Everything Roadster

Software Requirements Specifications

University of British Columbia Okanagan

Cosc 304 - Winter 2015

Team Members & Contact Information

Table Of Contents

1 Introduction

EverythingRoadster.ca is an e-commerce site specializing in the growing roadster market. The site will allow users to buy and sell Roadster accessories and parts in an online format. It consists of a mission statement, design perspective, preliminary analysis and constraints that EverythingRoadster might encounter.

1.1 Purpose

The purpose of this document serves to outline the design requirements and constraints for the EverythingRoadster online site.

1.2 Mission Statement

The mission of EverythingRoadster is to design and implement a clean and efficient e-commerce site to link buyers and sellers of motor vehicle parts in a safe and secure manner; all sellers will consist of approved factory suppliers. The system handles processing and tracking of orders, maintains a list of parts and the vehicles they are compatible with, and provides a means for customers to give feedback on the quality of their experience with a vendor.

1.3 Executive Summary

EverythingRoadster is a website designed to help customer and vendors of vehicle parts connect with one and other. Vendors have the ability to create listings that describe the price and quantity of the parts they are selling. Customers can then browse a complete list of parts that are currently for sale, or use searching and filtering tools to find the exact part they are looking for.

The typical workflow of a vendor begins with a request to become a vendor. EverythingRoadster will review the request and, provided that the vendor meets all expectations, will provide a vendor account to the vendor. With this account they will be able to specify which parts they current have for sale, the quantity of each part, and the price it is selling for. If the part already exists in the database, it is likely already associated with vehicles that it is compatible with. If it is not, however, the vendor will have the option of providing a list of vehicles with which the part is compatible. When a customer adds a part to an order, the associated vendor will be notified. Once the customer places and pays for the order, the vendor will be expected to ship the part to the address provided by the customer.

To become a customer approval by EverythingRoadster is not required; anyone may sign up as a customer. Once signed up, the customer will be able to browse all listings that are currently posted, and find information about specific parts and whether or not they are compatible with the customer’s vehicle of interest. They can add any available part to their shopping cart, and place an order for the parts in their cart at any time. They will be required to pay the price of each part, as well as the shipping costs associated with it, before the order will be processed.

EverythingRoadster is responsible for managing the order and financial transaction; however, it is the responsibility of the vendor to ship the part to the customer who purchased it. If the vendor fails to ship the part, it will be the responsibility of the customer and vendor to resolve the issue. A customer can provide feedback based on the quality of service a vendor has provided. If a vendor receives enough negative feedback their association with EverythingRoadster will be reconsidered, potentially resulting in the deletion of their account.

2 Domain Assumptions

2.1 Accounts

- Each account can only be of one type - Vendor, Customer, or Admin.
- Vendors must be certified seller/suppliers - they can also buy parts as a customer.
- Each account user will use their email (unique) and password to log in to the site.
- Admin accounts will have access to other administration features hidden from customers and vendors - find users, add vehicles, add parts, etc.
- Must be from Canada or US to list and order parts.

2.2 Parts

- Parts can be listed by vendors if they are unable to find the one corresponding to their listing.
- Admins can add parts in the database without adding a listing.
- The average price of a part will be calculated from all listings created for the part otherwise it will be null or 0 shown to user.

2.3 Storage

- Since vendors not employed by us are selling their roadster parts, storage facilities and considerations are not needed.

2.4 Shipping

- There are a total of four shipping options available that all vendors and customers must use but some may not be available depending on where the part is being shipped from.
- They are: USPS, FedEx, UPS, and Canada Post.
- The ship date is 3 days after order date and it must be shipped by the vendor by then but can be earlier.
- Can only ship to and from Canada and United States.

