

Initial Documentation: LoadSmart Final Project

GOALS

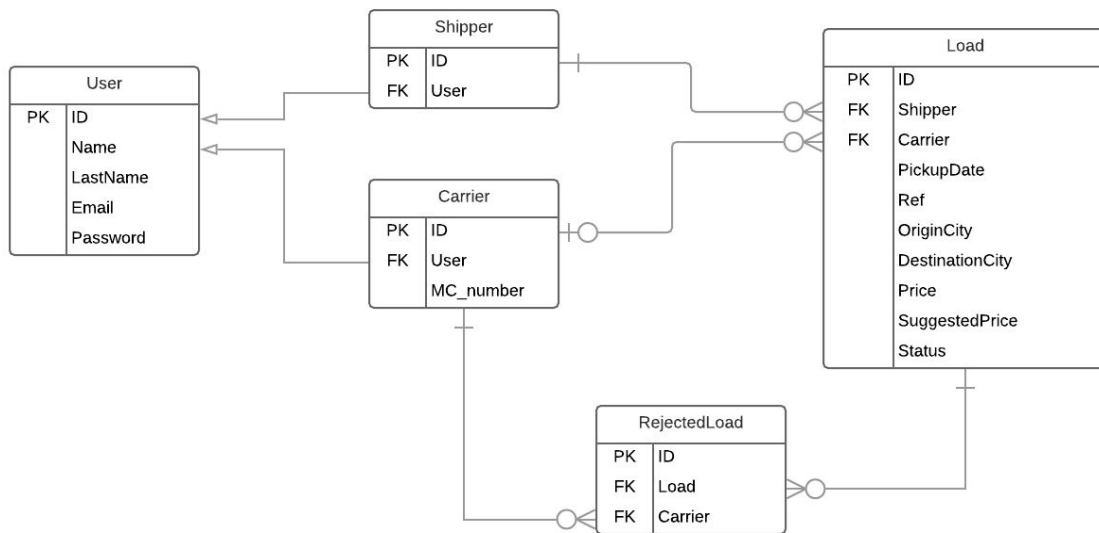
This project aims to create a platform that on one side allows a Shipper/ Freight Broker to post loads that they need to find a truck for (eg New York, NY - Albany, NY), and on the other side allows Carriers to have access to these loads and accept them.

PROPOSAL

This project can be divided into 4 modules:

1. Login/Registration
2. Shipper
3. Carrier
4. API

DATABASE DRAFT



FUNCTIONAL REQUIREMENTS

1. Login/Registration

RF001 > The system shall allow a Shipper to register.

RF001_1 > The system shall allow the Shipper to insert the following information: Name, Last Name, Email and Password.

RF002 > The system shall allow a Carrier to register.

RF002_1 > The system shall allow the Carrier to insert the following information: Name, Last Name, MC Number, Email, Password.

The image shows two side-by-side registration forms under the heading "Select your profile" with the subtext "Are you a Shipper or a Carrier?".

Left Form (Shipper):

- Two profile options: "I'M A SHIPPER looking for a truck" (with a green checkmark) and "I'M A CARRIER looking for a load".
- Input fields: First name, Last name, Email, Password.
- Green button: "SIGN UP FOR FREE".

Right Form (Carrier):

- Two profile options: "I'M A SHIPPER looking for a truck" and "I'M A CARRIER looking for a load" (with a green checkmark).
- Input fields: First name, Last name, Email, Password, MC number.
- Green button: "SIGN UP FOR FREE".

RF003 > The system shall allow a user to login.

RF003_1 > The system shall allow the user to insert the Email and Password to login.

RF003_2 > The system shall redirect the user to the respective page (Shipper page or Carrier page) according to the user profile (Shipper or Carrier).

The image shows a login form with a grey header containing the word "LOGIN".

- Input fields: Email, Password.
- Green button: "LOG IN".

2. Shipper

RF001 > The system shall allow a Shipper to add a new load.

RF001_1 > The system shall allow the Shipper to insert the following information when adding a new load: Pickup date, Ref #, Origin City, Destination City, Price.

RF001_1_1 > The system shall autocomplete the names of the Origin City and Destination City while the user is typing, using the Google Maps API.

RF001_2 > The system shall show the Suggested Price from Data Science Model.

RF001_2_1 > The system shall calculate the Suggested Price based on the distance in miles between the Origin City and Destination City: \$1.50 per mile.

RF001_3 > The system shall allow the user to post the load by clicking in a button.

RF001_3_1 > The system shall add the load information into the database, with Status “available”.

RF002 > The system shall allow a Shipper to view all of his available loads.

RF002_1 > The system shall allow the Shipper to view the following information of the available loads: Pickup date, Ref #, Origin City, Destination City, Status, Price.

RF002_2 > The system shall have buttons to Change Price (**RF003**) or Cancel (**RF004**), for each one of the available loads.

