ZOOBC

Project Design

University of British Columbia – Okanagan Campus
COSC 304 – Introduction to Databases – Prof. Dr. Ramon Lawrence – 2018

Daniella Davis

Igat Matthew

Jonathan Gresl

Wiliam Setiawan

Table of Contents

1 Introduction

1.1 Mission Statement

Our vision is to provide people all over the world with animals from areas all over the world; to build a web-based store where people can discover any animal that they might want to buy. Users should be able to create and manage their accounts. The system should also provide administrative functions for managing warehouses, products, customers, and orders.

1.2 Executive Summary

ZooBC is a web-based store which sells animals that come from all over the world. We hope to achieve in satisfying our customers by delivering animals in a timely manner and in a humane method. We facilitate purchases by ensuring that the deliveries are secured because we work only one shipping company to avoid any complications that may arise in working with multiple companies.

The *ZooBC* store provides a simple navigation for customers to peruse through the available contents. Customers can browse the website by category or quickly perform a direct search by typing in the name of the desired animal. Database administrators can maintain these transactions by logging into their account directly on the website. If the customers wish to purchase the animal, the customer must first register an account before doing so.

Customers may leave reviews on a purchased product to express their satisfaction to other customers; however, they are only allowed to leave one per item. Other features of the web-based store include multiple tools for the administrator to maintain the store. For example, an administrator will have the feature to display a graph reporting the sales of an item. An administrator can update quantities of available products to maintain consistency with the warehouses. This feature will be implemented with appropriate techniques derived from AJAX.

The website offers a payment option with 16-digit credit cards. It will have a prompt that ensures correct information is inputted. This website will support data validation by using triggers.

2 Domain Assumptions

2.1 User

- A user will log in using a unique email address.
- A customer cannot have more than one shopping cart at a time.
- Users must be logged in before they are able to add animals to a shopping cart.

2.2 Products (Animals)

- Animals will be the only product ever sold by ZooBC.
- An “Animal” refers to a product model.
- The animals for sale are mass-produced and not unique to the buyer.

2.3 Storage / Warehouses

- There will be one warehouse per continent.
- Warehouses are stocked by a 3rd party vendor unrelated to this domain.
- Each warehouse maintains a large supply of every animal offered.

2.4 Shipping

- Animal pickups at warehouses is not permitted; they can be delivered only.
- Orders are shipped to the same user who placed the order.
- The user’s profile address will be used as the default shipping address but can be edited.
- The animals are delivered directly to the order’s shipping address.
- Shipments are handled by one 3rd party shipping company for all countries.
- Ground and air transportation options are available for all animals of all sizes.
- **Shipping costs are calculated for each animal added and based on the animal size.**
- The health of the animals is maintained by the shipping provider.