2.5 Payments

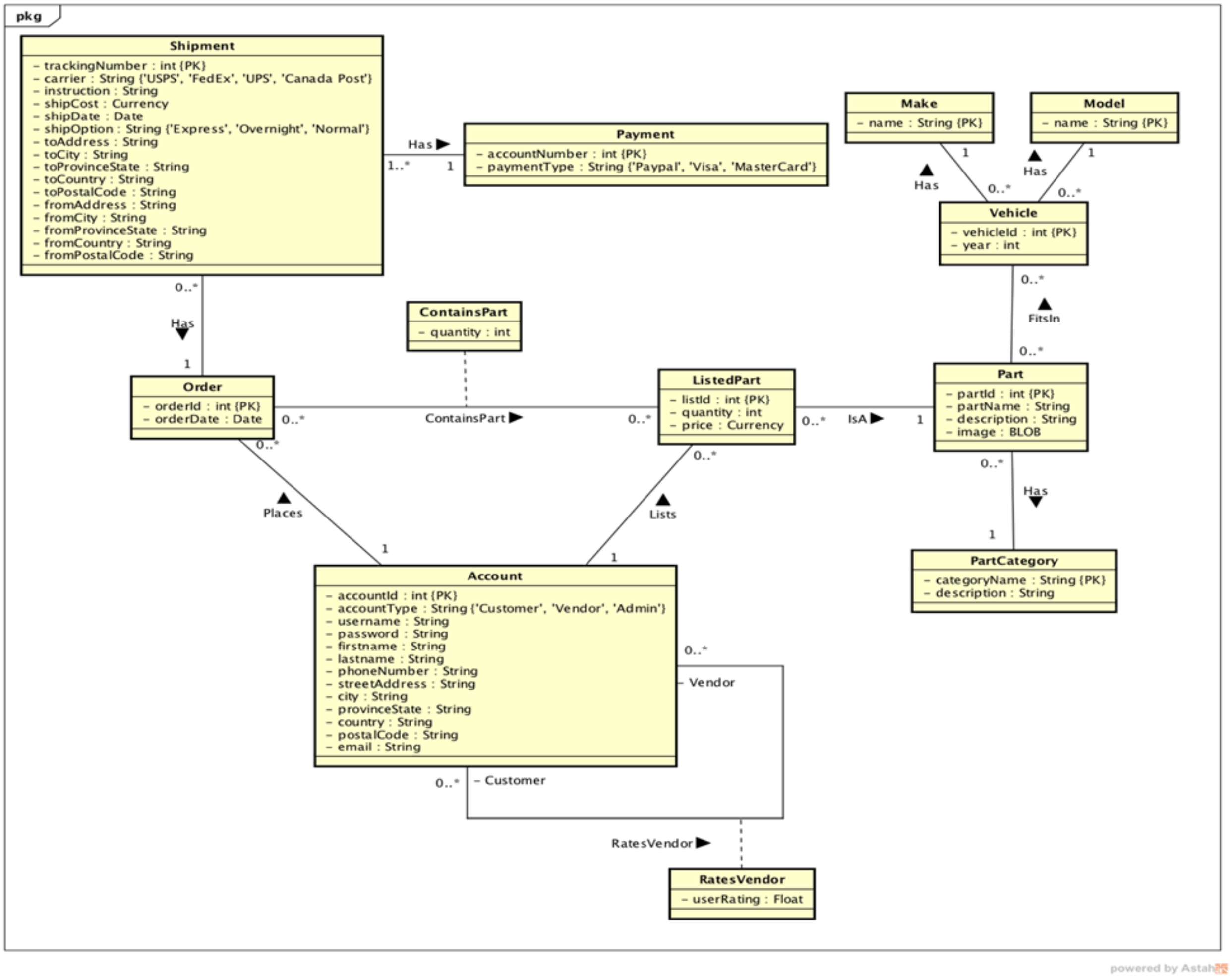
- Users will only be able to pay through Paypal, Visa, or MasterCard.
- Payments go through our site and then the vendors are paid in turn through us.
- Account numbers are held, but no other identifying information is stored for the account and the numbers will be encrypted.

2.6 Listings of parts

- Once a part has been ordered, vendors can no longer remove their listing. They can instead remove the remaining quantity if they no longer want to sell the product to anyone else
- The vendor cannot lower the quantity of available parts to a value less than the number of orders the part has been added to.
- If the vendor chooses to remove all remaining quantity, the customer will then need to request the part from the vendor.

3 Data Model Description

3.1 ER/UML Di



3.2 Relational Assumptions

Relationship	Description
Account (Customer) RatesVendor Account (Vendor)	This is a many-to-many relationship that allows one user, a customer, to rate another user, a vendor, based on the quality of their experience when purchasing a vehicle part from the vendor. The relationship consists of one attribute, userRating, which is a floating point value between 0.0 and 5.0 that stores the rating that the customer left. Customers are not required to rate the vendors they have purchased from.
Account Places Order	This is a one-to-many relationship that associates an order with the customer who placed it. A customer may not have placed any orders.
Account Lists ListedPart	This is a one-to-many relationship that associates a part that is for sale with the vendor who is selling the part. A vendor may not have listed any parts.
Order ContainsPart ListedPart	This is a many-to-many relationship between an order and parts that are included in the order. There is one attribute, quantity, stored in the relationship, which is an integer that is greater than zero, which represents the number of a given part that have been added to an order. An order

