



QSCU Merchandise Store

Document Status: Final Report
November 30th 2018

COSC 304 Team Members:

Removed

Table Of Contents

1. Introduction	2
1.1 Mission Statement	2
1.2 Purpose of Document	2
1.3 Executive Summary	2
2. System Features	2
3. Walkthrough	5
3.1 Administrator Walkthrough	5
3.2 User Walkthrough	7
3. UML Diagram	11
3.1 UML Description	11
3.1.1 User	12
3.1.2 Product	12
3.1.3 Warehouse	12
3.1.4 Order	12
3.1.5 Shipment	12
4. Relational Schema	13
4.1 SQL DDL	13
5. Website Interface	15
5.1 Site Map	15
5.2 Explanation of outline	15
6. Going Forward	16
6.1 Known Issues and Future Work	16

1. Introduction

1.1 Mission Statement

Our goal is to implement a system that will allow for a convenient and simple way to purchase Quantitative Science Course Union (QSCU) merchandise, however this system could also be implemented for any course union or club at UBC Okanagan.

1.2 Purpose of Document

This document highlights the design, features, and a walkthrough for QSCU's merchandise store. There will be both an administrator walkthrough, and a user walkthrough included. The website layout, database scheme, and database DDL will be outlined in this document.

1.3 Executive Summary

Since 2011, the QSCU has represented UBC Okanagan students majoring or minoring in Mathematics, Computer Science, Statistics, Physics and Data Science. The QSCU organizes academic events such as workshops and review sessions, as well as social event such as faculty mixers and badminton events. Through this website, the course union will have the ability to advertise and sell their branded merchandise while also creating a platform to showcase previous years' t-shirts. The users are required to create an account in order to purchase any of the merchandise and check out their cart. However, everyone will have the ability to browse the various items.

2. System Features

Final Overall System Features	Possible Points	% Complete	Code Files/Links
Hosting			
Running on COSC360 server	2	100%	http://cosc360.ok.ubc.ca/46418166/the-project-qscu-merch/
OR: Running on external hosted server	5	100%	http://23.95.191.102/qscu/src/
Main/Home Page			
Search for a product by name	1	100%	searchpage.php
Browse products by category	1	100%	searchpage.php
List products (by search/browse)	1	100%	searchpage.php , homeWithoutTables.php
List products with image	2	100%	homeWithoutTables.php,

			searchpage.php
Page header with menu	1	100%	header.php
Page header shows current logged in user	1	100%	header.php
Dynamic product on page based on sales	2	100%	homeWithoutTables.php
Shopping Cart			
Add to shopping cart	1	100%	addToCart.php
View shopping cart	1	100%	viewCart.php
Update item quantity in shopping cart	2	100%	updateCart.php
Remove item from shopping cart	1	100%	updateCart.php
Improved formatting/UI (e.g. in header bar)	1	100%	header.php
Cart stored in database between sessions	3	100%	addToCart.php
Checkout			
Checkout with customer id	1	100%	checkout-action.php
Checkout with payment/shipment info	2	100%	checkout-action.php
Checkout with data validation	3	100%	checkout.php
Inventory tracking per item	2	100%	checkout-action.php
Inventory tracking by store/warehouse	3	100%	Checkout-action.php, displayInventory.php
Multiple shipments per order supported	3	100%	checkout-action.php
Calculate taxes and shipping cost	2	100%	orderSummary.php
Product Detail Page			
Product detail page and item description	3	100%	singleProduct.php
Product detail has an image	1	100%	singleProduct.php
Product detail has an image from database	4	100%	singleProduct.php
User Accounts and Login			
Create user account page	3	100%	signup.php
Create user account with data validation	3	100%	signup.php, client-side-validation.js
Login/logout	2	100%	login.php, logout.php
Register user/Forgot password email	4	100%	forgotPass.php, forgot-password-validation.js
Product Reviews			
Ability to enter a review on a product	2	100%	reviewModal.js, setReview.php

