**Asset Optimization project**

**Requirements Elaboration for Cascade Engine**

**For: National Express Corporation**

**By: Bytecurve LLC**

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 6/1/2017 | 1.0 | Completed document incorporating all stakeholder discussions | Ranga Gopalan |
| 8/7/2017 | 1.1 | Updates to cascade matching process and UI, based on meeting on 8/7  No change to the sections on | Ranga Gopalan |
| 8/17/2017 | 1.2 | Added functionality to ‘save’ a snapshot of the state of the system as a backup and the ability to ‘restore’ a previously saved snapshot.  Added templates pertaining to the ‘match’ process to the document. | Ranga Gopalan |
| 8/30/2017 | 1.3 | Updated based on clarification meeting with Karen, Walter, Greg and Keshav on 8-30-2017. | Ranga Gopalan |

# Introduction

This document describes the detailed requirements for the development of the ‘Cascade Engine’. This is a further elaboration of the original ‘Asset Optimization Project Functional Specifications’ document.

The Cascade Engine application will provide three main capabilities, as follows:

1. The ability to capture, review and confirm ‘available’ assets [“Supply”]
2. The ability to capture, review and confirm ‘required’ assets [“Demand”]
3. Ability to match the ‘demand’ for vehicles from various locations (from bullet #2 above) against the ‘supply’ of available vehicles (from bullet #1 above), so that available vehicles can then be moved or ‘cascaded’ to the locations where they are needed.

Detailed requirements for each of the above activities are listed below.

# Vehicle ‘Supply’ Assessment

1. The user will select the school year for which the Supply is to be worked on.
2. Only NEC owned vehicles are to be considered for ‘cascading’. Buses operated by but not owned by NEC cannot be cascaded and hence should not be included among available buses. Vehicles with the following values for the ‘owner' attribute are to be excluded: ‘Customer’, ‘District’, ‘District-White’, ‘Sold’ and ‘State’.
3. The system will pull in a list of vehicles that are going to become available due to vehicles exceeding the ‘maximum age’ specified in the contract. This will be performed as follows:
   1. Get the MaxAge for all contracts from Salesforce.com (SFDC).
   2. For each contract, check the fleet of the CSC serving the contract and identify vehicles that will be over MaxAge as of August 1st of the user selected year.
   3. If a contract has no MaxAge specified, the system shall use a value of 12 years for MaxAge. Whenever the default is used, it will be highlighted on the display screen in a user identifiable manner.
   4. If a CSC is serving multiple contracts, the highest value of MaxAge specified among all these contracts will be applied to the entire fleet of the CSC, to identify vehicles that will be over-age for the selected school year.

The above activity will be performed through an automated interface, and will run every night, identifying and adding any new vehicles that satisfy the criteria to the pool of available buses.

1. The system will provide the user the ability to manually add additional assets to the availability list, from among all assets in the EAM application.
2. The system will allow the user the ability to review each asset in the availability list and confirm its availability. Only assets marked ‘available’ will be considered for matching by the Cascade Engine.
3. The system will provide a mechanism to the user to assign a ‘Reason for Availability’, to the assets being marked as available. This will be one of the following:
   1. Vehicle aged out by system
   2. CSC Site closing [provision for mass update]
   3. Route reduction
   4. Contractual mandate to replace with new bus
   5. District buying buses, NEC buses not needed
4. 'UI will include the following filter criteria, to select the ‘available’ assets for review:
   1. Contract name (dropdown list)
   2. CSC # (dropdown list)
   3. CSC Name (auto populate when CSC# is selected)
   4. State
   5. Date available
   6. Reason for availability
   7. Bus Type
   8. Max Age
   9. Include vehicles with one or more of the following dispositions from prior matches:
      1. Confirmed
      2. On hold
      3. System matched
      4. Open
   10. Asset #, with ability to enter multiple comma separated asset numbers

Vehicles matching all of the above criteria will be displayed except for the Asset # - all assets specified by their asset numbers will be displayed irrespective of their other attributes in the filter criteria.

1. The following information will be displayed on the vehicle availability screen:
   1. Asset number
   2. Owner
   3. CSC number
   4. CSC name
   5. State
   6. Contract name (if the CSC supports multiple contracts, comma separated list)
   7. Usage code
   8. Vehicle type
   9. Seating capacity
   10. A/C
   11. Wheelchair capacity
   12. Model year
   13. Reason for availability
   14. Match status (one of “Open”, “On hold”, “Confirmed” or “System Matched”)
   15. MaxAge (show contract actual and defaults if any assumed) comma separated value for each contract [this can go into the ‘details’ section if no space]
2. The following information will be displayed when the user clicks on the ‘Details’ button:
   1. All attributes in item #8 above
   2. Org
   3. Brake type
   4. Engine Make
   5. Fuel type
3. The following fields will be editable:
   1. Reason for Availability
   2. Date available
   3. Note field
   4. Availability flag

If the user marks a bus that was previously matched as ‘available’ the system will nullify the match and update the ‘vehicle requirements’ accordingly.

When a CSC is being closed, a large number of vehicles will need to be marked as ‘available’, with the same ‘reason for availability’ and the same value for ‘date available’. The system needs to provide a feature to perform a mass update of the above two fields for all vehicles displayed on the screen.

1. Export function – this will create an excel extract of all assets that are/were part of the availability pool at any time for the current school year. This will include the following attributes:
   1. Asset #
   2. Various vehicle attributes from items #7, #8, #9 and #10 above.
   3. Availability status
   4. If already matched against an existing requirement, the disposition from the last match, ie., ‘Confirmed’, ‘On hold’, ‘System matched’, ‘Transfer initiated’, ‘Transferred’.

