehicle.

DEFAULT_END_LOCATION_ID : int : 2 – This is the default Location id for the end Location for the Segment.

Shipment

fromLocationID : int – This is the ID of the Location for the starting Location of the Shipment

toLocationID : int – This is the ID of the Location for the ending Location of the Shipment

size : int – This is the volumetric size of the Shipment

weight : int – This is the weight of the Shipment

earliestArrival : int – This is the earliest time that the Shipment is allowed to arrive at its end Location.

LatestArrival : int – This is the latest time that the Shipment is allowed to arrive at its end Location.

Id : int – This is the unique Shipment ID assigned by the database

earliestDeparture : int -This is the earliest time that a Shipment is allowed to leave the start Location

latestDeparture : int – This is the latest time that a Shipment is allowed to leave the end Location.

History : ArrayList<ShipmentHistory> - This is a list of Shipment History objects that display the path the Shipment took to get to its current Location.

CurrentLocation : int – This is the ID of the current Location of the Shipment

timeToLoad : int – This is the time it takes to load the Shipment onto a Vehicle.

TimeToUnload : int – This is the time it takes to unload the Shipment from the Vehicle.

ShippedID : int – This is the ID of the Shipper that the Shipment was sent by.

TollRoads : Boolean – This is a Boolean indicating if the Shipment is allowed to take toll roads.

CongestionByPass : Boolean - This is a Boolean indicating if the Shipment is allowed to perform a bypass to avoid congestion.

MaxStops : int – This is the maximum number of stops the Shipment is allowed to take.

HazmatConstraints : String – These are the various hazardous material constraints that apply to this Shipment.

LoadingRate : int – This is the rate that the Shipment can be loaded onto the Vehicle.

TrailerType : String – This is the type of trailer that the Shipment requires.

LoadingType : String – NEED MORE INFORMATION

unloadingType : String – NEED MORE INFORMATION

prefCarriers : String – This is a list of the preferred Carriers that the Shipment would like to travel by.

DEFAULT_START_LOCATION_ID : int : 1 – This is the default start Location ID for the Shipment.

DEFAULT_END_LOCATION_ID : int : 2 – This is the default end Location ID for the Shipment.

DEFAULT_PRIORITY : int : 1 – This is the default priority level of the Shipment.

DEFAULT_SIZE : int : 50 – This is the default volume of the Shipment.

DEFAULT_WEIGHT : int : 100 – This is the default weight for the Shipment.

DEFAULT_EARLIEST_ARRIVAL : int : 50 – This is the default earliest arrival time that the Shipment can reach its destination.

DEFAULT_LATEST_ARRIVAL : int : 50 – This is the default latest arrival time that the Shipment can reach its destination.

DEFAULT_EARLIEST_DEPARTURE : int : 0 – This is the default earliest time that the Shipment can leave the start Location.

DEFAULT_LATEST_DEPARTURE : int : 0 – This is the default latest time that the Shipment can leave the start Location.

DEFAULT_TIME_TO_LOAD : int : 0 – This is the default time to load the Shipment onto a Vehicle.

DEFAULT_TIME_TO_UNLOAD : int : 0 – This is the default time to unload the Shipment from a Vehicle.

DEFAULT_SHIPPER_ID : int : 1 – This is the default id for the Shipper that sent the Shipment.

DEFAULT_MAX_STOPS : int : 100 – This is the default number of stops the Shipment is allowed to make on its path from the start to end.

DEFAULT_HAZMAT_CONSTRAINTS : String : “defaultHazardConstraints” - This is the default value for the hazardous material constraints for the Shipment.

DEFAULT_LOADING_RATE : int : 1 – This is the default loading rate for the Shipment onto the Vehicle.

DEFAULT_TRAILER_TYPE : String : “defaultTrailerType” - This is the default trailer type that the Shipment can use.

DEFAULT_UNLOADING_TYPE : String : “defaultUnloadingType” - This is the default unloading type for the Shipment.

DEFAULT_LOADING_TYPE : String : “defaultLoadingType” - This is the default loading type for the Shipment.

DEFAULT_PREFERRED_CARRIERS : String : “defaultPreferredCarriers” - This is the default preferred carriers value for the Shipment.