2.5 Payment

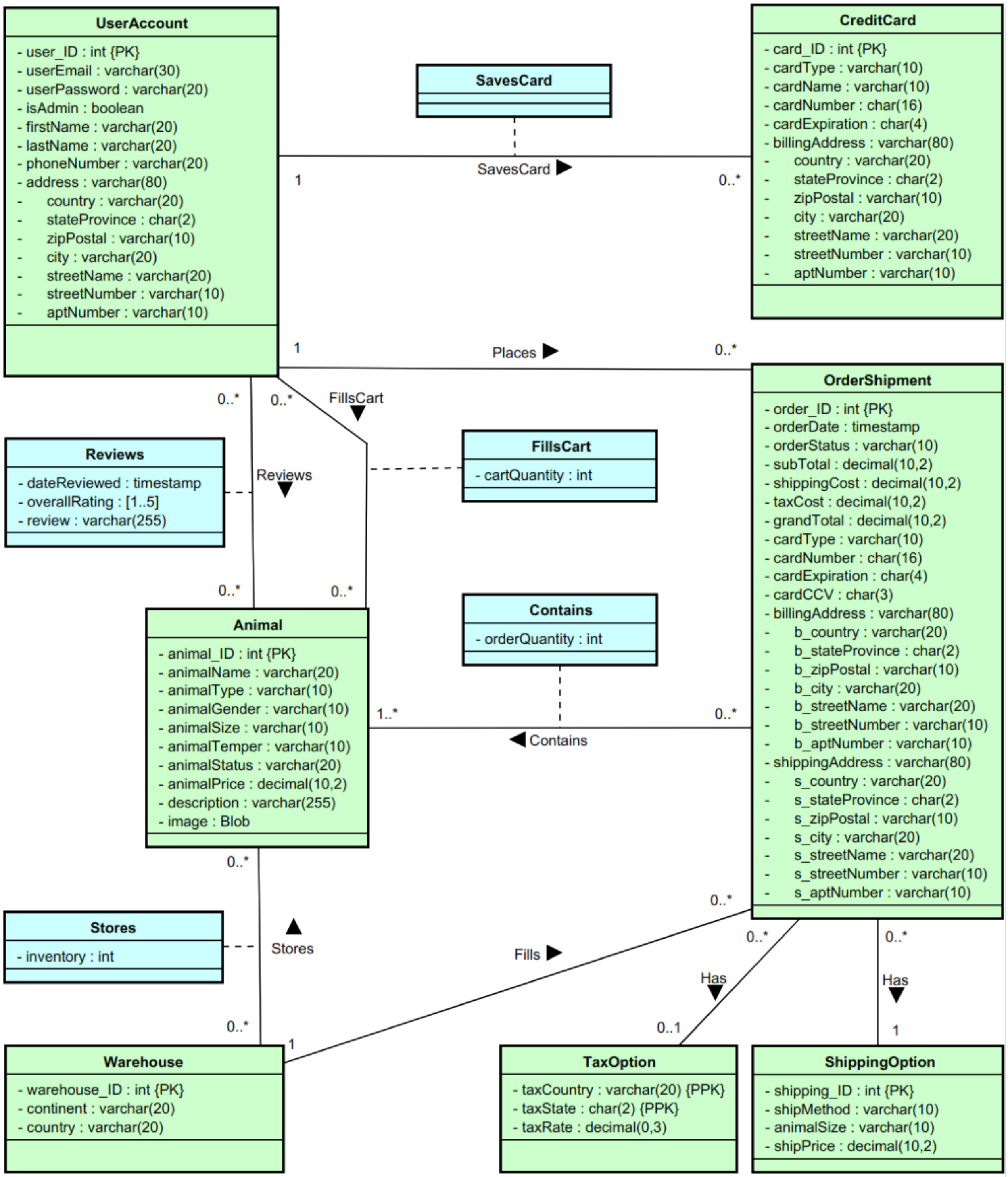
- All transactions will be processed using a credit card (VISA, Mastercard, AMEX).
- **VISA and Mastercard numbers must be 16 digits. Amex number must be 15 digits.**
- All transactions are advertised and processed using CAD currency.
- There will only be one tax rate per country and state / province combination.
- Returns or refunds will not be accepted.

2.6 Legal

- Verification of government permits for importing or exporting animals is not required.