	must contain at least one part.
Shipment Has Order	This is a many-to-one relationship between a shipment of parts and the order it is associated with. An order does not need to have a shipment associated with it because the order may not have been shipped yet; however, a shipment must be associated with an order. And their may be multiple shipments per order if ordered parts are coming from different vendors
Shipment Has Payment	This is a one-to-many relationship between a shipment and the payments that are made towards it. A payment must be made towards a shipment, and a payment must be associated with a shipment.
ListedPart IsA Part	This is a many-to-one relationship between a part that a user has listed and a known part. This allows multiple listings of the same part to exist. A part must be associated with a listed part; however, a part does not need to have any listings.
Part Has PartCategory	This is a many-to-one relationship between a part and a category. A part must have an associated category; however, a category does not need to contain any parts.
Part FitsIn Vehicle	This is a many-to-many relationship between parts and vehicles. A part may not fit in any vehicles, and a vehicle may not have any parts that fit in it in the database.
Vehicle Has Make	This is a many-to-one relationship between vehicles and their make. A vehicle must have a make; however, a make does not need to have any vehicles associated with it.
Vehicle Has Model	This is a many-to-one relationship between vehicles of different years and their models. A vehicle must have a model; however, a model not need to have any vehicles associated with it.

3.3 Entity Description

3.3.1 Account

Attribute	Description
accountId {PK}	Computer generated ID that uniquely identifies a user account
accountType	User selected type for their account of value “Admin”, “Customer” or “Vendor”
email	The user account email address used to log in with as a username
password	Account password set on sign up by user
firstname	The first name of the account’s user
lastname	The last name of the account’s user
phoneNumber	The phone number of the account’s user
streetAddress	The account user’s address used for default shipping and billing address
city	The account user’s city
provinceState	Province or state where the account user lives
country	Country where account user lives
postalCode	Account user’s postal code

3.3.2 Part

Attribute	Description
partId {PK}	Computer generated ID that uniquely identifies a part

categoryName {FK}	Name of category part is grouped in - is a FK to PartCategory.categoryName
partName	Name of the part - created by vendor if part not in system already - or admin created
description	Full description of the part
image	An image showing the part

3.3.3 PartCategory

Attribute	Description
categoryName {PK}	A name to uniquely identify a grouping of parts
description	Description of the category and the parts it encapsulates

3.3.4 Vehicle

Attribute	Description
vehicleId {PK}	Computer generated ID that uniquely identifies a vehicle
makeName {FK}	Name of the make of the vehicle - is a FK to Make.name
modelName {FK}	Name of the model of the vehicle - is a FK to Model.name
year	The year of the vehicle

3.3.5 Make

Attribute	Description
name {PK}	Automobile manufacturer name that is unique

3.3.6 Model

Attribute	Description
name {PK}	Unique name of a model from a make of automobiles

3.3.7 Order

Attribute	Description
orderId {PK}	Computer generated ID that uniquely identifies an order
customerId {FK}	Account with type “Customer” who ordered these parts - is a FK to Account.accountId
orderDate	Date the order was placed

3.3.8 Shipment

Attribute	Description
trackingNumber {PK}	Auto-generated package tracking number that uniquely identifies shipment and tracks it

orderId {FK}	ID of order in shipment - is a FK to Order.orderId
carrier	Carrier used to deliver shipment of value ‘USPS’, ‘FedEx’, ‘UPS’, ‘Canada Post’
instruction	Optional instructions for the carrier
shipCost	Additional cost of sending the shipment
shipDate	Date the item will be shipped
shipOption	Options for shipment speed of value ‘Express’, ‘Overnight’, ‘Normal’
toAddress	Address shipment sent to
toCity	City shipment sent to
toProvinceState	Province or state shipment sent to
toCountry	Country shipment sent to
toPostalCode	Postal Code shipment sent to
fromAddress	Address shipment sent from
fromCity	City shipment sent from
fromProvinceState	Province or state shipment sent from
fromCountry	Country shipment sent from
fromPostalCode	Postal code shipment sent from

3.3.9 Payment

Attribute	Description
accountNumber {PK}	Account user’s credit or paypal account number
shipmentId {FK}	ID of the shipment the payment was processed for - is a FK to Shipment.trackingNumber
paymentType	Type of payment of value ‘Paypal’, ‘Visa’, ‘MasterCard’