MIN_SIZE : int: 0 – This is the minimum size that the Shipment can be.

MAX_SIZE : int : 100 - This is the maximum size that the Shipment can be.

MIN_PRIORITY : int : 1 – This is the minimum priority that the Shipment can have.

MAX_PRIORITY : int : 10 – This is the maximum priority that the Shipment can have.

MIN_WEIGHT : int : 1 – This is the minimum weight that the Shipment can have.

MAX_WEIGHT : int : 500 – This is the maximum weight that the Shipment can have.

MIN_LOAD_TIME : int : 0 – This is the minimum load time that the Shipment takes to load onto the Vehicle.

MAX_LOAD_TIME : int : 100 – This is the maximum load time that the Shipment takes to load the Vehicle.

MIN_LOAD_RATE : int : 1 – This is the minimum load rate that the Shipment can have.

MAX_LOAD_RATE : int : 100 – This is the maximum load rate that the Shipment can have.

MIN_STOPS : int : 1 – This is the minimum number of stops that the Shipment can have before reaching its destination.

MAX_STOPS : int : 500 – This is the maximum number of stops that the Shipment can have before reaching its destination.

ShipmentHistory

id : int – This is the unique ShipmentHistory id assigned by the database.

SegmentID - This is the id of the Segment that the Shipment traveled along.

ShipmentID – This is the id of the shipment that is moving along this path.

NodeNumber – NEED MORE INFORMATION

DEFAULT_NODE_NUMBER : int : 0 – NEED MORE INFORMATION

DEFAULT_SEGMENT_ID : int : 1 – This is the default id of the Segment for the ShipmentHistory.

DEFAULT_SHIPMENT_ID : int : 1 – This is the default id of the Shipment for the ShipmentHistory.

Shipper

id : int – This is the unique id for the Shipper assigned by the database.

LocationID : int – This is the id of the Location that the Shipper can be found at.

PrefCarriers : String – This is the list of Carriers the Shipper prefers to use.

CompanyName : String – This is the company name of the Shipper

ContactName : String – This is the name of contact person at the Shipper's company.

Phone : String – This is the phone number used to contact the Shipper

email : String – This is the email address that can be used to contact the Shipper

DEFAULT_COMPANY_NAME : String : “defaultCompanyName” - This is the default company name for the Shipper

DEFAULT_CONTACT_NAME : String : “defaultContactName” - This is the default contact name for the Shipper

DEFAULT_EMAIL_ADDRESS : String : “defaultEmailAddresss” - This is the default email address for the Shipper

DEFAULT_LOCATION_ID : int : 1 – This is the default id for the Location of the Shipper

DEFAULT_PHONE_NUMBER : String : “123-456-7890” - This is the default phone number for the Shipper.

DEFAULT_PREFERRED_CARRIERS : String : “defaultPreferredCarriers” - This is the list of preferred carriers that the Shipper uses.

ShippingRate

id : int – This is the unique ShippingRate id assigned by the database.

Carrier : Carrier – This is the Carrier that uses this ShippingRate

startLocation : Location – This is the starting Location for which this rate will apply

endLocation : Location – This is the ending Location for which this rate will apply

type : TravelType – This is the type of Vehicle that uses this ShippingRate.

Weight1 : double – This is the weight for under which Rate1 shipping fees will apply

Weight2 : double – This is the weight for which Rate2 shipping fees will apply

Weight3 : double – This is the weight for which Rate3 shipping fees will apply

Rate1 : double – This is the first tier of shipping rates

rate2 : double – This is the second tier of shipping rates

rate3 : double – This is the third tier of shipping rates

mileRate : double – This is the rate of shipping fees that get applied per mile traveled.

FlatRate : double – This is a single flat rate that may apply to the Shipment.

Rank : int – NEED MORE INFORMATION

DEFAULT_WEIGHT1 : double : 10 – This is the default value of weight1 for which rate1 will apply.

DEFAULT_WEIGHT2 : double : 20 – This is the default value of weight2 for which rate2 will apply.

DEFAULT_WEIGHT3 : double : 30 – This is the default value of weight3 for which rate3 will apply.

DEFAULT_RATE1 : double : 10 – This is the default shipping fees that apply to Shipments under weight1

DEFAULT_RATE2 : double : 20 – This is the default shipping fees that apply to Shipments under weight2.