3 Data Model

3.1 UML Diagram


UserAccount
- user_ID : int {PK}
- userEmail : varchar(30)
- userPassword : varchar(20)
- isAdmin : boolean
- firstName : varchar(20)
- lastName : varchar(20)
- phoneNumber : varchar(20)
- address : varchar(80)
- country : varchar(20)
- stateProvince : char(2)
- zipPostal : varchar(10)
- city : varchar(20)
- streetName : varchar(20)
- streetNumber : varchar(10)
- aptNumber : varchar(10)
SavesCard
SavesCard ▶
1
0..*
CreditCard
- card_ID : int {PK}
- cardType : varchar(10)
- cardName : varchar(10)
- cardNumber : char(16)
- cardExpiration : char(4)
- billingAddress : varchar(80)
- country : varchar(20)
- stateProvince : char(2)
- zipPostal : varchar(10)
- city : varchar(20)
- streetName : varchar(20)
- streetNumber : varchar(10)
- aptNumber : varchar(10)
1
Places ▶
0..*
0..*
0..*
FillsCart ▼
Reviews
- dateReviewed : timestamp
- overallRating : [1..5]
- review : varchar(255)
Reviews ▼
FillsCart
- cartQuantity : int
0..*
0..*
Animal
- animal_ID : int {PK}
- animalName : varchar(20)
- animalType : varchar(10)
- animalGender : varchar(10)
- animalSize : varchar(10)
- animalTemper : varchar(10)
- animalStatus : varchar(20)
- animalPrice : decimal(10,2)
- description : varchar(255)
- image : Blob
Contains
- orderQuantity : int
1..*
0..*
◀ Contains
OrderShipment
- order_ID : int {PK}
- orderDate : timestamp
- orderStatus : varchar(10)
- subTotal : decimal(10,2)
- shippingCost : decimal(10,2)
- taxCost : decimal(10,2)
- grandTotal : decimal(10,2)
- cardType : varchar(10)
- cardNumber : char(16)
- cardExpiration : char(4)
- cardCCV : char(3)
- billingAddress : varchar(80)
- b_country : varchar(20)
- b_stateProvince : char(2)
- b_zipPostal : varchar(10)
- b_city : varchar(20)
- b_streetName : varchar(20)
- b_streetNumber : varchar(10)
- b_aptNumber : varchar(10)
- shippingAddress : varchar(80)
- s_country : varchar(20)
- s_stateProvince : char(2)
- s_zipPostal : varchar(10)
- s_city : varchar(20)
- s_streetName : varchar(20)
- s_streetNumber : varchar(10)
- s_aptNumber : varchar(10)
0..*
Stores
- inventory : int
▲ Stores
Fills ▶
0..*
0..*
0..*
Has ▼
Has ▼
0..*
1
0..1
1
Warehouse
- warehouse_ID : int {PK}
- continent : varchar(20)
- country : varchar(20)
TaxOption
- taxCountry : varchar(20) {PPK}
- taxState : char(2) {PPK}
- taxRate : decimal(0,3)
ShippingOption
- shipping_ID : int {PK}
- shipMethod : varchar(10)
- animalSize : varchar(10)
- shipPrice : decimal(10,2)

3.2 Entity Descriptions

UserAccount	
Attribute	**Description**
user_ID (Primary Key)	Uniquely identifies a user. Auto-increments during account creation.
userEmail	Used for account access. Should match a@a.com format.
userPassword	Used for account access. Should be at least 5 characters long.
isAdmin	Boolean flag to identify administrator account type. Triggers administrator portal.
firstName	First name of the user. First letter should be stored as upper case.
lastName	Last name of the user. First letter should be stored as upper case.
phoneNumber	Phone number of the user. Stored as a string. Should match # (###) ### - #### format.
address	The address of the user. Composite key containing the sub-attributes below.
country	The country that the user lives in. Used as default for shipping.
stateProvince	The state or province that the user lives in. Used as default for shipping.
zipPostal	The zip or postal code for the user's address. Used as default for shipping.
city	The city, town, or village that the user lives in. Used as default for shipping.
streetName	The name of the street that the user lives on. Used as default for shipping.
streetNumber	The house or building number of the user's address. Used as default for shipping.
aptNumber	The suite, condo, or apartment number for the user's address. Optional.

Animal	
Attribute	**Description**
animal_ID (Primary Key)	Uniquely identifies an animal name (model). Auto-increments when adding to Animal.
animalName	The name of the animal (model). [Frog, Eagle, Piranha, Starfish, Mammoth, Chameleon]
animalType	Defines the type of animal. [Amphibian, Bird, Fish, Invertebrate, Mammal, Reptile]
animalGender	Defines the gender of the animal. [Female, Male]
animalSize	Defines the size of the animal. [Tiny, Small, Medium, Large, Giant]
animalTemper	Defines the temperament of the animal. [Domestic, Farm, Wild, Ferocious]
animalStatus	Defines the conservation status of the animal. [Common, Threatened, Endangered]
animalPrice	The pre-tax price of the animal for the customer.
description	A lengthy description (up to 255 characters) of the animal.
image	An image of the animal. Stored in database as binary large object.