# Vehicle ‘Demand’ Assessment

1. Information about all vehicle ‘requirements’ will be entered through Excel import of ‘Cascade Request forms’. The excel file will contain the following information:
   1. Requestor name
   2. Date of request
   3. CSC name
   4. CSC #
   5. State
   6. Org
   7. Reason for the requirement, with the following possible values:
      1. “New biz” (to meet new business needs)
      2. “Age” (to replace aging bus)
      3. “Mech” (to replace bus with mechanical issues)
      4. “Add Route” (to cover for additional routes)
      5. “Spare” (to meet spare requirements)
      6. “Physical Damage” (to meet accident replacement)
   8. Name of contract
   9. MaxAge
   10. Average Age
   11. When age requirements are to be met (“Jan”, “Beg SY”, “Not Specified”, blank)
   12. # of buses required (of the same type).
   13. Vehicle attributes
       1. Bus type
       2. Seating capacity
       3. W/C capacity
       4. Fuel type required
       5. Brake type required
       6. Other special requirements
   14. School start date
   15. Date units needed at CSC
   16. Other requirements (free form text)
2. The file uploads will include the following validation rules:
   1. CSC number and location state match
   2. Vehicle types should match one of valid types, per EAM
   3. Contract name will be validated against Salesforce, and if invalid, the system will raise a warning message to notify the user, but will allow the user to proceed with the upload.
3. For the above excel uploads, the system shall provide a brief message indicating:
   1. The number of records being added to the system
   2. Error and warning messages from validations, if any, with details.

If there are warnings but no errors, the user will have the option to proceed with the upload. If there are any errors, the user will need to correct the errors and re-load the file.

1. All excel uploads will be incremental, ie., the system will not check if any of the records in the file are already in the system.
2. The system shall include user interface (UI) screens for the user to review, validate and update current vehicle requirements. The details are as follows:

UI Requirements:

1. The UI will include the following filter criteria, to select the ‘vehicle requirements’ the user may want to work on. Only those records that satisfy all of the specified criteria will be displayed.
   1. Contract name (dropdown list)
   2. CSC # (dropdown list)
   3. CSC Name (auto populate when CSC# is selected)
   4. State
   5. Date vehicle required
   6. Include requirements with one or more of the following status (“New”, “Verified”, “Pending”, “Deleted”)
   7. Include vehicles with one or more of the following dispositions from prior matches:
      1. Confirmed
      2. On hold
      3. System matched
      4. Open
   8. The file-name of previously uploaded files. When a previous file-name is specified, all records that were sourced originally from that file will be included but will be filtered based on other filter criteria that may also be specified.
2. The following information will be displayed on the vehicle Requirements screen:
   1. Contract name
   2. CSC #
   3. CSC Name
   4. State
   5. Reason for request
   6. Status (‘Verified’, ‘Deleted’ or ‘Pending’
   7. Vehicle attributes:
      1. Bus type
      2. Vehicle capacity
      3. W/C capacity
      4. A/C (Y/N)
      5. Max Age
   8. Disposition of prior match (‘Confirmed’, ‘On hold’, ‘System matched’, ‘Open’)
3. When a ‘Requirement’ is for multiple buses of the same attribute, the requirement will be displayed as multiple line items, one line for each vehicle. The user will be able to view all the requirements collectively by choosing appropriate filter criteria.
4. The UI shall allow the user to make the following edits / updates to the displayed vehicle requirements:
   * 1. Request reason
     2. Any of the vehicle attributes
     3. Status field, to mark each request as ‘Verified’, ‘Deleted’ or ‘Pending’. If any of the status changes results in revoking a previously made match, the system will return the previously matched vehicle to the ‘available’ pool.
     4. Comments / notes section, for each request.
5. The user will have the option to view full details of the ‘Requirement’ by clicking on a ‘Details’ button, in which case the following attributes will be displayed:
   1. Brake type
   2. All attributes in #7 above
6. Export function – this will create an excel extract of all the ‘Requirements’ that are/were part of the ‘Requirements’ pool at any time for the current school year. This will include the following attributes:
   1. Various attributes as called out in section #7 above.

# Matching Vehicle Requirements with Availability

The following section covers various scenarios related to the interaction of the user with the Cascade Engine application and the system behavior

## Key Scenarios

1. User wants to run the Cascade Engine
2. User wants to review the matches identified by the system after a Cascade Run
3. User wants to:
   1. Perform ongoing review of matches from prior Runs and update match status
   2. Update status of certain matches in the system, based on administrators own information
   3. Review system updates from nightly batch updates
4. User wants to save a copy of the current state of the system.
5. User wants to restore a previously saved state of the system.

## Flow of Events – Run Cascade Engine

### Actors

Application administrator

### Pre-conditions

1. The administrator has reviewed the ‘Availability’ of buses and ‘confirmed’ vehicles exist in the available pool of vehicles.
2. Vehicle ‘Requirements’ have been uploaded and validated by the administrator.

### Main flow – Run Cascade Engine

| **Step** | **User Action** | **System Action** |
| --- | --- | --- |
|  | User clicks on the ‘Run Cascade Engine’ option in the navigation menu. | The ‘Run Cascade Engine’ screen is displayed, with the option to set the following parameter options before running the ‘cascade engine’:   1. Requesting location (suggested search dropdown, default blank) 2. Target contract name (suggested search dropdown, default blank) 3. Target state (dropdown list, default blank) 4. ‘Requirement types to include’ items to include (check boxes)    1. Open (New requirement, not part of prior Runs, OR no matches found in prior runs; checked by default)    2. System matched (checked by default)    3. On hold    4. Confirmed   *(‘transfer-initiated’ is not an option as at this status, its past the point of putting back in the mix for re-matching).*   1. Option to restrict the cascade to same state (unchecked by default) 2. Restrict the distance the bus will need to be moved to a set value 3. Restrict the move cost to a set amount 4. Threshold value for the ‘vehicle matching score’ below which the match is to be rejected. 5. Include new business (check box, unchecked by default) |
|  | User completes the parameter list as desired and clicks on the ‘Run’ button.  The user may also click on the ‘Reset’ button at any time to reset the parameters to their default values. | 1. The system will run the Cascade Engine applying the criteria selected by the user. 2. Only confirmed ‘Requirements’ will be considered for the Cascade. 3. Only vehicles confirmed ‘Available’ will be considered for the Cascade. 4. After the Cascade Run is complete, the system will display the ‘Cascade Results’ page.   The rules for the ‘match process’ are discussed in Appendix A.  *(Additional details in the next section “Work with Cascade Engine results”* |