Display product review on product detail	1	100%	singleProduct.php, reviewModal.js
Restrict to one review per user on item purchased	2	100%	setReview.php, DB requirement
Administrator Portal			
Secured by login	3	100%	login.php, getLogin.php
List all customers	1	100%	admin_controller.php, adminList.php
Add new product	2	100%	addProduct.php, adminList.php
Update/delete product	2	100%	editProduct.php
Upload a photo to database for product	4	100%	addProduct.php
Add/update warehouse, customer	2	50%	adminList.php, getUserList.php
Report with a graph	3	100%	displayInventory.php
Database System/General			
Use AJAX for some pages	4	100%	searchpage.php, adminList.php, singleProduct.php, addToCart.php, addToCart.js, search_controller.js
User interface and navigation/usability	5	100%	All non-action files. Eg: profile.php
Contact Us form that sends message via email (see walk through)	Optional - 2	100%	contactUs.php
Custom 404 Error Page	Optional - 1	100%	http://cosc360.ok.ubc.ca/46418166/the-project-qscu-merch/src/error404.php

In addition to the above features, we also implement potent form validation, so the user cannot break our website. For example, if a user enters a string for their credit card number, we are able to catch that it is invalid. There is also a plethora of client-side and server-side validation. User's can also have separate billing and shipping addresses, and their billing information auto populates the form upon checkout if they have already submitted their billing information. The user is then able to edit this information, or keep it as is.

Instead of deleting a product entirely, an administrators enables it allowing it to be sold. This allows administrators to go back and look at old products that were sold.

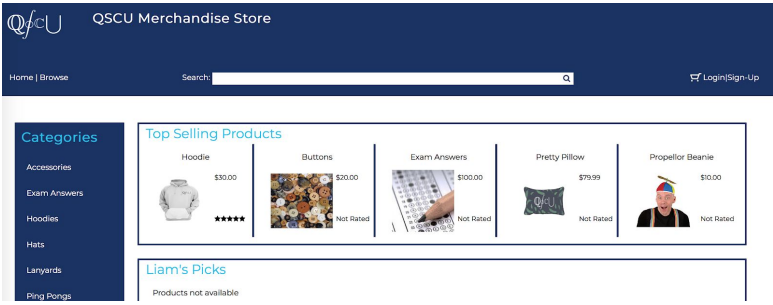
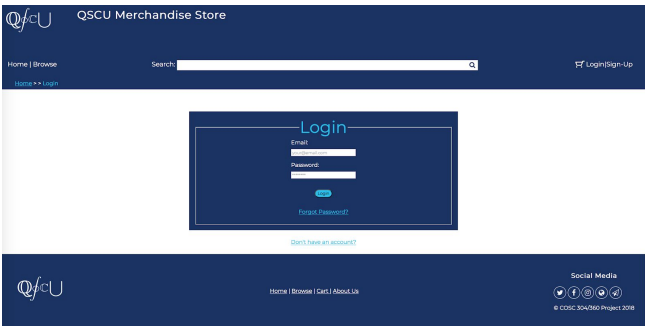

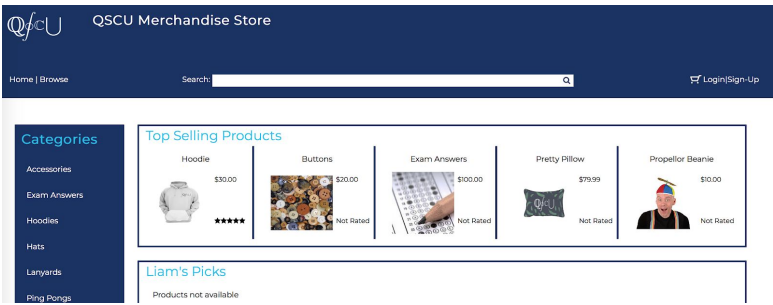
User's can be banned by administrators, and if a user is banned and they attempt to checkout, they will automatically be logged out and redirected to a ban page.

3. Walkthrough

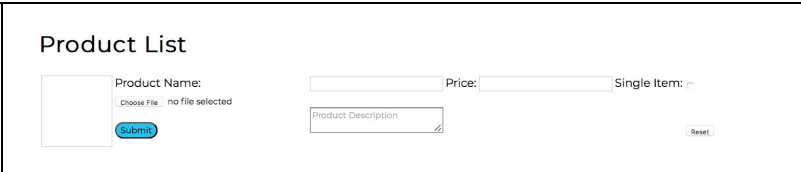
For this walkthrough, please use:

<http://cosc360.ok.ubc.ca/46418166/the-project-qscu-merch/>

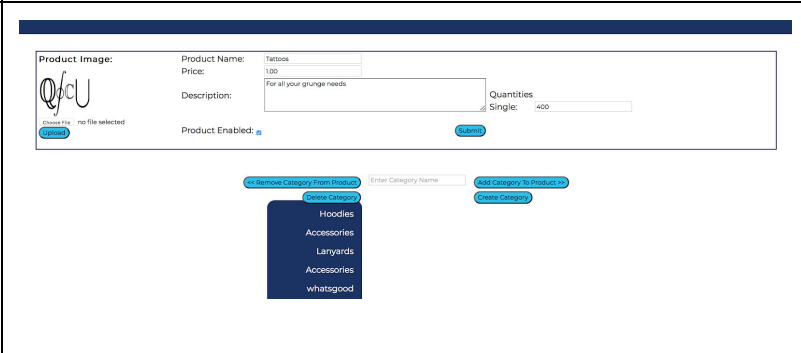
3.1 Administrator Walkthrough

<p>1</p>	<p>Welcome to our homepage! Please click the <i>login</i> link on the right side of the header, which will direct you to our login page</p>	
<p>2</p>	<p>An administrator account has been created, and can be accessed with the email <i>rl@ubc.ca</i> and the password <i>1234</i>. Please login to this account.</p>	
<p>3</p>	<p>You will now notice in the header and <i>admin</i> link, please click this link</p>	
<p>4</p>	<p>This will direct you to the administrator's portal, where the home page displays all products.</p>	

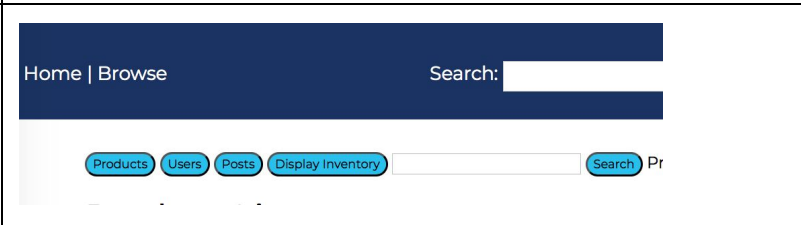
5 To add a product, click add product and fill in the details. Note that for an image, the file size limit is 10 MB.



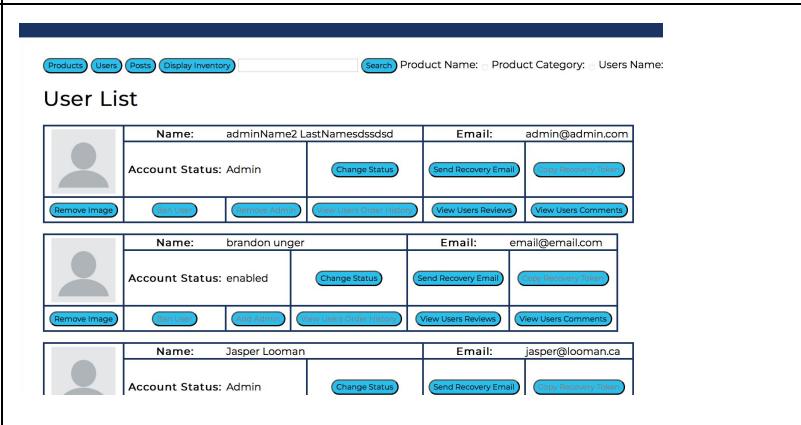
6 This will direct you to an edit product page, where you can further fill in details, including creating a new category for your product or adding it to an existing category.. Once you are satisfied, feel free to click the *Admin* link again, on the right hand side of the header. Note that your new product is now displayed at the bottom of the list



7 To display a list of user's, click the *Users* button



8 This will direct you to a page of Users with accounts. You are able to edit these users by giving them administrative privileges or banning them. If you wish, feel free to change the status of a user by clicking *Change Status*, or remove their profile image. Just note that you must refresh the page to see that their image has been removed. You may also view any reviews or comments, if the specific user has made any



9 To display warehouse inventory, click the *Display Inventory* link. This will direct you to our warehouse inventory page that displays a warehouse and its current inventory. It also displays a chart of how each product price and quantity contributes to our total value of assets

Warehouse Product Inventory List [Return To Admin](#)

Warehouse Number	Location
1	Kelowna

Product Id	Product Name	Size	Quantity
7	Hoodie	large	5
7	Hoodie	medium	5
7	Hoodie	small	10
7	Hoodie	xl	5
8	Propellor Beanie	single	100
9	QSCU Lanyard	single	30
10	Buttons	single	3
11	Exam Answers	single	20
12	Socks	single	0
13	Sweatpants	large	2

<p>10</p>	<p>You may also click Posts in the main page of the admin portal. This will display a any reviews and comments that a product has received.</p>	
-----------	---	--

Thus concludes the administrative walkthrough. To logout, simply click the *Logout* link on the right side of the header. Now for a typical user.