CreditCard	
Attribute	**Description**
card_ID (Primary Key)	Uniquely identifies a credit card stored by a user.
cardType	Defines the type of credit card. [VISA, Mastercard, AMEX]
cardName	The nick name of the card as specified by the user.
cardNumber	The credit card account number. Entered as #### #### #### #### but trimmed after.
cardExpiration	The expiration year and month of the credit card. Stored in MMYY format.
billingAddress	The credit card's billing address. Composite key containing the sub-attributes below.
country	The country for the credit card's address. Used to help determine the tax rate.
stateProvince	The state or province for the credit card's address. Used to help determine the tax rate.
zipPostal	The zip or postal code for the credit card's address.
city	The city, town, or village for the credit card's address.
streetName	The name of the street for the credit card's address.
streetNumber	The house or building number for the credit card's address.
aptNumber	The suite, condo, or apartment number for the credit card's address. Optional.

Warehouse	
Attribute	**Description**
warehouse_ID (Primary Key)	Uniquely identifies a warehouse. Auto-increments when adding to Warehouse.
continent	The continent in which the warehouse is located. [Africa, Antarctica, Asia, Europe, North America, Central America, South America, Australia]
country	The country in which the warehouse is located.

OrderShipment

Attribute	Description
order_ID (Primary Key)	Uniquely identifies an order. Auto-incremented when adding to Order.
orderDate	The date and time that the order was placed. Determined by server at time of purchase.
orderStatus	The status of the order. Administrators can modify. [In Process, Shipped, Delivered]
subTotal	The total cost of the order before taxes and shipping costs.
shippingCost	The shipping cost determined by the shipping method selected by the customer.
taxCost	The cost of taxes for the order. Calculated by multiplying the tax rate with the sub-total.
grandTotal	The overall cost of the product (animal) sub-total, shipping cost, and taxes.
cardType	Defines the type of credit card. [VISA, Mastercard, AMEX]
cardNumber	The credit card account number. Entered as #### #### #### #### but trimmed after.
cardExpiration	The expiration year and month of the credit card. Stored in MMYY format.
cardCCV	The **C**ard **V**erification **V**alue number found on the back of the credit card in ### format.
billingAddress	The billing address for the order. **Stored at the time of purchase**.
b_country	The country for the charged credit card's address. Used to help determine the tax rate.
b_stateProvince	The state of province for the charged credit card's address.
b_zipPostal	The zip or postal code for the charged credit card's address.
b_City	The city, town, or village for the charged credit card's address.
b_streetName	The name of the street for the charged credit card's address.
b_streetNumber	The house or building number for the charged credit card's address.
b_aptNumber	The suite, condo, or apartment number for the charged credit card's address. Optional.
shippingAddress	The shipping address for the order. **Stored at the time of purchase**.
s_country	The country for the shipping address. Used to help determine the tax rate.
s_stateProvince	The state of province for the shipping address. Used to help determine the tax rate.
s_zipPostal	The zip or postal code for the shipping address.
s_city	The city, town, or village for the shipping address.
s_streetName	The name of the street for the shipping address.
s_streetNumber	The house or building number for the shipping address.
s_aptNumber	The suite, condo, or apartment number for the shipping address. Optional.

ShippingOption

Attribute	Description
shipping_ID (Primary Key)	Uniquely identifies a shipping option. Auto-increments when adding to Shipping Option.
shipMethod	Defines the shipping option associated with the shipping price. [Ground, Air]
animalSize	The animal size associated with the shipping price. [Tiny, Small, Medium, Large, Giant]
shipPrice	The price associated with the shipping method for the given animal size.

TaxOption

Attribute	Description
taxCountry (Partial Primary Key)	Matches with the country from the order's billing address to determine the tax rate.
taxState (Partial Primary Key)	Matches with the state or prov. from the order's billing address to determine tax rate.
taxRate	The tax rate specified by the country and state or prov. Applied to the order's sub-total.

FillsCart / OrderContains

Attribute	Description
cartQuantity	The quantity of a specific animal (model) added to the user's shopping cart.
orderQuantity	The quantity of a specific animal (model) added to the order.

Reviews

Attribute	Description
dateReviewed	The date and time that the user left a review on the animal (model) received.
overallRating	A rating between 1 and 5 (lowest to highest) of the animal (model) received. [1..5]
review	A descriptive review of the animal. One review per an animal, per a user - after purchase.