## Flow of Events – Review / accept matches after a Cascade Run

### Actors

Application administrator

### Pre-conditions

The administrator has just run the Cascade Engine

### Main flow – Work with Cascade Engine results

| **Step** | **User Action** | **System Action** |
| --- | --- | --- |
|  | User clicks on the ‘Cascade Results’ option in the navigation menu. | The ‘Cascade Results’ screen is displayed. This includes a ‘filter criteria’ section that can be used to narrow or expand the result-set per users’ preference.  The following filter criteria will be available:   1. Requesting location (suggested search dropdown, default blank) 2. Region (dropdown list, of all Operations Hierarchy regions). 3. Target contract name (suggested search dropdown, default blank) 4. Target state (dropdown list, default blank) 5. Match status – check boxes to select one or more of the following: 6. Open - no matches found 7. System matched (checked by default) 8. On hold 9. Confirmed 10. Transfer initiated 11. Transfer complete 12. Exception   **Note:**   * All requirements that were matched by the last Run of the Cascade Engine will have a default value of ‘System Matched’. * If the user checked the option to include requirements that were previously placed ‘on hold’ or ‘confirmed’, new matches would be assigned to these Requirements. * Requirements for which no match can be found will have a status of Open.  1. The number of results to return from the match process. Drop-down box with three choices: 2. Top 3 results 3. Top 5 results (selected by default) 4. All results that pass the match criteria   **Note:** since the system is providing multiple asset matches for each Requirement, the same asset may be included as an option for multiple requirements. |
|  | The user selects the filter criteria desired and clicks on ‘Filter’.  [UI mockup has the button labeled as ‘Submit’, this needs to be changed.] | The system displays the results filtered as per user entered criteria.  For each Requirement, the following details will be displayed:  Attributes of the Requirement:   1. Contract name 2. CSC# and name 3. State 4. Bus type 5. Max Age   Attributes of the matching Asset:   1. The matching assets identified by the system will be displayed as a drop-down list in the ‘Matching Asset’ field. Depending on the user selected parameter in the filter criteria, the top 3, 5 or all matches will be displayed in the decreasing order of their rank (ranking approach in Appendix A). |
|  | For each Requirement, the user selects one of the system suggested matches from the drop-down list. | The following attributes related to the selected asset are displayed:   1. Vehicle year 2. Vehicle match score 3. State specification match (Yes or Partial) 4. Retrofit cost 5. Transfer distance |
|  | The user clicks on the ‘More’ button in the ‘Action’ column. | All attributes of the ‘Requirement’ and the attributes of the selected asset are displayed, as follows:   |  |  | | --- | --- | | **Requirements attributes** | **Attributes of proposed asset match** | | Reason for requirement (one of items in section 3, #1.7) | Reason for availability | | Contract name | Owner | |  | Org | | MaxAge | Model year | | Bus type | Bus type | | Seating capacity | Seating capacity | | Wheel chair capacity | Wheel chair capacity | | A/C (Y/N) | A/C (Y/N) | | Fuel type | Fuel type | | Brake type | Brake type | | Special requirement | Fuel type | | School start date | Engine make | | Date vehicle needed | Usage code | | Other requirements (from Cascade form upload) |  |   The above information is for the users’ reference, to help them ensure proper matching. The user will be able to close this window after reviewing the attributes. |
|  | After reviewing all the attributes of the Requirement and the proposed Match, the user changes the ‘Match Status’ column to ‘On Hold’. | The updated status is displayed in the ‘Match Status’ column. |
|  | **Manual match:**  The user clicks on the ‘Manual match’ option | The user is provided the option to select an asset from a ‘suggested search’ drop-down list; all assets currently available in EAM will be available for selection. |
| User selects an asset number from the drop-down list, and clicks on the ‘check’ button. | The system checks if the asset is assigned to another Requirement; if it is, the system will display a message indicating the CSC to which the vehicle is assigned, and the status of the current match. |
| The user over-rides the previous match  OR  The user clicks on ‘Cancel’ to retain the previous match. | The user selected asset is matched to the current Requirement and the status is set to ‘on hold’. The Requirement to which the asset was previously assigned is now marked as ‘Open’.  The system ignores the match and the user is returned back to the ‘cascade match’ screen, with previous values populated as before. |
|  | The user clicks on ‘Submit’ button. | The system saves all changes made to the ‘Match Status screen since the last time the Save button was clicked.  The Match Result screen is refreshed, such that any asset that is put ‘on hold’ for a Requirement will no longer be a choice for any of the remaining Requirements, even if the asset was previously shown as a possible match. |
|  | The user clicks on the ‘Reset’ button | All changes made since the last ‘submit’ are revoked and the data is restored to the time of last Submit. |

## Flow of Events – Resume review of Cascade Matches, update status of previous matches, review updates from nightly batch job

### Actors

Application administrator

### Pre-conditions

* 1. Cascade Runs have been executed previously, user wants to perform ongoing review of the status of current cascades
  2. Nightly updates are being applied to status of Asset transfers

### Main flow – Ongoing review/update of prior Cascade matches

| **Step** | **User Action** | **System Action** |
| --- | --- | --- |
|  | User clicks on the ‘Cascade Results’ option in the navigation menu. | The ‘Cascade Results’ screen is displayed. This includes the filter criteria already described in the previous flow. |
|  | Based on inputs from offline follow-ups, user makes the following updates:   * ‘on hold’ to ‘confirmed’, if cascade is good to go. * ‘on hold’ to ‘open’, if cascade is not possible * ‘on hold’ or ‘confirmed’ to ‘transfer initiated’ if paperwork for transfer has started.   After making the above updates, the user clicks on ‘Submit’. | The system will save the updates as entered by the user.  If status is marked as ‘open’ the Requirement will be included in subsequent Cascade Runs, and the Asset previously assigned is released to the Available pool (user would need to take if off the pool in the Availability screen if it is not to be included in future Cascade Runs).  Nightly batch job actions:  If current location of an Asset in ‘Transfer initiated’ status:   * Matches the target location, the system will change the status to ‘Transfer completed’. * Does not match the target location, but is different from source location, the system will change the status to ‘Exception’ (the user will need to investigate, and if the issue cannot be resolved, the user will need to change the status of this Requirement to ‘Open’. |
|  | User clicks on the ‘Export’ button | The system provides an option to create an excel extract of all matches completed for the school year. All attributes of the ‘Requirement’ and the attributes of the selected asset will be included, as follows:   |  |  | | --- | --- | | **Requirements attributes** | **Attributes of proposed asset match** | | Reason for requirement (one of items in section 3, #1.7) | Reason for availability | | Contract name | Owner | |  | Org | | MaxAge | Model year | | Bus type | Bus type | | Seating capacity | Seating capacity | | Wheel chair capacity | Wheel chair capacity | | A/C (Y/N) | A/C (Y/N) | | Fuel type | Fuel type | | Brake type | Brake type | | Special requirement | Fuel type | | School start date | Engine make | | Date vehicle needed | Usage code | | Other requirements (from Cascade form upload) |  | |