3.2 User Walkthrough

<p>1</p>	<p>A user is greeted by our homepage, which displays typical functions such as a search bar, and various featured products</p>	
<p>2</p>	<p>We encourage the user sign up for an account, so please click the <i>Signup</i> link located on the right hand side of the header. This will direct you to the sign up page, so please enter the required information.</p>	
<p>2. a</p>	<p>If a user already has an account the they may login to their existing account using the <i>Login</i> link, also located in on the right hand side of the header bar.</p>	

<p>2. b</p>	<p>If a user forgets their password, we have implemented a password recovery system via the user's email. They will be sent a token and a link that will allow them to reset their password.</p>	
<p>3</p>	<p>If you created a new account, you will be directed to the profile page, where you can fill in and edit your information, but let's leave this for now. Note that the right side of the header now has your account name and a <i>logout</i> button. Feel free to fill in this information, but note that the file size of you profile picture must be maximum 1 MB</p>	
<p>4</p>	<p>Going back to the homepage, why don't you select one of the top products. This will direct you to the product's page, that includes more details and reviews about the product. You can submit a review if you like by clicking the review button. You are also able to read other reviews of the product left by other users, as well as comments on the reviews</p>	
<p>5</p>	<p>To write a review, simply click Write a Review. Rate and item out of 5 stars, write a comment, and submit. You must refresh the page for the review to appear. You can also comment on a review by clicking the comment button beside the review. For the comment to appear, the page must refresh</p>	

6 To add the product to your cart, simply select the amount and size that you would like, and click add to cart. To view your cart, click the shopping cart icon on the right side of the header bar, beside your name. From here we can view the items in your cart, and you can edit it as well by changing an item's quantity, or removing it entirely

Your Shopping Cart

Product Name	Quantity	Size	Price	
Hoodie	2	medium	\$60	Update Item Remove
Your Subtotal: \$60			Check-out	

7 Once you are ready to checkout, click the *checkout* link, which will direct you to our checkout page, where you may either pay via credit card or PayPal. When you are ready, be sure to submit your order

Check Out

Name on Card:

Credit Card Number:

Expiry Date (MMYY):

CCV:

Select a shipping address:

- Billing address
- New shipping address

[Check-Out](#)

Billing Address:

Address Line:

City:

Country:

Province/State:

Postal Code:



8 Clicking checkout will send you to a confirmation page, that includes taxes in your total price. You may also edit your order if you wish to do so, which will take you back to your cart.

Home | Browse Search: [Johnny](#) | [Logout](#)

[Home](#) >> [Cart](#) >> [Checkout](#) >> [Your Order](#)

Your Order:

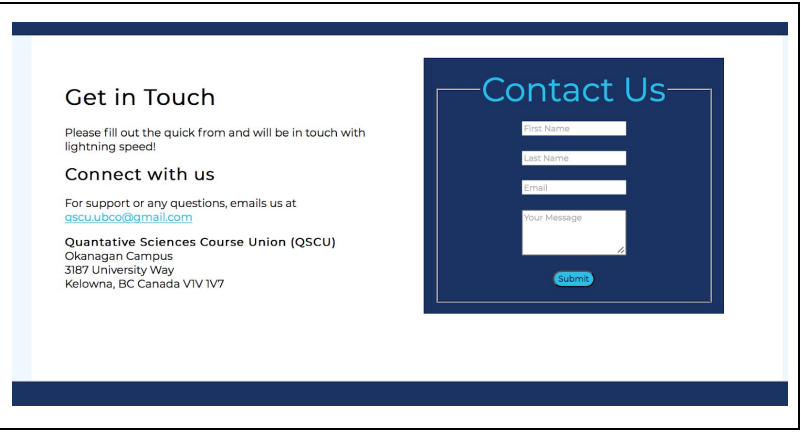
Quantity	Product	Size	Price
3	Hoodie	medium	90