Stores

Attribute	Description
inventory	The quantity of a specified animal (model) at a specified warehouse.

3.3 Relational Assumptions

Relationship	Description
UserAccount **FillsCart** → Animal	A user can have zero to many animals in their shopping cart. An animal can be in zero to many users' shopping carts.
UserAccount **Reviews** → Animal	A user can review zero to many animals. An animal can have zero to many reviews.
UserAccount **SavesCard** → CreditCard	A user can save zero to many credit cards. A credit card can only belong to one user.
UserAccount **Places** → OrderShipment	A user can place zero to many orders / shipments. An order / shipment can only be placed by one user.
OrderShipment **Contains** → Animal	An order / shipment must contain at least one animal. An animal may be on zero to many orders / shipments.
OrderShipment **Has** → ShippingOption	An order / shipment must have one shipping method. A shipping option can be used by zero to many orders / shipments.
OrderShipment **Has** → TaxOption	An order / shipment can have zero or one tax option. A tax option can be used by zero to many orders / shipments.
Warehouse **Stores** → Animal	A warehouse can store zero to many animals. An animal can be stored in zero to many warehouses.
Warehouse **Fills** → OrderShipment	A warehouse can fill zero to many orders / shipments. An order / shipment can only be filled by one warehouse.

3.4 Relational Schema – DDL

```
CREATE TABLE Animal (
        animal_ID INTEGER AUTO_INCREMENT,
        animalName VARCHAR(20) NOT NULL,
        animalType VARCHAR(10) NOT NULL,
        animalGender VARCHAR(10) NOT NULL,
        animalSize VARCHAR(10) NOT NULL,
        animalTemper VARCHAR(10) NOT NULL,
        animalStatus VARCHAR(20) NOT NULL,
        animalPrice DECIMAL(10,2) NOT NULL,
        description VARCHAR(255) NOT NULL,
        image BLOB,
        PRIMARY KEY (animal_ID)
);

CREATE TABLE UserAccount (
        user_ID INTEGER AUTO_INCREMENT,
        userEmail VARCHAR(30) NOT NULL,
        userPassword VARCHAR(20) NOT NULL,
        isAdmin BOOLEAN,
        firstName VARCHAR(20) NOT NULL,
        lastName VARCHAR(20) NOT NULL,
        phoneNumber VARCHAR(20) NOT NULL,
        address VARCHAR(80) NOT NULL,
        country VARCHAR(20) NOT NULL,
        stateProvince CHAR(2) NOT NULL,
        zipPostal VARCHAR(10) NOT NULL,
        city VARCHAR(20) NOT NULL,
        streetName VARCHAR(20) NOT NULL,
        streetNumber VARCHAR(10) NOT NULL,
```

```
CREATE TABLE OrderContains (
        order_ID INTEGER,
        animal_ID INTEGER,
        orderQuantity INTEGER,
        PRIMARY KEY (order_ID, animal_ID)
);

CREATE TABLE SavesCard (
        user_ID INTEGER,
        card_ID INTEGER,
        PRIMARY KEY (user_ID, card_ID),
        FOREIGN KEY (user_ID)
                REFERENCES UserAccount(user_ID)
                ON DELETE NO ACTION ON UPDATE CASCADE,
        FOREIGN KEY (card_ID)
                REFERENCES CreditCard(card_ID)
                ON DELETE NO ACTION ON UPDATE CASCADE
);

CREATE TABLE OrderShipment (
        order_ID INTEGER AUTO_INCREMENT,
        orderDate TIMESTAMP NOT NULL,
        orderStatus VARCHAR(10) NOT NULL,
        subTotal DECIMAL(10,2),
        shippingCost DECIMAL(10,2),
        taxCost DECIMAL(10,2),
        grandTotal DECIMAL(10,2),
        cardType VARCHAR(10) NOT NULL,
        cardNumber CHAR(16) NOT NULL,
    aptNumber VARCHAR(10),
    PRIMARY KEY (user_ID),