AVAILABLE LOADS						
PICK UP	REF. #	ORIGIN	DESTINATION	STATUS	PRICE	
07/12	2375656	Tampa, FL	Newark, NJ	Load Available	\$1,350	Edit Rate Cancel
07/12	234274HF	Atlanta, FL	Miami, FL	Load Available	\$2,400	Edit Rate Cancel
07/12	2375656	Tampa, FL	Newark, NJ	Load Available	\$1,400	Edit Rate Cancel
07/13	0986746FH	Atlanta, GA	Miami, FL	Load Available	\$1,350	Edit Rate Cancel
07/14	28377AFH	New York, NY	Dallas, TX	Load Available	\$2,400	Edit Rate Cancel

RF003 > The system shall allow a Shipper to change the price of his loads.

RF003_1 > The system shall allow the Shipper to change the price for the available loads, after a Carrier accepts the load it is not possible to change the price.

RF004 > The system shall allow a Shipper to Cancel an available load.

RF004_1 > The system shall remove this load from the available loads list.

RF004_2 > The system shall modify the database by deleting this Load.

RF005 > The system shall allow a Shipper to view all of his accepted loads.

RF005_1 > The system shall allow the Shipper to view the following information of the accepted loads: Pickup date, Ref #, Origin City, Destination City, Status, Price.

RF005_1_1 > The system shall show the Status information in this format: “Accepted by *Carrier Name*”.

RF005_2 > The system shall have buttons to Cancel (**RF006**), for each one of the accepted loads.

ACCEPTED						
PICK UP	REF. #	ORIGIN	DESTINATION	STATUS	PRICE	
07/12	23884764	New York, NY	Albany, NY	Accepted by Argo Transport	\$650	Cancel
07/14	756037521	Newark, NJ	Chicago, IL	Accepted by John's Trucking	\$900	Cancel
07/15	847502927	Newark, NJ	Los Angeles, CA	Accepted by T1. Transport	\$2,800	Cancel

RF006 > The system shall allow a Shipper to Cancel an accepted load.

RF006_1 > The system shall remove this load from the accepted loads list.

RF006_2 > The system shall modify the database by deleting this Load.

3. Carrier

RF001 > The system shall allow a Carrier to view all available loads.

RF001_1 > The system shall allow the Carrier to view the following information of the available loads: Pickup date, Origin City, Destination City, Shipper's Name, Status, Price.

RF001_1_1 > The system shall show to the Carrier 95% of the price inserted by the Shipper, because the Loadsmart's margin should be 5% of the total value.

RF001_2 > The system shall have buttons to Reject (**RF002**) and Accept (**RF003**) a Load, for each one of the available loads.

RF001_3 > The system shall allow the Carrier to view only the available loads that were not rejected by this Carrier.

RF002 > The system shall allow a Carrier to reject an available load.

RF002_1 > The system shall not show the rejected load on the available loads list for this Carrier.

RF002_2 > The system shall show the rejected load on the rejected loads list (**RF004**) for this Carrier.

RF002_3 > The system shall modify the database by creating a RejectedLoad object.

RF003 > The system shall allow a Carrier to accept an available load.

RF003_1 > The system shall not show the accepted load on the available loads list for this Carrier.

RF003_1 > The system shall show the accepted load on the accepted loads list (**RF005**) for this Carrier.

RF003_2 > The system shall modify the database by changing the load "Status" attribute to "accepted".

RF004 > The system shall allow a Carrier to view all of his rejected loads.

RF004_1 > The system shall show the rejected loads list for this Carrier.

RF005 > The system shall allow a Carrier to view all of his accepted loads.

RF005_1 > The system shall allow the Carrier to view the following information of the accepted loads: Pickup date, Origin City, Destination City, Shipper's Name, Status, Price.

RF005_2 > The system shall have buttons to Drop the Load (**RF006**), for each one of the accepted loads.

RF006 > The system shall allow a Carrier to drop an accepted load.

RF006_1 > The system shall not show the dropped load on the available loads list for this Carrier.

RF006_2 > The system shall show the dropped load on the rejected loads list (**RF004**) for this Carrier.

RF006_3 > The system shall modify the database by creating a RejectedLoad object.

RF006_4 > The system shall modify the database by changing the load "Status" attribute to "available", this way other Carriers can view the load on their available loads list.

4. API

Shipper's API

RF001 > The API shall require the user to authenticate itself.

RF002 > The API shall allow the user to post a load.

RF003 > The API shall allow the user to view all available loads.

RF004 > The API shall allow the user to view all accepted loads.

Carrier's API

RF001 > The API shall require the user to authenticate itself.

RF002 > The API shall allow the user to accept a load.

RF003 > The API shall allow the user to reject a load.

RF004 > The API shall allow the user to drop a load.

RF007 > The API shall allow the user to view all rejected loads.