Subtotal: \$90
Taxes: \$10.80
Total Cost: \$100.80

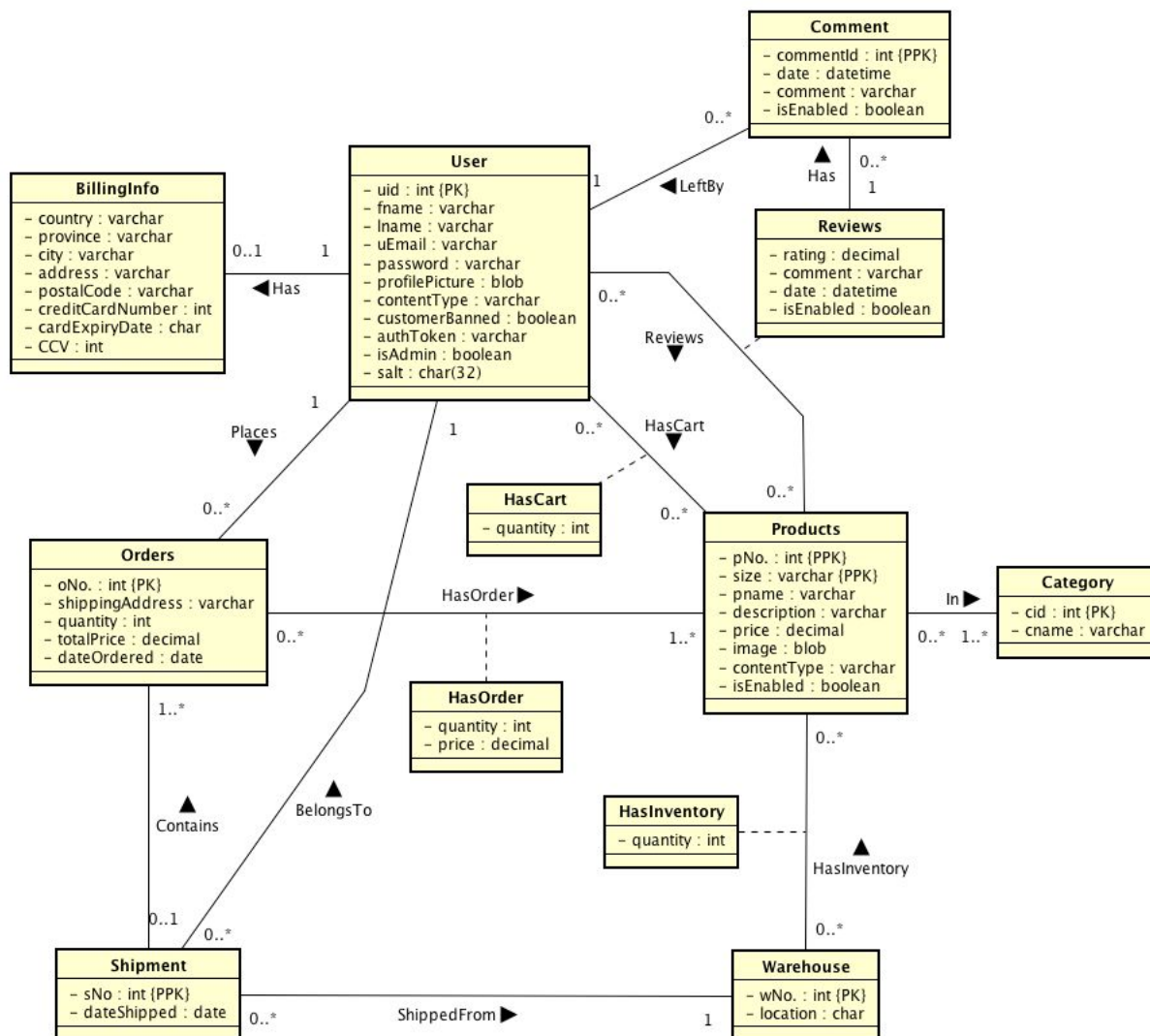
[Edit](#) [Confirm](#)

<p>9</p>	<p>Upon clicking <i>confirm</i>, you will be directed to a confirmation page which will display a message stating your order has been received and on its way. If not enough of the ordered product is in stock, message will be displayed apologizing for the inconvenience.</p>	
<p>10</p>	<p>Returning home, we can try the search feature. If you search for for a string that exists in a product name, ie: <i>Hoo</i>, any product matching <i>Hoo</i> will appear. In this case our hoodies will populate the results. If you type a string that does not appear in any product name, then no results will appear</p>	
<p>11</p>	<p>Clicking the Browse link on the left side of the header will open up a category window, where you can browse products by their category.</p>	

12 In the footer, there is a *Contact Us* link. Clicking this link will direct you to a page that will send an email to the QSCU directly.