    UNIQUE (userEmail)
);

CREATE TABLE Warehouse (
    warehouse_ID INTEGER AUTO_INCREMENT,
    continent VARCHAR(20),
    country VARCHAR(20),
    PRIMARY KEY (warehouse_ID)
);

CREATE TABLE CreditCard (
    card_ID INTEGER AUTO_INCREMENT,
    cardType VARCHAR(10) NOT NULL,
    cardName VARCHAR(10) NOT NULL,
    cardNumber CHAR(16) NOT NULL,
    cardExpiration CHAR(4) NOT NULL,
    billingAddress VARCHAR(80) NOT NULL,
    country VARCHAR(20) NOT NULL,
    stateProvince CHAR(2) NOT NULL,
    zipPostal VARCHAR(10) NOT NULL,
    city VARCHAR(20) NOT NULL,
    streetName VARCHAR(20) NOT NULL,
    streetNumber VARCHAR(10) NOT NULL,
    aptNumber VARCHAR(10),
    user_ID INTEGER,
    PRIMARY KEY (card_ID)
);

CREATE TABLE ShippingOption (
    shipping_ID INTEGER AUTO_INCREMENT,
    shipMethod VARCHAR(10),
    animalSize VARCHAR(10),
    shipPrice DECIMAL(10,2),
    PRIMARY KEY (shipping_ID)
);

CREATE TABLE TaxOption (
    taxCountry VARCHAR(20),
    taxState CHAR(2),
    taxRate DECIMAL(2,2),
    PRIMARY KEY (taxCountry, taxState)
);  CREATE TABLE Stores (
    warehouse_ID INTEGER,
    animal_ID INTEGER,
    inventory INTEGER,
    PRIMARY KEY (warehouse_ID, animal_ID)
);

CREATE TABLE FillsCart (
    user_ID INTEGER,
    animal_ID INTEGER,
    cartQuantity INTEGER,
    PRIMARY KEY (user_ID, animal_ID),
    FOREIGN KEY (user_ID)
        REFERENCES UserAccount(user_ID)
        ON DELETE NO ACTION ON UPDATE CASCADE,
    FOREIGN KEY (animal_ID)
        REFERENCES Animal(animal_ID)
        ON DELETE NO ACTION ON UPDATE CASCADE

);
```

```
    cardExpiration CHAR(4) NOT NULL,
    cardCCV CHAR(3) NOT NULL,
    billingAddress VARCHAR(80) NOT NULL,
    b_country VARCHAR(20) NOT NULL,
    b_stateProvince CHAR(2) NOT NULL,
    b_zipPostal VARCHAR(10) NOT NULL,
    b_city VARCHAR(20) NOT NULL,
    b_streetName VARCHAR(20) NOT NULL,
    b_streetNumber VARCHAR(10) NOT NULL,
    b_aptNumber VARCHAR(10),
    shippingAddress VARCHAR(80) NOT NULL,
    s_country VARCHAR(20) NOT NULL,
    s_stateProvince CHAR(2) NOT NULL,
    s_zipPostal VARCHAR(10) NOT NULL,
    s_city VARCHAR(20) NOT NULL,
    s_streetName VARCHAR(20) NOT NULL,
    s_streetNumber VARCHAR(10) NOT NULL,
    s_aptNumber VARCHAR(10),
    warehouse_ID INTEGER,
    shipping_ID INTEGER,
    user_ID INTEGER,
    animal_ID INTEGER,
    PRIMARY KEY (order_ID),
    FOREIGN KEY (user_ID)
        REFERENCES UserAccount(user_ID)
        ON DELETE NO ACTION ON UPDATE CASCADE,
    FOREIGN KEY (animal_ID)
        REFERENCES Animal(animal_ID)
        ON DELETE NO ACTION ON UPDATE CASCADE,
    FOREIGN KEY (warehouse_ID)
        REFERENCES Warehouse(warehouse_ID)
        ON DELETE NO ACTION ON UPDATE CASCADE,
    FOREIGN KEY (shipping_ID)
        REFERENCES ShippingOption(shipping_ID)
        ON DELETE NO ACTION ON UPDATE CASCADE,
    FOREIGN KEY (b_country, b_stateProvince)
        REFERENCES TaxOption(taxCountry, taxState)
        ON DELETE NO ACTION ON UPDATE CASCADE
);

CREATE TABLE Reviews (
    user_ID INTEGER,
    animal_ID INTEGER,
    dateReviewed TIMESTAMP,
    overallRating INTEGER,
    review VARCHAR(255),
    CHECK (overallRating >= 1 AND overallRating <= 5),
    PRIMARY KEY (user_ID, animal_ID),
    FOREIGN KEY (user_ID)
        REFERENCES UserAccount(user_ID)
        ON DELETE NO ACTION ON UPDATE CASCADE,
    FOREIGN KEY (animal_ID)
        REFERENCES Animal(animal_ID)
        ON DELETE NO ACTION ON UPDATE CASCADE

);
```