3.3.10 ListedPart

Attribute	Description
listId {PK}	Computer generated ID that uniquely identifies a listing
vendorId {FK}	ID of vendor account who listed the part - is a FK to Account.accountId
partId {FK}	ID of part the listing is of - is a FK to Part.partId
quantity	Number of this part vendor has and lists to sell
price	Price per one part for this specific listing

3.3.11 FitsIn

Attribute	Description
partId {PK} {FK}	ID of a part - is a FK to Part.partId
vehicleId {PK} {FK}	ID of a vehicle part works for - is a FK to Vehicle.vehicleId

3.3.12 ContainsPart

Attribute	Description
orderId {PK} {FK}	ID of an order that listed part is in - is a FK to Order.orderId
listId {PK} {FK}	ID of a vendor’s listing for a part - is a FK to ListedPart.listId
quantity	Number of parts from the listing the customer bought in the order

3.3.13 RatesVendor

Attribute	Description
customerId {PK} {FK}	ID of the Customer account who rates a vendor - is a FK to Account.accountId
vendorId {PK} {FK}	ID of the Vendor account who got rated - is a FK to Account.accountId
userRating	Rating score left by customer for vendor between 0 and 5.0

3.4 Relational Schema - SQL DDL

```
CREATE TABLE Account(
accountId INTEGER,
accountType VARCHAR(10) CHECK (accountType IN
(‘Customer’, ‘Vendor’, ‘Admin’)),
email VARCHAR(254) NOT NULL UNIQUE,
password VARCHAR(25) NOT NULL,
firstname VARCHAR(30) NOT NULL,
lastname VARCHAR(30) NOT NULL,
phoneNumber VARCHAR(16),
streetAddress VARCHAR(30) NOT NULL,
city VARCHAR(20) NOT NULL,
provinceState VARCHAR(20) NOT NULL,
country VARCHAR(20) NOT NULL,
postalCode VARCHAR(8) NOT NULL,
PRIMARY KEY(accountId)
);
```

```
CREATE TABLE PartCategory(
categoryName VARCHAR(50),
description VARCHAR(MAX),
PRIMARY KEY(categoryName));
```

```
CREATE TABLE Shipment(
trackingNumber INTEGER,
orderId INTEGER,
carrier VARCHAR(25) CHECK (carrier IN (‘USPS’,
‘FedEx’, ‘UPS’, ‘Canada Post’)),
instruction VARCHAR(MAX),
shipCost DECIMAL(5,2) NOT NULL,
shipDate DATETIME NOT NULL,
shipOption VARCHAR(25) CHECK (shipOption IN
(‘Express’, ‘Overnight’, ‘Normal’)),
toAddress VARCHAR(30) NOT NULL,
toCity VARCHAR(20) NOT NULL,
toProvinceState VARCHAR(20) NOT NULL,
toCountry VARCHAR(20) NOT NULL,
toPostalCode VARCHAR(8) NOT NULL,
fromAddress VARCHAR(30) NOT NULL,
fromCity VARCHAR(20) NOT NULL,
fromProvinceState VARCHAR(20) NOT NULL,
fromCountry VARCHAR(20) NOT NULL,
fromPostalCode VARCHAR(8) NOT NULL,
paymentId INTEGER NOT NULL,
PRIMARY KEY(trackingNumber),
FOREIGN KEY(orderId) REFERENCES Order(orderId)
ON DELETE CASCADE ON UPDATE CASCADE,
FOREIGN KET(paymentId) REFERENCES
Payment(accountNumber)
ON DELETE NO ACTION ON UPDATE CASCADE);
```

```
CREATE TABLE Part(
partId INTEGER,
categoryName VARCHAR(50),
partName VARCHAR(50),
description VARCHAR(MAX),
PRIMARY KEY(partId),
FOREIGN KEY(categoryName) REFERENCES
PartCategory(categoryName)
```

```
CREATE TABLE Vehicle(
vehicleId INTEGER,
makeName VARCHAR(25),
modelName VARCHAR(25),
year INTEGER NOT NULL,
PRIMARY KEY(vehicleId),
FOREIGN KEY(makeName) REFERENCES Make(name)
ON DELETE SET NULL ON UPDATE CASCADE,
```