3. UML Diagram



3.1 UML Description

Displayed above is the UML diagram that represents our database structures.

3.1.1 User

A user is identified by a unique ID number (uid). Included is data about the user such as their name and email address (required), as well as if they have been banned or are an admin. A user is able to upload a profile picture as well.

1. A user may choose to input their billing info, however upon checkout, they have the choice of paying with PayPal.
2. Each user has a cart, where they store products that they are interested in. Items in their cart can be edited by increasing or decreasing the quantity, or removed entirely. A user may checkout with their cart

3.1.2 Product

Each product is uniquely identified by its product number and size. If a product has no size, or is one size fits all, the *size* attribute is set to "single". The attribute *isEnabled* is set to true if the product is currently being sold, false otherwise.

1. Each Product has reviews, created by a logged in user. These reviews include a rating and description
 - 1.1. Each review can have multiple comments, made by users
2. Each product belongs to a category. A product must be in at least one category, but may belong to multiple. The size of a product is not included in this relation
 - 2.1. Each category is identified by its ID, and has a name

3.1.3 Warehouse

A warehouse is identified by its ID, and each warehouse has a location.

1. Each warehouse has an inventory of products

3.1.4 Order

An order is placed by the user, and contains products. An order contains a shipping address, which is not necessarily the same as the user's billing address. An order has a total order price. Each order is placed by a single user, and must contain at least one product

1. The products in the order are specified by their primary key (*pNo*, *size*). The price of the product in an order is determined by the quantity ordered and the product's individual price

3.1.5 Shipment

A shipment is shipped from a single warehouse, and may contain multiple orders under the assumption they have been ordered by the same user, and are being delivered to the same address.

4. Relational Schema

4.1 SQL DDL

A relational schema as constructed from the UML diagram written using SQL DDL.

```

DROP TABLE IF EXISTS BillingInfo;
DROP TABLE IF EXISTS Comment;
DROP TABLE IF EXISTS Reviews;
DROP TABLE IF EXISTS HasCart;
DROP TABLE IF EXISTS ProductInCategory;
DROP TABLE IF EXISTS HasOrder;
DROP TABLE IF EXISTS HasInventory;
DROP TABLE IF EXISTS Orders;
DROP TABLE IF EXISTS Shipment;
DROP TABLE IF EXISTS Category;
DROP TABLE IF EXISTS Warehouse;
DROP TABLE IF EXISTS Product;
DROP TABLE IF EXISTS User;

CREATE TABLE Product (
    pNo          INTEGER NOT NULL
AUTO_INCREMENT,
    pname       VARCHAR(50),
    size        VARCHAR(32) NOT NULL
DEFAULT 'single',
    price       DECIMAL(9,2),
    image       BLOB,
    contentType VARCHAR(20),
    description VARCHAR(500),
    isEnabled   BOOLEAN,

    PRIMARY KEY (pNo, size)
);

CREATE TABLE User (
    uid          INTEGER NOT NULL
AUTO_INCREMENT,
    fname       VARCHAR(40),
    lname       VARCHAR(40),
    uEmail      VARCHAR(50) NOT
NULL,
    password    VARCHAR(200) COLLATE
latin1_general_cs NOT NULL,
    profilePicture BLOB,
    contentType VARCHAR(20),

    customerBanned   INTEGER,
    authToken        VARCHAR(32),
    isAdmin           TINYINT(1),
    salt              CHAR(32) NOT NULL,

    PRIMARY KEY (uid)
);

CREATE TABLE Category (
    cid          INTEGER NOT NULL
AUTO_INCREMENT,
    cname        VARCHAR(50),

    PRIMARY KEY (cid)
);

CREATE TABLE ProductInCategory (
    pNo          INTEGER,
    cid          INTEGER,

    PRIMARY KEY (pNo, cid),
    CONSTRAINT
FK_ProductInCategory_Product FOREIGN
KEY (pNo) REFERENCES Product(pNo)
ON DELETE NO ACTION ON UPDATE
CASCADE,
    CONSTRAINT
FK_ProductInCategory_Category FOREIGN
KEY (cid) REFERENCES Category(cid)
ON DELETE NO ACTION ON UPDATE
CASCADE
);

CREATE TABLE Warehouse (
    wNo          INTEGER NOT NULL
AUTO_INCREMENT,
    location     VARCHAR(50),

    PRIMARY KEY (wNo)
);