## Flow of Events – Save a copy of the ‘current state’ of the system; restore the system to a previously saved version

### Actors

Application administrator

### Pre-conditions

User has the necessary credentials to perform the ‘save’ and ‘restore’ functionality

### Main flow – ‘save’ current state of the system, ‘restore’ a previously saved version.

| **Step** | **User Action** | **System Action** |
| --- | --- | --- |
|  | On the ‘Cascade Engine Results’ screen, user clicks on the ‘Backup’ button. | The system displays an entry field for the user to enter a label/name for the the version to be saved. This will be used to help the user identify the version being saved, if it is to be restored at a later time. |
|  | User enters a label of length not to exceed 20 characters, and clicks on ‘save’.  User also has the option to ‘cancel’ the operation. | The system creates a backup of the current state of the system. The backup will include:   1. All ‘available’ assets 2. All ‘requirements’ 3. Current status of all matches in the system.   The system displays a success / failure message depending on whether the backup operation completed successfully. |
|  | The user clicks on the ‘Restore from backup’ button | The system displays a popup listing all the available backups and the timestamp of when the backup was taken.  Only backups taken in the last 15 days will be visible. |
|  | User selects the backup version to restore from and clicks on ‘Restore’.  The user also has the option to ‘cancel’ the operation. | The system completes the restore, and displays a success / failure notification.  During the restore process, the system will make the following adjustments:   1. Any new ‘Requirements’ uploaded since the time of the backup being restored will remain in the system, but the status of these will be reset to “New”, as if it had just been uploaded into the system. 2. Any matches with the following statuses will not be updated: ‘transfer initiated’, ‘transfer completed’, or ‘Exception’. |
|  | Note:   1. The backup version continues to be available for future restores if needed. 2. Saved backups will become unavailable after they are 15 days old. | |

**Appendix A: Cascade Engine Matching Process**

The system will apply the parameters specified by the user in the ‘Match Criteria’ screen when the Cascade Match is run.

The matching process will go through the following logic for each ‘requirement’ and identify all assets in the available pool that satisfy the specified criteria for a match. All the matching assets will then be sorted, best match first, and based on the number of results the user wishes to see for each Requirement, the system will display these in the ‘Match Results’ screen.

# 'Matching process’ steps applied to each requirement

## Apply State filter:

For the ‘target state’ where the cascading vehicle needs to go, lookup the ‘source states’ from where an incoming vehicle will be allowed. If the ‘available vehicle’ is from one of the allowed states, or if the vehicle has been in one of the allowed states in the past [feasibility of this is to be confirmed], continue to the next filter; if not, reject this vehicle and move on to the next ‘available’ vehicle.

The following table specifies the ‘source’ states from which vehicles can be allowed for each ‘target’ state. ‘Cost Factor’ is a numerical value that will be used for ranking different choices when multiple vehicles in the availability pool match the ‘requirements’ criteria.

**State Filter template:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr #** | **Target state** | **Allowed source state** | **Cost factor** |
| 1 | CA | CA | 5 |
| 2 | NY | NY | 5 |
| 3 | NJ | NY | 8 |
| 4 | NJ | CT |  |
| 5 | NJ | DE |  |

## Apply ‘Vehicle Type’ filter:

Look up the ‘vehicle types’ that are considered a ‘good match’ and ‘ok match’ for the type of vehicle ‘required’.

1. If the ‘available’ vehicle is of one of the types in the ‘good match’ category, add 1 to the match-score.
2. Else, if the ‘available’ vehicle is of one of the types in the ‘ok match’ category, add 0.5 to the match-score.
3. If neither of the above is true, reject this vehicle and move on to the next ‘available’ vehicle.

The following table specifies the ‘source’ vehicle-types that are ‘good’ or ‘acceptable’ matches for each ‘target’ vehicle-type.

**“Vehicle Type” Filter**

|  |  |  |
| --- | --- | --- |
| **‘Required’ bus type** | **Good match** | **Acceptable match** |
| AAD | AAD | AAS, AWD, AWS |
| AAS | AAS | AAD, AWD, AWS |
| AWD | AWD | AWS |
| AWS | AWS | AWD |
| CA | CA | DAR, DAF |
| CW | CW | DWF, DWR |
| DAF | DAF | DAR |
| DAR | DAR | DAF |
| DWF | DWF | DWR, CW |
| DWR | DWR | DWF, CW |
| MC | MC |  |
| MV | MV | VN |
| PU | PU |  |
| SUV | SUV | VN |
| MPV | MPV | SUV, VN |
| VN | VN | SUV |
| AU | AU | SUV, VN, MV |

## Apply vehicle attribute matches

### Apply MaxAge requirements:

1. If Vehicle Age >= MaxAge reject the vehicle, move to next available bus
2. If (MaxAge – Vehicle Age) is >= X, add 1 to the match-score, if not add 0.5 to the score

(X is a configurable value, default 3 years)

### Apply Vehicle Capacity check:

If (seating capacity of the ‘Available vehicle’ >= seating capacity of the ‘Required vehicle’) AND (seating capacity of the ‘Available vehicle’ – seating capacity of the ‘Required vehicle) is:

1. <= Y% of the ‘Required’ seating capacity, add 1 to match-score.
2. > Y% of the ‘Required’ seating capacity, add 0.5 to the match-score.

Else, reject the ‘available vehicle’ and move on to the next

### Apply wheel-chair capacity check:

If w/c capacity of the ‘Available vehicle’ is >= w/c capacity of the ‘Required vehicle’, add 1 to match-score, and if w/c capacity of the ‘Available vehicle’ is 1 less than the w/c capacity of the ‘Required vehicle’, add 0.5 to the match-score, provided there is at least one wheel-chair slot on the bus (a request for a bus with wheel-chair cannot be satisfied with a bus with no w/c capacity).

