## State Machine Diagram Explanation 🡪 remove quotation marks

### Login

In addition to the entry and exit conditions, the *login* form has internal *update* and *submit* events that causes internal transitions. These *updates* include changing the *email*, *password*, and *accessCode* fields, while *submit* queries the *KTCS\_Member* and *Admin* tables to validate login credentials.

### Homepageform

In addition to the entry and exit conditions, the *Homepageform* form has internal *update*, *locationDropdown*, and *resCheck* events that causes internal transitions. These *updates* include changing the *startDate*, *endDate*, and *parkingLocation* fields while trying to reserve a car. *locationDropdown* queries the *Parking\_Locations* table to return all locations, while *resCheck* queries the *Reservations* relation to delete any tuples that have an end date smaller than the current timestamp.

### homepageQueryLocations

In addition to the entry and exit conditions, the *homepageQueryLocations* form has an internal *locCheck* event that causes internal transition. *locCheck* queries and shows KTCS\_Locations.

### memberRentalHistory

In addition to the entry and exit conditions, the *memberRentalHistory* form has an internal *rentalCheck* event that causes internal transition. *rentalCheck* queries and shows Car\_Rental\_History for the logged in *MemberID*.

### pdform

In addition to the entry and exit conditions, the *pdform* form has internal *pickup, dropoff, resCheck, noReservation,* and *completeReservation* events that cause internal transitions. *pickup* allows the user to update the odometer and pickup status upon filling out the pickup form, while *dropoff* allows the user to update the odometer, dropoff status, rating, and comment upon filling out the dropoff form. *resCheck* queries the *Reservations* relation to check if there are any reservations for the logged in *MemberID*, *noReservation* indicates to the user that they have no current reservation, while *completeReservation* indicates to the user that their reservation forms have already been completed.

### pdhandle

In addition to the entry and exit conditions, the *pdhandle* form has internal *pickup* and *dropoff* events that cause internal transitions. *Pickup* queries *Car\_Rental\_History* for the correct car info for the user, while *dropoff* queries *Car\_Rental\_History* and *Rental\_Comments* to return the relevant car info for the user and allow for the user to add comments.

### homepageHandle

In addition to the entry and exit conditions, the *homepageHandle* form has an internal *availableCars* event that causes an internal transition. *availableCars* queries the *Car* and *Reservations* relations to return any available cars and their information to the user.

### reservationHandle

In addition to the entry and exit conditions, the *reservationHandle* form has an internal *createRes* event that causes an internal transition. *createRes* queries the *Reservations* and *Payment\_History* relations to insert the reservation and the new *Payment\_History* object.

### adminHomepage

In addition to the entry and exit conditions, the *adminHomepage* form has internal *update*, *monthlyInvoiceDropdown, rentalHistoryDropdown, availableCarDropdown, maxMinRentals,* and rentalCommentDropdown events that cause internal transitions. *update* allows the admin to change the *invoiceMemberID, invoiceStartDate, invoiceEndDate, rentalHistoryCar, availableCarLocations, distanceSinceMaintenance, rentalOnDay,* and *rentalCommentMemberID*. The *monthlyInvoiceDropdown* queries the *KTCS\_Member* relation to allow the admin to select a member to view their monthly invoice, *rentalHistoryDropdown* queries the *Car* relation to allow the admin to select a car to view its rental history, and *availableCarDropdown* queries the *Parking\_Locations* relation to allow the admin to view all available cars and reservations, if any, at a selected location. Additionally, *maxMinRentals* queries the *Car\_Rental\_History* and *Car* relations to view the cars that have the most and least amount of rentals, while *rentalCommentDropdown* queries the *KTCS\_Member* relation to view all comments made by a certain user.

### availableCarsAtLocation

In addition to the entry and exit conditions, the *availableCarsAtLocation* form has an internal *carAndReservationInfo* event that causes an internal transition. *carAndReservationInfo* queries the *Car* and *Reservations* relations to view all available cars at a location and reservations for them, if any.

### monthlyInvoiceHandle

In addition to the entry and exit conditions, the *monthlyInvoiceHandle* form has an internal *checkPayments* event that causes an internal transition. *checkPayments* queries the *Payment\_History* relation to view all payment objects for a given member.

### addCarForm

In addition to the entry and exit conditions, the *addCarForm* form has internal *locationDropdown* and *update* events that cause internal transitions. *locationDropdown* queries the *Parking\_Locations* relation to view all locations to add the new car to, while *update* allows the admin to change the *VIN*, *make, model, year, location, colour, picture,* and *rentalFee* fields of the added car.

### addCarHandle

In addition to the entry and exit conditions, the *addCarHandle* form has an internal *insertCar* event that causes an internal transition. *insertCar* queries the *Car* relation to insert the provided new car information into the table.

### carRentalHistoryHandle

In addition to the entry and exit conditions, the *carRentalHistoryHandle* form has an internal *displayRentals* event that causes an internal transition. *displayRentals* queries the *Car\_Rental\_History* relation to return all rental history on the selected car.

### CarsDistanceSinceMaintenanceHandle

In addition to the entry and exit conditions, the *CarsDistanceSinceMaintenanceHandle* form has an internal *carsTravelledSinceUserEnteredDistance* event that causes an internal transition. *carsTravelledSinceUserEnteredDistance* queries the *Car\_Maintenance\_History* , *Car\_Rental\_History*, and *Car* relations to return all cars that have travelled more than the user-entered distance since their last maintenance.

### DamagedRepairCarsHandle

In addition to the entry and exit conditions, the *DamagedRepairCarsHandle* form has an internal *viewBrokenCars* event that causes an internal transition. *viewBrokenCars* queries the *Car\_Rental\_History*, *Car\_Maintenance\_History,* and *Car* relations to return all cars that have been ‘damaged’ or ‘not running’ and have not yet been maintenance to repair them.

### RentalsOnDayHandle

In addition to the entry and exit conditions, the *RentalsOnDayHandle* form has an internal *viewReservationsOnDate* event that causes an internal transition. *viewReservationsOnDate* queries the *Reservations* relation to view all reservations on scheduled during the selected date.

### rentalCommentHandle

In addition to the entry and exit conditions, the *rentalCommentHandle* form has an internal *viewRentalComments* event that causes an internal transition. *viewRentalComments* queries the *Rental\_Comment* relation to view all rental comments made by a user.

### rentalCommentObj

In addition to the entry and exit conditions, the *rentalCommentObj* form has an internal *update* event that causes an internal transition. *update* allows the admin to edit a reply comment to a user’s comment that has yet to be responded to by an admin.

### insertReplyComment

In addition to the entry and exit conditions, the *insertReplyComment* form has an internal *updateReplyComment* event that causes an internal transition. *updateReplyComment* queries the *Rental\_Comment* relation to insert the admin’s reply comment to the user into the relation.

### Register

In addition to the entry and exit conditions, the *Register* form has internal *update* and createUser events that cause internal transitions. *update* allows the new user to enter their information, namely their *firstName, lastName, address, postalCode, city, province, country, phoneNumber, email, driverLiscense,* and *creditCard.* *createUser* queries the *KTCS\_Member* relation to insert the new user object and their information into the relation.

### logout

In addition to the entry and exit conditions, the *logout* form has an internal *clearSession* event that causes an internal transition. *clearSession* destroys the current session that was running while the user or admin was logged in.