```

```

);

CREATE TABLE HasInventory (
    wNo          INTEGER NOT NULL,
    pNo          INTEGER NOT NULL,
    size         VARCHAR(32) NOT NULL,
    quantity     INTEGER,

    PRIMARY KEY (pNo, size, wNo),
    CONSTRAINT FK_HasInventory_Product
FOREIGN KEY (pNo, size) REFERENCES
Product(pNo, size)
        ON DELETE NO ACTION ON UPDATE
CASCADE,
    CONSTRAINT
FK_HasInventory_Warehouse FOREIGN KEY
(wNo) REFERENCES Warehouse(wNo)
        ON DELETE NO ACTION ON UPDATE
CASCADE
);

CREATE TABLE Shipment (
    sNo          INTEGER NOT NULL
AUTO_INCREMENT,
    dateShipped  DATE,
    uid          INTEGER NOT NULL,
    shippedFrom  INTEGER NOT NULL,

    PRIMARY KEY (sNo),
    CONSTRAINT FK_Shipment_User FOREIGN
KEY (uid) REFERENCES User(uid)
        ON DELETE NO ACTION ON UPDATE
CASCADE,
    CONSTRAINT FK_Shipment_Warehouse
FOREIGN KEY (shippedFrom) REFERENCES
Warehouse(wNo)
        ON DELETE NO ACTION ON UPDATE
CASCADE
);

CREATE TABLE HasCart (
    uid          INTEGER NOT NULL,
    pNo          INTEGER,
    size         VARCHAR(32),
    quantity     INTEGER,

    PRIMARY KEY (uid, pNo, size),
    CONSTRAINT FK_HasCart_User FOREIGN
KEY (uid) REFERENCES User(uid)
        ON DELETE NO ACTION ON UPDATE
CASCADE,
    CONSTRAINT FK_HasCart_Product
FOREIGN KEY (pNo, size) REFERENCES
Product(pNo, size)
        ON DELETE NO ACTION ON UPDATE
CASCADE
);

CREATE TABLE Orders (
    oNo          INTEGER NOT NULL
AUTO_INCREMENT,
    shippingAddress VARCHAR(50),
    totalPrice    DECIMAL(10,2),
    dateOrdered   DATE,
    uid          INTEGER NOT NULL,
    sNo          INTEGER NOT NULL,

    PRIMARY KEY (oNo),
    CONSTRAINT FK_Orders_User FOREIGN
KEY (uid) REFERENCES User(uid)
        ON DELETE NO ACTION ON UPDATE
CASCADE,
    CONSTRAINT FK_Orders_Shipment
FOREIGN KEY (sNo) REFERENCES
Shipment(sNo)
        ON DELETE NO ACTION ON UPDATE
CASCADE
);

CREATE TABLE HasOrder (
    oNo          INTEGER,
    pNo          INTEGER,
    size         VARCHAR(32),
    quantity     INTEGER,
    price        DECIMAL(10,2),

    PRIMARY KEY (oNo, pNo, size),
    CONSTRAINT FK_HasOrder_Orders
FOREIGN KEY (oNo) REFERENCES
Orders(oNo)
        ON DELETE NO ACTION ON UPDATE
CASCADE,
    CONSTRAINT FK_HasOrder_Product
FOREIGN KEY (pNo, size) REFERENCES
Product(pNo, size)
        ON DELETE NO ACTION ON UPDATE
CASCADE
);