4 Web Interface

4.1 Site Description

ZooBC's homepage will have a header that will serve as the primary navigation tool. The user will be able to select any of the relevant buttons to go to the sites corresponding to the buttons This navigation bar will also provide a search bar that allows users to search for any animals that they would like to purchase. The navigation bar also includes the login option and will display currently logged in user. This header will be present in every page to simplify and improve user's experience on navigating the site.

The body of the page will include a featured animal, which can be clicked to go directly to their specific page and some information on our company.

The search bar will list all relevant result after user inputs a query. The user can then select any of the result to be lead directly into that animal's specific page, which will have a short description and their pricing.

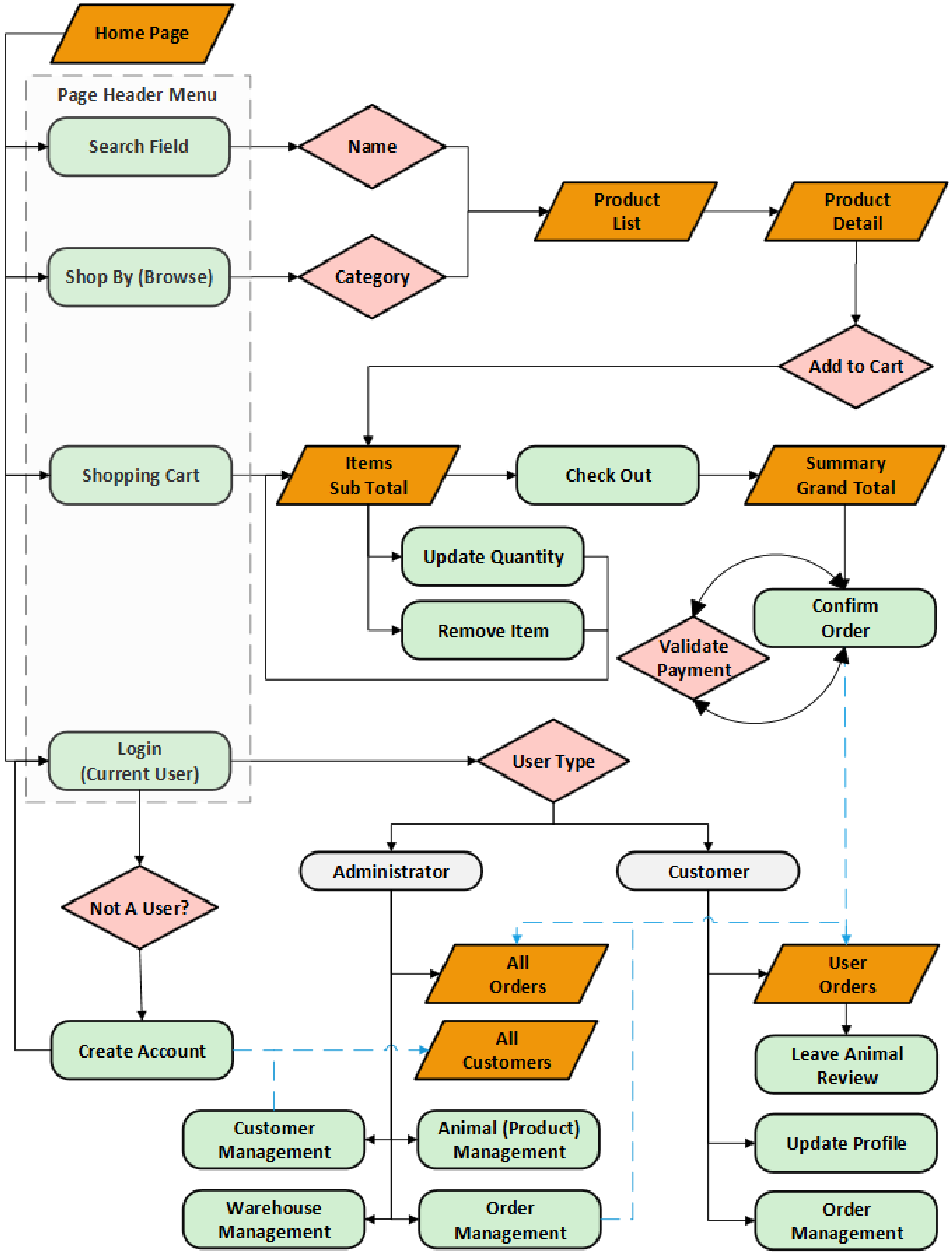
The user can also search based on categories, where the user can choose whether to choose mammals, amphibians, fish, reptile, invertebrate, or birds. Choosing any of these options will lead to a similar page as the ones after using the search bar and will list animal that falls under the chosen category.