Else, reject the vehicle and move on to the next vehicle in the available pool.

If no w/c capacity is requested, add 1 to the match-score.

### Apply ‘Brake-type’ check:

If the brake-type of the ‘Required vehicle’ exactly matches the brake-type of the ‘available vehicle’, add 1 to match-score.

If brake-type of ‘Required vehicle’ is ‘Hydraulic’ and the ‘Available vehicle’ has ‘Air’ brake, or vice versa, add 0.5 to match-score.

If no brake type is specified in the requirement, add 1 to match-score.

### Apply ‘Fuel-type’ check:

If the fuel-type of the ‘Required vehicle’ exactly matches the fuel-type of the ‘available vehicle’, add 1 to match-score.

If fuel-type of ‘Required vehicle’ is ‘Gas’ and the fuel type of the ‘Available vehicle’ is ‘Diesel’, or vice versa, add 0.5 to match-score.

If fuel type of the ‘Required vehicle’ is CNG or LPG and the match with the ‘Available vehicle’ is not exact, reject the vehicle and move on to the next vehicle in the ‘available pool’.

If no fuel type is specified in the requirement, add 1 to match-score.

### Apply A/C check:

Determine if the ‘required bus’ needs A/C based on attributes of the ‘requirement’

If the ‘available bus’ does not have A/C, add to the retrofit cost, based on the type of the bus. If this pushes the total retrofit cost over the limit specified in the ‘Cascade Run’ parameters, reject the vehicle and move on to the next bus in the availability pool.

The following table specifies the vehicle attributes and the criteria for what is to be considered a ‘good’ or ‘acceptable’ match for each of these criteria.

**Vehicle attribute match criteria:**

| **Sr #** | **Parameter** | **Good match** | **OK match** | **Comments** |
| --- | --- | --- | --- | --- |
| 1 | Vehicle age | MaxAge - Vehicle age >=X | MaxAge - Vehicle age <X and >=1 | Vehicle age = current year - model year X will have a default value of 3 years |
| 2 | Seating capacity | Available capacity - Required capacity >= Y% of Required capacity | Available capacity >= Required capacity | Y will have a default value of 10% |
| 3 | W/C capacity | Available w/c capacity - Required w/c capacity >= 0 | Available w/c capacity - Required w/c capacity = Z | Z will have a default value of -1 (ie., if the Available capacity is 1 less than the Required capacity, it is manageable)  W/C capacity has to be more than 1 as a requirement for this. |
| 4 | Brake type | If Available Brake type = Required brake type | (Available = Air) and (Required = Hydraulic) OR (Available = Hydraulic) and (Required = Air) |  |
| 5 | Fuel type | If Available Fuel type = Required Fuel type | (Available = Gas) and (Required = Diesel) OR (Available = Diesel) and (Required = Gas) | Gas and Diesel are interchangeable, LPG, CNG etc must be exact match |
| 6 | A/C | Present if required | Not present | If A/C required and is not on the bus add to the retrofit cost based on the type of bus. |

**Retrofit Cost Template**

The following template will be used for the computation of ‘Retrofit cost’ when certain attributes are not met in the ‘available’ bus but can be added as a retrofit with incremental cost.

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Qualifier** | **Cost** |
| A/C | Vehicle type = A | $4,000 |
| A/C | Vehicle type = C | $8,500 |
| A/C | Vehicle type = D | $9,500 |
| Stepwell heater | Vehicle type = C | $2,500 |
| Rear/Mid-ship Heater | Vehicle type = C | $2,500 |
|  |  |  |
|  |  |  |

## Apply State Specifications

The following template will be used to specify ‘State Specifications’ that must be met by every incoming vehicle into a state as part of the Cascade process. If the ‘available’ bus is from the same state as the ‘required’ bus, the ‘State Specifications’ step will be skipped.

In the following template:

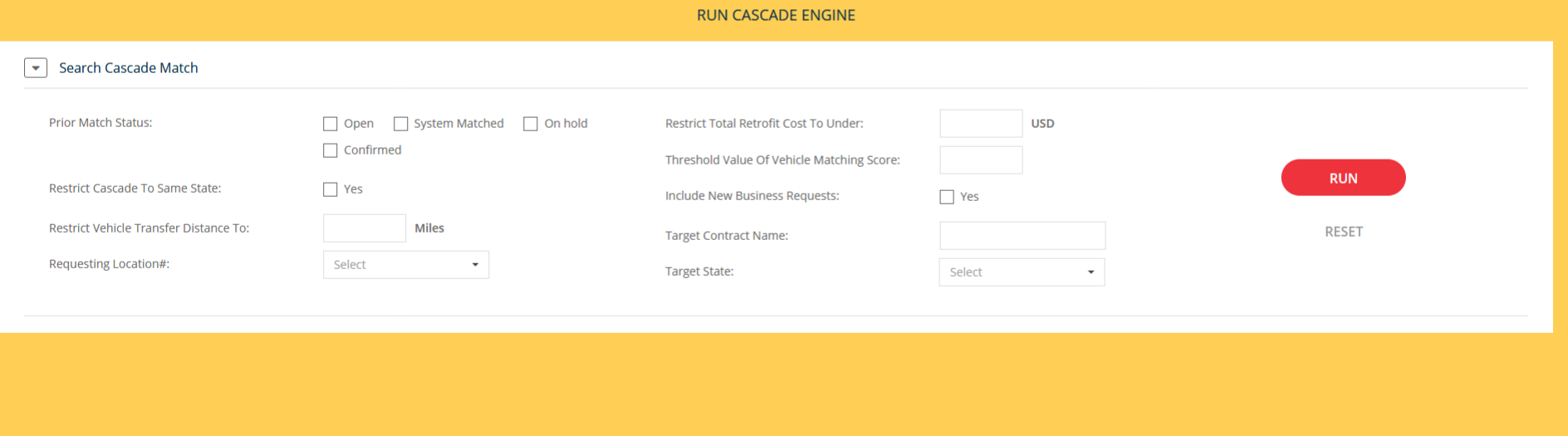
* The ‘attribute’ column represents the vehicle attribute that needs to be taken into consideration
* The ‘required value’ indicates the value to be met for the corresponding attribute
* The ‘qualifier’ attribute indicates what types of vehicles the corresponding state specification applies to. This will vary by state, and are represented in the respective ‘state’ column. If a state has no mandated specification for a given attribute, the column for that state/attribute will be blank.