ON DELETE SET NULL ON UPDATE CASCADE);	FOREIGN KEY(modelName) REFERENCES Model(name) ON DELETE SET NULL ON UPDATE CASCADE);
CREATE TABLE ContainsPart(orderId INTEGER, listId INTEGER, quantity INTEGER NOT NULL CHECK (quantity > 0), PRIMARY KEY(orderId, listId), FOREIGN KEY(orderId) REFERENCES Order(orderId) ON DELETE CASCADE ON UPDATE CASCADE, FOREIGN KEY(listId) REFERENCES ListedPart(listId) ON DELETE NO ACTION ON UPDATE CASCADE);	CREATE TABLE Order(orderId INTEGER, customerId INTEGER, orderDate DATETIME NOT NULL, PRIMARY KEY(orderId), FOREIGN KEY(customerId) REFERENCES Account(accountId) ON DELETE SET NULL ON UPDATE CASCADE);

CREATE TABLE Make(
name VARCHAR(25),
PRIMARY KEY(name));
	CREATE TABLE Model(
	name VARCHAR(25),
	PRIMARY KEY(name));

CREATE TABLE RatesVendor(customerId INTEGER, vendorId INTEGER, userRating DECIMAL(2,1) NOT NULL CHECK (userRating BETWEEN 0.0 AND 5.0), PRIMARY KEY(customerId, vendorId), FOREIGN KEY(customerId) REFERENCES Account(accountId) ON DELETE CASCADE ON UPDATE CASCADE, FOREIGN KEY(vendorId) REFERENCES Account(accountId) ON DELETE CASCADE ON UPDATE CASCADE);	CREATE TABLE ListedPart(listId INTEGER, vendorId INTEGER NOT NULL, partId INTEGER NOT NULL, quantity INTEGER NOT NULL CHECK (quantity >= 0), price DECIMAL(10,2) NOT NULL CHECK (price >= 0.0), PRIMARY KEY(listId), FOREIGN KEY(vendorId) REFERENCES Account(accountId) ON DELETE SET NULL ON UPDATE CASCADE, FOREIGN KEY(partId) REFERENCES Part(partId) ON DELETE NO ACTION ON UPDATE CASCADE);
CREATE TABLE Payment(accountNumber INTEGER, paymentType VARCHAR(25) CHECK (paymentType IN (‘Paypal’, ‘Visa’, ‘MasterCard’)), PRIMARY KEY(accountNumber));	CREATE TABLE FitsIn(partId INTEGER, vehicleId INTEGER, PRIMARY KEY(partId, vehicleId), FOREIGN KEY(partId) REFERENCES Part(partId) ON DELETE CASCADE ON UPDATE CASCADE, FOREIGN KEY(vehicleId) REFERENCES Vehicle(vehicleId) ON DELETE CASCADE ON UPDATE CASCADE);

4 Site Features

4.1 Customer

4.1.1 MVP - Minimum Viable Product

- Browse for roadster parts by category or search by name or description keywords.
- See average list price for a part plus list of all listings for a part and their prices.
- Add a chosen quantity of listed parts to a shopping cart.
- Create orders, have them shipped and pay for them.
- Their history of ordered parts and shipments.
- Can create an account and change settings and their information.
- Password recovery through email.

- Ability to rate vendors that they have bought from.

4.1.2 Stretch Goals

- Request a part that currently has no listings as needed to entice vendors to list.
- Change ratings they have given a vendor or dispute disputes.
- Cancel orders, return orders for refund - at the moment not supported.
- Allow customers outside of US and Canada to order parts.

4.2 Vendor

4.2.1 MVP

- All customer options - they just have additional features.
- List a quantity of a part at the price of their choosing.
- If a part, vehicle, make, or model does not exist in the database when a vendor is creating a listing, they should be given the option to add it.
- List of their current part listings and sold parts.
- Can remove listings of parts that haven't had an order yet but can't remove the listings of sold parts.
- Able to see their average customer given rating in settings screen.

4.2.2 Stretch Goals

- See list and count of customer part requests by part.
- Dispute ratings made by customers.
- Give refund for a returned part.
- Add categories for parts and add vehicles if their isn't an appropriate one for their part.
- Allow vendors outside of Canada and US to sell parts.

4.3 Admin

4.3.1 MVP

- Search for individual customers and vendors by id or email.
- Create reports on various statistics of the site - current number of listings, sales in date range, etc.
- Change account information of a customer or vendor when requested by customer service.
- Add part data, part category data, and roadster vehicle data.

4.3.2 Stretch Goals

- Remove flagged accounts that are deemed scams and their listings.