# EcoPRT Database Schema

\*All Object IDs are created by MongoDB and are hexadecimal which are made of:

* a 4-byte value representing the seconds since the Unix epoch,
* a 3-byte machine identifier,
* a 2-byte process id, and
* a 3-byte counter, starting with a random value.

(see [MongoDB documentation](https://docs.mongodb.com/manual/reference/method/ObjectId/)) \*

\* Tables are called Collections in MongoDB \*

\*\* If there is a row with a strikethrough, then it is not in the current database, but may be used in the future \*\*

Users Collection (if one device per user)

Collection Name: Users

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Format** | **Description** |
| \_id | ObjectID | * Unique identifier for the user. Auto generated. * For this Collection it’s shown as \_id, but in other Collections it called userID |
| username | String | * Screen name chosen by the user. Unique (or not?). |
| hash | String | * SHA512 hash of password + salt |
| salt | String | * Randomly generated string that gets concatenated to the password before the password is hashed. |
| email | String | * Email the user gives types in the field when creating an account |
| lowercaseEmail | String | * When an email is entered, it will be changed to all lowercase just in case a user enters a capital somewhere * This is what is checked when a user is logging in. * Does it so that all emails are the same format. |
| firstName | String | * First Name of the user that has created an account |
| lastName | String | * Last Name of the user that has created an account |
| birthday | Date | * Birthday of the user * Represented as milliseconds in the database * Front end will take mm/dd/yyyy * This will be used to keep track of how old users are that way underage users aren’t making accounts |
| userType | Number | * 0 for admin * 1 for normal user |

Password/Authentication Token Collection (if one user can log into multiple devices)

Collection Name: AuthTokens

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Format** | **Description** |
| userID | ObjectID | * Unique identifier for the user. |
| authToken | String | * Random string of letters and numbers |

Vehicles Collection

Collection Name: Vehicles

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Format** | **Description** |
| \_id | ObjectID | * Unique identifier for the vehicle. Auto generated. |
| name | String | * Name of the vehicle |
| enabled | Boolean | * False - vehicle should cease operation and not be assigned to rides * True - vehicle can do normal behavior (drive on edges, pick up passengers, go on rides) |
| batteryLife | Number | * Current Battery life of the vehicle. * Represented as a percentage |
| coordinates | Array of Numbers | * Location of the vehicle. * Represented as Latitude and Longitude.   + Decimal Numbers   + Length of 2 |
| steeringAngle | Number | * Angle the wheels are facing * Represented as radians |
| headingAngle | Number | * Angle the vehicle is facing * Represented as radians |
| speed | Number | * Current speed of the vehicle * Represented as mph (for now) |
| rideID | ObjectID | * What ride the vehicle is currently doing |
| edgeID | ObjectID | * What edge the vehicle is currently on |
| rideQueue | Array of rideIDs | * List of rides that are queued up for this vehicle |

Vehicle History Collection

Collection Name: vehiclesHistory

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Format** | **Description** |
| \_id | ObjectID | * Unique identifier for the vehicle history entry. |
| vehicleID | ObjectID | * ID of the vehicle whose history this is for |
| batteryLife | Number | * Current Battery life of the vehicle. * Represented as a percentage |
| coordinates | Array of Numbers | * Location of the vehicle. * Represented as Latitude and Longitude.   + Decimal Numbers   + Length of 2 |
| steeringAngle | Number | * Angle the wheels are facing * Represented as radians |
| headingAngle | Number | * Angle the vehicle is facing * Represented as radians |
| speed | Number | * Current speed of the vehicle * Represented as mph (for now) |
| rideID | ObjectID | * What ride the vehicle is currently doing |
| edgeID | ObjectID | * What edge the vehicle is currently on |
| rideQueue | Array of edgeIDs | * List of rides that are queued up for this vehicle |
| ~~goal~~ | ~~Number~~ | * ~~Different tasks or statuses for the vehicle~~   + ~~On~~   + ~~Off~~   + ~~Stopped~~   + ~~Idle~~   + ~~Ride in Progress~~   + ~~Has Passengers~~   + ~~vehicle is moving~~ |
| ~~isOn~~ | ~~Boolean~~ | * ~~Whether the vehicle is on or off.~~   + ~~True if On~~   + ~~False is Off~~ |