**State Specification Template:**

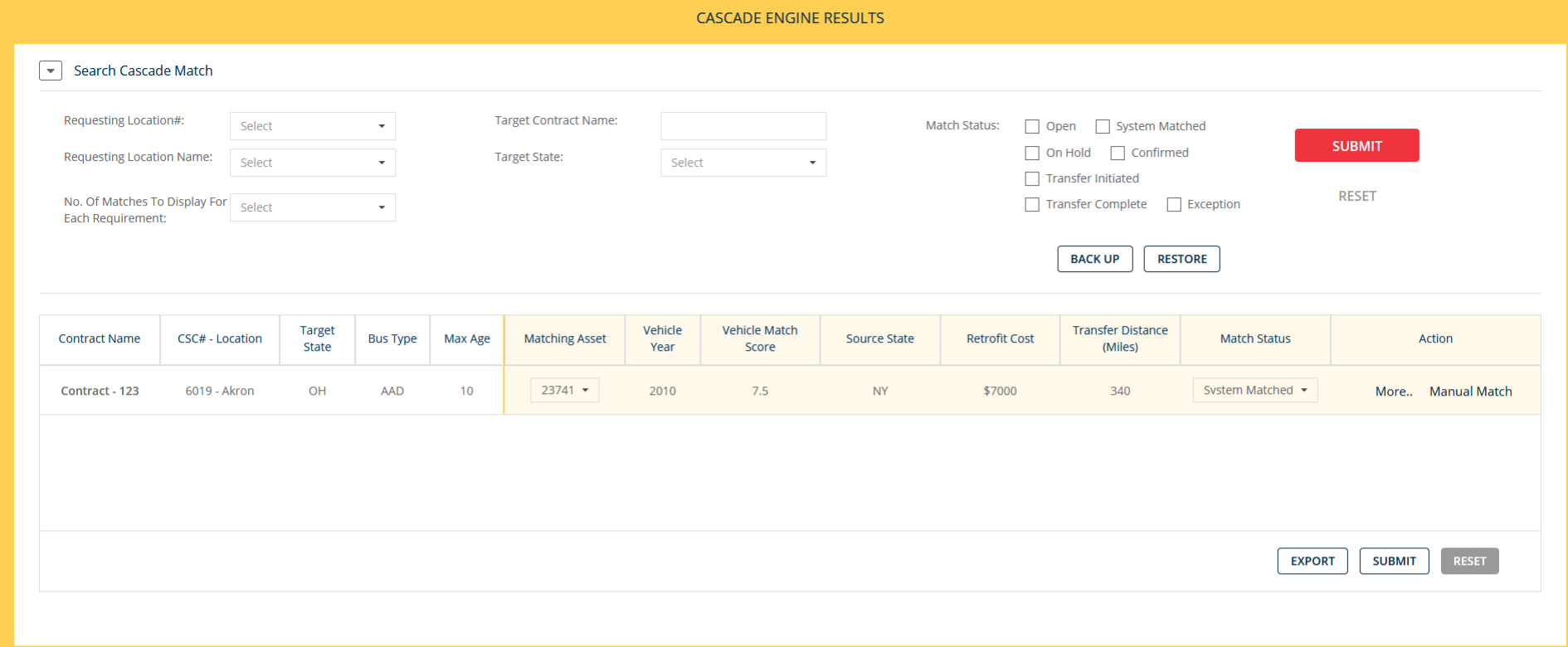
|  |  |  | **States** | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Attribute** | **Required value** | **Qualifier** | **AL** | **AZ** | **LA** | **OR** | **FL** | **…other states** |
| A/C | Present | Route type | Sped | All |  |  |  |  |
| Backup system | Present | Model year |  |  | All | 2007 |  |  |
| Brake | Air | Vehicle type |  |  |  |  |  |  |
| Brake | Air | Model year |  |  |  |  |  |  |
| Brake | Air | Capacity |  |  |  |  | 41 |  |
| Crossing arm |  |  |  |  |  |  |  |  |
| Door |  |  |  |  |  |  |  |  |
| Emergency exit/door |  |  |  |  |  |  |  |  |
| Fire suppression system |  |  |  |  |  |  |  |  |
| Heater |  |  |  |  |  |  |  |  |
| Lift |  |  |  |  |  |  |  |  |
| Plywood floor thickness |  |  |  |  |  |  |  |  |
| Seatbelts |  |  |  |  |  |  |  |  |
| Seat height |  |  |  |  |  |  |  |  |
| Windows |  |  |  |  |  |  |  |  |