DEFAULT_RATE3 : double : 30 – This is the default shipping fees that apply to Shipments under weight3.

DEFAULT_MILE_RATE : double : 10 – This is the default shipping fees that will be applied per mile the Shipment must travel.

DEFAULT_FLAT_RATE : double : 1 – This is the default shipping fee that will be applied to the Shipment at the flat expense.

DEFAULT_RANK : int : 1 – NEED MORE INFORMATION.

TravelType

vehicleTypeID : int – This is the unique vehicle type id assigned by the database

vehicleTypeName : String – This is the name of the vehicle type.

VehicleMode : String – This is the mode of travel for the travel type.

Trailer1 : String – This is first trailer that is attached to this travel type

Trailer2 : String - This is the second trailer that is attached to the travel type

minCap : double – This is the minimum capacity that the travel type needs to load for it to leave its Location

maxCap : double – This is the maximum capacity that the travel type can have so that it can leave its Location.

ActCap : double – This is the actual capacity of the travel type currently

maxWeight : double – This is the maximum weight that the Travel Type can carry

serviceType : String – NEED MORE INFORMATION

radiation : Boolean – This is the Boolean that indicates if this Travel Type can handle radioactive Shipments.

Refrigeration : Boolean – This is the Boolean that indicates if the Travel Type can handle Shipments that require refrigeration.

Hazmat : Boolean – This is the Boolean that indicates if this Travel Type can handle hazardous material Shipments.

Explosives : Boolean – This is the Boolean that indicates if this Travel Type can handle explosive Shipments.

Tracking : Boolean – This is the Boolean that indicates if this Travel Type has tracking available.

DEFAULT_VEHICLE_TYPE_ID : int : 0 – This is the default id or the Travel Type

DEFAULT_VEHICLE_TYPE_NAME : String : “defaultVehicleTypeName” - This is the default vehicle type name for the Travel Type.

DEFAULT_VEHICLE_MODE : String : “defaultVehicleMode” - This is the default value for the TravelType's vehicle mode of travel.

DEFAULT_TRAILER1 : String : “defaultTrailer1” - This is the default value for trailer1

DEFAULT_TRAILER2 : String : “defaultTrailer2” - This is the default value for trailer2

DEFAULT_MINIMUM_CAPACITY : double : 0 – This is the default minimum capacity of the TravelType.

DEFAULT_MAXIMUM_CAPACITY : double : 100 – This is the default maximum capacity of the TravelType.

DEFAULT_MAXIMUM_WEIGHT : double : 500 – This is the default maximum capacity of the TravelType.

DEFAULT_SERVICE_TYPE : String : “defaultServiceType” - This is the default service type for the for the TravelType.

Truck

DEFAULT_TRUCK_NAME : String -This is the default truck name, if no other name is set, this is the name that will be chosen.

Routing

RoutingAlgorithm

route : ArrayList<Segment> - This is the route that the Shipment needs to take to get from its current Location to the end Location

metric : WeightedMetric – This is the cost function that is used to determine the lowest costs from Location to Location.

CurrentTime : int – This is the current time

shipment : Shipment – This is the shipment that is being routed from its current Location to the end location.

PathFound : boolean – This is the boolean indicating if the path was successfully found or not

maxTries : int – This is the maximum number of attempts the algorithm is allowed to try and find a path from the current Location of the Shipment to the ending Location of the shipment.

WeightedMetric

distance : int – This is the level of importance for the distance, the higher it is the more important.

Time : int – This is the level of importance for the time, the higher it is the more important

cost : int – This is the level of importance for the cost, the higher it is the more important.

DistNWeight : double – This is the normalized weight of the importance of distance

timeNWeight : double – This is the normalized weight of the importance of the time

costNWeight : double – This is the normalized weight of the importance of the cost

MIN_DISTANCE : int : 0 – This is the minimum importance of the distance

MAX_DISTANCE : int : 100 – This is the maximum importance of the distance.

MIN_TIME : int : 0 – This is the minimum importance of the time

MAX_TIME : int : 100 – This is the maximum importance of the time

MIN_COST : int : 0 – This is the minimum importance of the cost

MAX_COST : int : 100 – This is the maximum importance of the cost.