Edge Collection

Collection Name: Edges

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Format** | **Description** |
| \_id | Object ID | * Unique identifier for an edge. Auto generated |
| startingNode | ObjectID | * First node of the edge (should refer to item in Node Collection) |
| endingNode | ObjectID | * Last node of the edge (should refer to item in Node Collection) |
| distance | Number | * How long the edge is between two nodes. * Represented in miles |
| waypoints | Array of Waypoints | * All the points that represent an edge. * A waypoint Includes:   + Coordinates   + Speed   + Steering angle   + Heading angle   + Timestamp   + Tick   + Actuator   + Motor Control Flag   + Motor Throttle * Format and description of waypoint in another Collection |

Waypoint (not an actual collection in the database)

|  |  |  |
| --- | --- | --- |
| *coordinates* | *Array of Numbers* | * *Location of the waypoint* * *Represented as Latitude and Longitude*   + *Decimal number*   + *Length of 2* |
| *speed* | *Number* | * *Speed of the vehicle when it reaches this point* * *Represented as mph (as of now)* |
| *steeringAngle* | *Number* | * *Angle of the vehicle’s wheels when it reaches this waypoint* * *Represented as radians* |
| *headingAngle* | *Number* | * *Angle the vehicle is facing when it reaches this waypoint* * *Represented as radians* |
| *timeStamp* | *Date* | * *Date and time the vehicle reached the waypoint* |
| *tick* | *Number* | * *Index of the waypoint on the edge that the vehicle has arrived at.* * *Will get higher as you go to a new waypoint* |
| *actuator* | *Bytes* | * *Array of 3 numbers* * *3 bytes stored as numbers in database* |
| *motorControlFlags* | *Int8* | * *8 boolean values representing a bit* |
| *motorThrottle* | *Byte* |  |

Node Collection

Collection Name: Nodes

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Format** | **Description** |
| \_id | ObjectID | * Unique identifier for a node. Auto generated |
| coordinates | Array of Numbers | * Location of the node * Represented as Latitude and Longitude   + Decimal numbers   + Length of 2 |
| name | String | * Name of the node (optional) |
| type | Number | * What type of node this is   + 0: pickup station   + 1: charging station   + 2: docking station   + 3: not a station/just a node |
| ~~stationInfo~~ |  | ~~Has certain things depending on what type of station it is.~~ |

Rides Collection

Collection Name: Rides

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Format** | **Description** |
| \_id | ObjectID | * Unique identifier for a ride. Auto generated |
| vehicleID | ObjectID | * Vehicle that picks up the passengers and is doing the ride |
| userID | ObjectID | * ID of the user(s) that is/are in the ride |
| startingNode | ObjectID | * Node that the vehicle is staring the ride from |
| endingNode | ObjectID | * Node that the vehicle will stop at |
| pickupNode | ObjectID | * Node that the user will be picked up from by the vehicle |
| dropoffNode | ObjectID | * Node that the user will be dropped off at by the vehicle |
| passengerCoordinates | Array of Numbers | * Location of the user who requested a ride * Represented as longitude and latitude * Decimal number * Length of 2 |
| numberOfPassengers | Number | * How many people the vehicle has for the ride |
| requestTime | Date | * Time when the user requests a ride after hitting the request button. * Represented a milliseconds |
| vehicleArrivalTime | Date | * Time when the vehicle arrives at the passenger’s location to get picked up at. * Represented as milliseconds |
| pickupTime | Date | * Time when the passenger gets in the vehicle and starts their ride. * Represented as milliseconds |
| dropoffTime | Date | * Time when the vehicle arrives at the destination and drops the passenger(s) off. * Represented as milliseconds |
| distance | Number | * Distance of the total trip * Represented as miles |
| currentTask | Number | * Number that corresponds to the status of the ride:   + 0: created and needs a path   + 1: found path, heading to starting node   + 2: passenger is picked up   + 3: ride is completed   + 4: ride was cancelled |
| pathChosen | Array of edgeIDs | * What path the vehicle is taking |
| sentTask | Boolean | * True - The vehicle is ready to do the ride * False -The vehicle is not ready for a ride yet |