# UI Mockups related to Cascade Engine

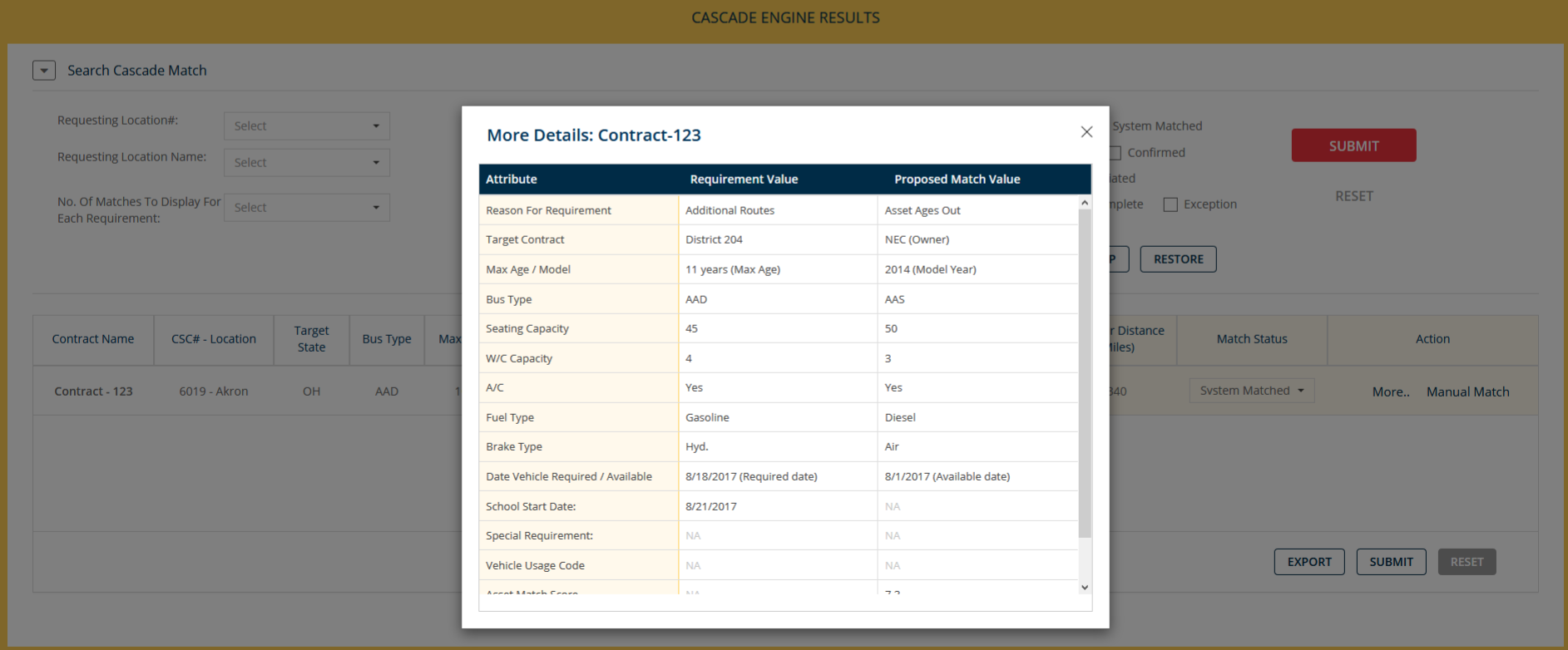
### UI screen to initiate Cascade Run

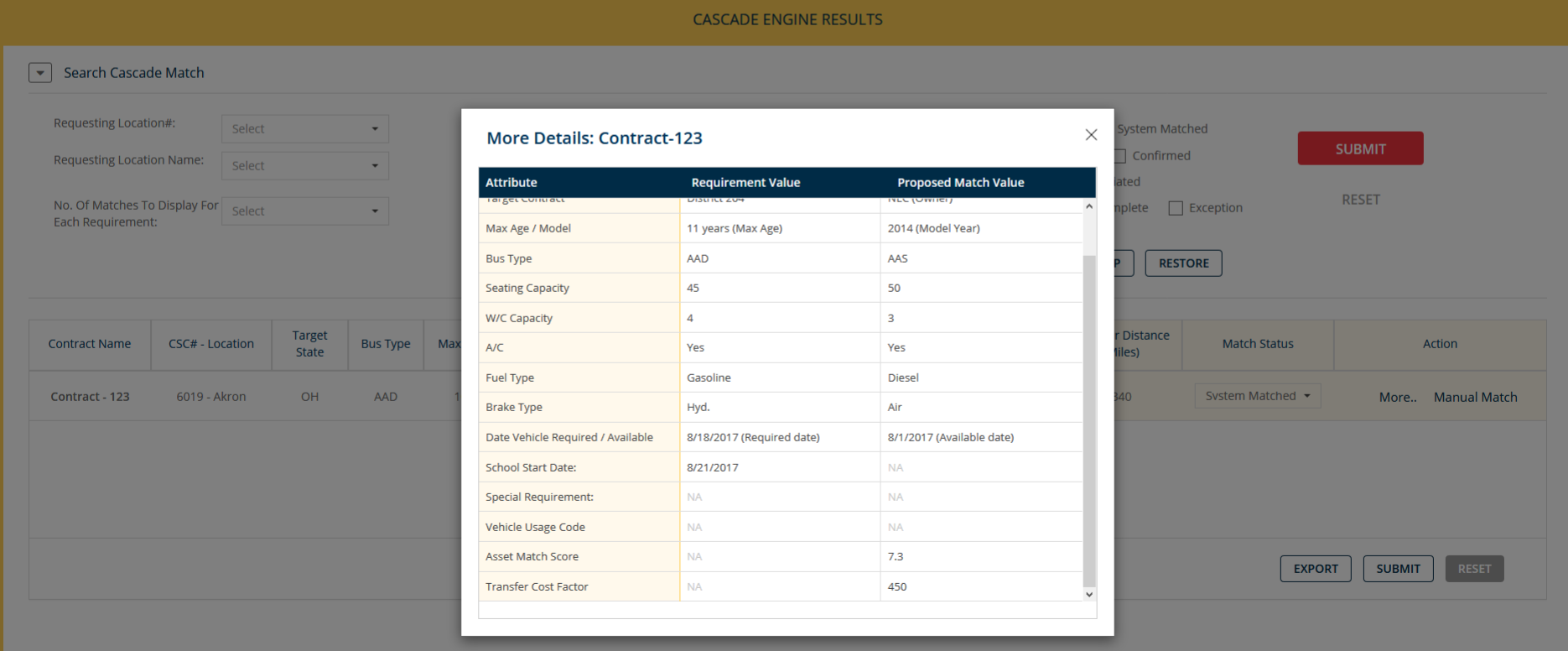


### UI to Review Cascade Results



### Pop-up view, ‘More’ button to view