* RateCompany ---=====-------
  + Why: It is important for customers to rate the company that they deal with to let the other customers to know more about this company and the quality of its services.
  + Expected Input: Company name, Rate.
  + Expected Output: Rate value specified for the selected company.
  + Interacts To: Customer
  + Process: The customer asks the system to rate a company and enters the company name and the rate that he decides after finishing the trip and then the system save that rate to the company to be visible for all customers
* SendMessage
  + Why: The users should be able to interact with each other by sending message and responding to it.
  + Expected Input: Written message text.
  + Expected Output: A hint to the users that the message is sent.
  + Interacts To: User.
  + Process: The user asks the system to send a new message and then enter the message and choose the other user to be sent to.
* ShareTrip
  + Why: The customer may need to share a trip with other customers “Friends” to attend this trip with him/her .
  + Expected Input: Trip.
  + Expected Output: A shared trip between customers that related to the sharer customer.
  + Interacts To: Customer.
  + Process: The customer asks the system to share a trip and choses it then the system shares it between the customers related to that sharer customer.
* ReserveTrip
  + Why: The user must be able to reserve a trip as he wants to attend.
  + Expected Input: Trip that the customer specifies.
  + Expected Output: Notification with a successful reservation for the trip.
  + Interacts To: Customer.
  + Process: The customer asks the system to reserve a trip after viewing its details then the system make the customer able to pay the fees and reserve a seat for the customer on the trip.
* PayFees
  + Why: The customer must pay the fees for the trips that he/she reserves.
  + Expected Input: Trip that the user reserved.
  + Expected Output: Notification with a successful pay and notification from the bank.
  + Interacts To: Customer.
  + Process: After the customer chooses a trip to attend he/she asks the system to pay the fees for this trip and complete the reservation, Then the system deals with the bank to withdraw the price of the trip from the user’s account.
* ReceiveMessage
  + Why: The user should be able to receive the messages that other users sent to him/her/it to v be able to view and respond to it.
  + Expected Input: User’s id.
  + Expected Output: A list with all messages that had sent to this user.
  + Interacts To: User.
  + Process: The system receive the messages sent to a specific user and make him/her/it able to view and respond to it.
* SearchForTrip ------=====------
  + Why: The customer must be able to search for a trip and view all trips related to it and also be able to sort these trips with a specific type such as price, period, location, date.
  + Expected Input: Search key, Sort type.
  + Expected Output: A list with all trips related to the trip that the user has searched for.
  + Interacts To: Customer.
  + Process: The user asks the system to search for a trip by entering a search key and sort type then the system lists all trips that match with this key and sorted by the type that the user specifies.
* SignUp
  + Why: The user must be able to sign up to the system to be able to use its functions
  + Expected Input: User name, email, password, phone number, ...
  + Expected Output: Notification with a successful Sign up and ability to log in and use the system.
  + Interacts To: User.
  + Process: The user asks the system to sign up and then enters information about him/her/it self the the system checks these information and then send a notification to the user with the successful sign up and log the user in to the system.
* SignIn
  + Why: Necessary feature enables the user to login to ethe system
  + Expected Input: userName, password
  + Expected Output: Confirmation Message
  + Interacts to: Customer or Company
  + Process: Receives the parameters from The View working with the model to search the database for the userName and returns true if found and the password matches else returns false
* SignUp
  + Why: Necessary feature gives the system the ability to add new users
  + Expected Input: User information like: name, userName, password, email ...
  + Expected Output: Confirmation Message
  + Interacts to: Customer or Company
  + Process: Receives the parameters for the View sends them to the model to add new user to the database
* ViewTrips
  + Why: enables the user to view all trips on the system
  + Expected Input: None
  + Expected Output: List of all trips
  + Interacts to: Customer or Company
  + Process: Calls the TripModel to get all Trips from the database sort and return them to the view
* ExpandTrip
  + Why: enables the user of view trip’s details
  + Expected Input: tripId
  + Expected Output: List of Details
  + Interacts to: Customer or Company
  + Process: Calls the TripModel to get all details about the trip prepare the and return them to the view
* AddTrip
  + Why: enables a Company of adding new trip
  + Expected Input: trip details
  + Expected Output: confirmation message
  + Interacts to: Company
  + Process: gets the parameters from the View and sends them to the Model to store them
* EditTrip -----++++------
  + Why: enables a Company of editing/updating trip detais
  + Expected Input: tripId, tripDetails
  + Expected Output: confirmation message
  + Interacts to: Company
  + Process: Gets the parameters for the view and sends them to the model to update the trip details CAN ADD OFFER
* CancelTrip
  + Why: enables a Company of canceling trip but still viewable for the users
  + Expected Input: tripId , cancelMessage
  + Expected Output: Confirmation Message
  + Interacts to: Company
  + Process: gets a tripId and mark it as canceled and add the cancel message to the trip details
* RemoveTrip
  + Why: enable a Company of deleting a trip
  + Expected Input: tripId
  + Expected Output: Confirmation message
  + Interacts to: Company
  + Process: gets a trip id and calls the TripModel to delete the trip from the database
* Report
  + Why: enable a user of Reporting other user
  + Expected Input: userId, reportMessage
  + Expected Output: confirmation Message
  + Interacts to: Customer or Company
  + Process: gets a reported user id and calls ReportModel to add new report with the report message