The user login page will allow user to input their username and password if they have already registered. If not, the sign-up link is also available, where they can fill in the relevant information in the provided space.

Each user has their own account page, where they can browse their orders, update profile information, make an order, or review an animal that they have bought.
Administrators also have additional features, namely browsing through all orders, looking at the customer data, warehouse & order management.

Lastly, users have access to their shopping carts. They will be filled with animals that users have added to it. It will then ask user for payment and shipping information. Afterwards, the site will display the total cost of the order and a confirm order button.

4.2 Site Map


Home Page
Page Header Menu
Search Field
Name
Shop By (Browse)
Category
Product List
Product Detail
Add to Cart
Shopping Cart
Items Sub Total
Check Out
Summary Grand Total
Update Quantity
Remove Item
Confirm Order
Validate Payment
Login (Current User)
User Type
Administrator
Customer
Not A User?
All Orders
User Orders
Create Account
All Customers
Leave Animal Review
Customer Management
Animal (Product) Management
Update Profile
Warehouse Management
Order Management
Order Management

4.3 Planned Features

Final Overall System Features	Points	% Complete	Code Files
Hosting			
Running on cosc304 server	2		
Main/Home Page			
Search for a product by name	1		
Browse products by category	1		
List products (by search/browse)	1		
List products with image	2		
Page header with menu	1		
Page header shows current logged in user	1		
Shopping Cart			
Add to shopping cart	1		
View shopping cart	1		
Update item quantity in shopping cart	2		
Remove item from shopping cart	1		
Improved formatting/UI (e.g. in header bar)	1		
Cart stored in database between sessions	3		
Checkout			
Checkout with customer id	1		
Checkout with payment/shipment info	2		
Checkout with data validation	3		
Inventory tracking per item	2		
Inventory tracking by store/warehouse	3		
Calculate taxes and shipping cost	2		
Product Detail Page			
Product detail page and item description	3		
Product detail has an image	1		
Product detail has an image from database	4		
User Accounts and Login			
Create user account page	3		
Create user account with data validation	3		
Login/logout	2		
Page listing all orders for user	1		

Final Overall System Features	**Points**	**% Complete**	**Code Files**
Product Reviews			
Ability to enter a review on a product	2		
Display product review on product detail	1		
Restrict to one review per user on item purchased	2		
Administrator Portal			
Secured by login	3		
List all customers	1		
List report showing total sales/orders	1		
Report with a graph	3		
Add new product	2		
Update/delete product	2		
Change order status/ship order	1		
Upload a photo to database for product	4		
Database restore with SQL script	2		
Add/update warehouse, customer	2		
Database System/General			
Implement some validation using triggers	2		
Use AJAX for some pages	4		
Bonus – User interface and navigation/usability			
Bonus – User can save multiple credit cards			