details of ‘Required’ and ‘Proposed’ assets

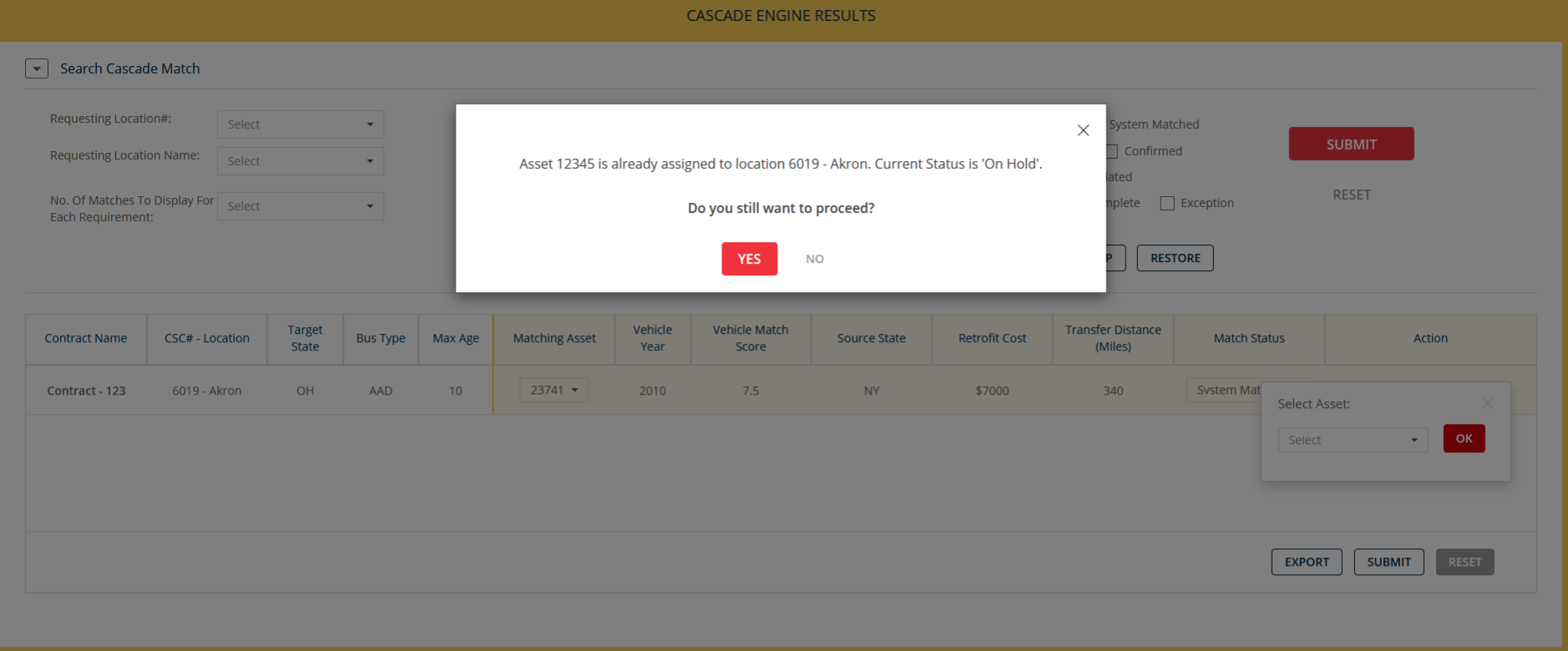




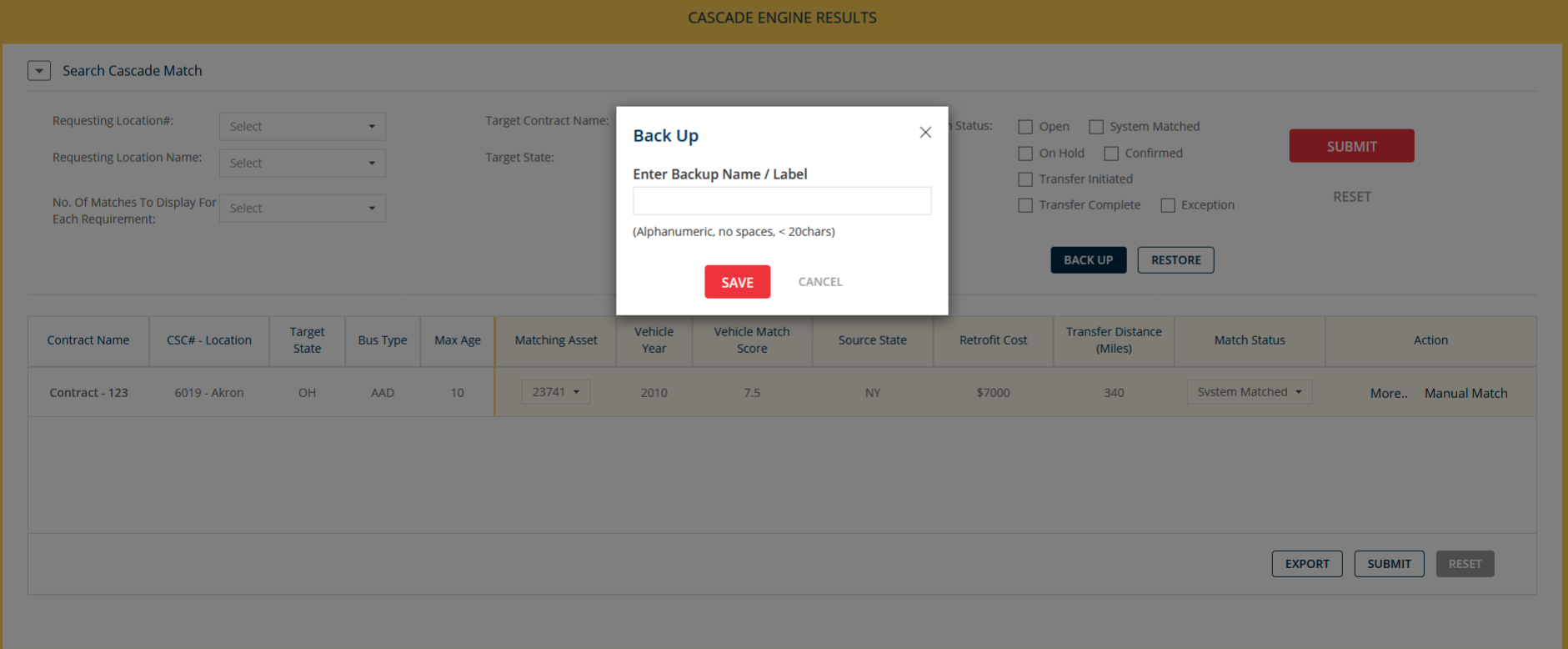
### Manual match

# 

### Confirmation screen, Manual match



### Initiate Back-up



### Restore from Back-up