```

```
CREATE TABLE Reviews (
    uid          INTEGER NOT NULL,
    pNo          INTEGER,
    rating       INTEGER,
    comment      VARCHAR(500),
    date         DATETIME,
    isEnabled   BOOLEAN,

    PRIMARY KEY (uid, pNo),
    CONSTRAINT FK_Reviews_User FOREIGN
KEY (uid) REFERENCES User(uid)
    ON DELETE NO ACTION ON UPDATE
CASCADE,
    CONSTRAINT FK_Reviews_Product
FOREIGN KEY (pNo) REFERENCES
Product(pNo)
    ON DELETE NO ACTION ON UPDATE
CASCADE
);
```

```
CREATE TABLE Comment (
    commentId   INTEGER NOT NULL
AUTO_INCREMENT,
    uid         INTEGER NOT NULL,
    pNo         INTEGER,
    leftBy      INTEGER NOT NULL,
    date        DATETIME,
    comment     VARCHAR(250),
    isEnabled   BOOLEAN,

    PRIMARY KEY (commentId, uid, pNo),
```

```
    CONSTRAINT FK_Comment_Reviews
FOREIGN KEY (uid, pNo) REFERENCES
Reviews(uid, pNo) ON DELETE NO ACTION
ON UPDATE CASCADE,
    CONSTRAINT FK_Comment_User FOREIGN
KEY (leftBy) REFERENCES User(uid)
    ON DELETE NO ACTION ON UPDATE
CASCADE
);
```

```
CREATE TABLE BillingInfo (
    uid          INTEGER NOT NULL,
    country      VARCHAR(40),
    province     VARCHAR(5),
    city         VARCHAR(50),
    address      VARCHAR(50),
    postalCode   VARCHAR(15),
    creditCardNumber VARCHAR(19),
    cardExpiryDate CHAR(4),
    CCV          INTEGER,

    PRIMARY KEY (uid),
    CONSTRAINT FK_BillingInfo_User
FOREIGN KEY (uid) REFERENCES User(uid)
    ON DELETE NO ACTION ON UPDATE
CASCADE
);
```

5. Website Interface

5.1 Site Map

Provided is the link to full site map: <https://www.gloomaps.com/hwAbsYmEQe>. We also provide a selection of images and descriptions in the event that the link does not work.

5.2 Explanation of outline

Image 1 is the shopping cart branch. A user's shopping cart is accessible directly from the home page of the website via a shopping cart icon. Additionally, the shopping cart can be accessed from the



product page as shown with the green labels in the above images. Technically, the shopping cart may also be accessed from any page on the site with a link in the header, but the sitemap would simply get messy if every link were to be shown explicitly. From the shopping cart page, a user will be prompted to login if they haven't already. If they choose to login, it will take them to the login branch. If they are logged in already they may proceed to checkout.

Image 2 is the Login branch. This branch is directly accessible from the home page, and all other pages via the header. Additionally, a user will be sent to the login branch if they try to view their cart without being logged in. At the login page, a user will either login and go to their account, create an account, or be linked to the forgotten password page.

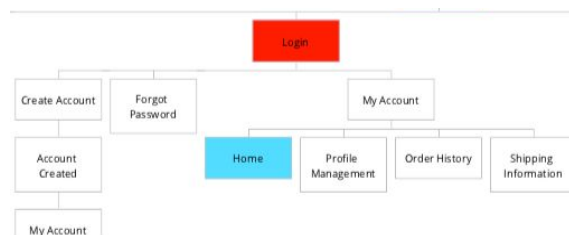


Image 3 is all the other main branches that are accessible from the home page. Specifically, the logout page, which will then return the user to the home page, the basic product page, which can lead a user to the shopping cart branch and the contact us page which will give a user the option to send an email to the QSCU, and when they press send, will lead them to a page detailing whether the message was sent properly or if it failed.



The colours in the sitemap are simply in place to help with showing where certain pages will lead to. Each of the highest level pages in the three images are direct children of the home page.

6. Going Forward

This website will eventually be operating as part of QSCU's main website - qscu.org. Moving forward, the website will undergo minor tweaks to better suit the QSCU's needs, and actual QSCU merchandise will be entered into the database. This site will play a major role in the course union's merchandise sales and will be accessible to all students, faculty and staff of UBC Okanagan.

6.1 Known Issues and Future Work

Going forward, the QSCU insert their products into the website via the administrator portal. The QSCU must also purchase an online database system if they wish to implement

this system. In addition, we wish to implement an invoice system for user orders, where an electronic receipt will be sent to a user via email once they place an order.

A known issue has to do with the user's cart. Currently, if a user is not logged in, then adds items to their cart, then logs into their account, their cart information will be lost. Going forward this must be fixed.