UBCVIP

In case public education wasn't classist enough.

Names Removed

CONTENTS

1 INTRODUCTION

Don't you hate lugging your bags around school? Aren't you sick of wasting time looking for parking spots? Haven't you ever wished there was a homework fairy? And what is the point of all your money if you can't spend it on a better university experience?

UBCVIP can help you with that. From extra time with a professor to a literal mule to carry your bags, *UBCVIP*'s four-tiered system offers plans to suit every price range, even poor people.

1.1 PURPOSE OF DOCUMENT

This document outlines the design and underlying structure of *UBCVIP*'s website, including our assumptions, descriptions, and site map.

1.2 MISSION STATEMENT

To bring our customers the highest quality experience during their university years. To provide the most lavish and class-appropriate facilities to our customers. To ensure the perpetuation of the bourgeoisie economic supremecy in society.

1.3 EXECUTIVE SUMMARY

UBCVIP is a web store designed specifically for University of British Columbia students to purchase and gain access to different premiums. These premiums are fashioned to make the student life more relaxed and manageable. Customers have the ability to make an account with *UBCVIP* and sign up for one of four membership subscriptions–Platinum, Gold, Silver, Rust–depending on the budget of the student and the premiums that they wish to have access to. It is important to note that as the membership level increases (from Rust to Silver to Gold to Platinum), the number of products that are included in the package for free also increases. In other words the higher the membership level, the more premiums are available to the customer. No matter what your membership level, you may purchase products for a price at any time. If the product is a subscription period item, the customer must repay for the item at the end of the current subscription period, or lose access.

The website provides simple navigation and allows the user to search and filter the *UBCVIP* product database. An unregistered user may view all the products offered by *UBCVIP* and add to their shopping cart, but all users must make an account to proceed to checkout. By default, any unregistered user is considered to possess a Rust membership for the purposes of how the products are displayed. All the products listed on the website contain at least one picture of the product, a description of the product, and the price of the product for the corresponding membership.

To register a new account, the user must provide a valid student number, a unique email address, a "safe" password, and their first and last names. The user has the ability to change this information at anytime. A newly registered customer is not required to pay for a membership immediately because, upon registering, they are set to have the free Rust membership with auto-renew turned on. If the user decides to upgrade their membership, they may do so at anytime. All memberships are based on a monthly subscription. If the user has auto-renew disabled, their membership level will return to Rust at the end of the subscription period. If the user decides to upgrade or downgrade their membership in the middle of a subscription period, their current subscription will continue until the end of the subscription period, at which time their new subscription will begin.

2 DOMAIN ASSUMPTIONS

2.1 CUSTOMER

- Customers are identified by a unique, auto-incremented integer that has no meaning to the customers themselves.

- Customer student ID numbers must be unique and not null, thus a Customer must be a UBC student.

- Customer email addresses must be unique and not null.

- Non-customers may browse the website, but you must sign up for a MembershipLevel to make a purchase.

- Membership levels are monthly subscriptions (see 2.2).

- A customer can opt to auto-renew their subscription.

- A customer can opt to save their credit card information.

- There are only four types of membership levels: Platinum, Gold, Silver, and Rust.

- The membership level Rust (MembershipLevel.mid = 1) is the free membership package; Rust auto-renews by default.

- If a customer has not selected to auto-renew and the end date of their subscription passes (see 2.2), then they are downgraded to Rust until they manually renew.

2.2 ORDERS

- Orders for memberships and orders for products are recorded separately.

- All order IDs are unique and auto-incremented.

- MembershipOrder tracks the start date and end date for a membership subcription.

- MembershipOrder tracks whether the order is an initial sign-up or a renewal.

- A customer may not sign up for a membership and purchase products at the same time.

- Customers can only pay via credit card.

2.3 PRODUCTS

- Products are identified by unique, auto-incremented integers.

- A customer of any membership level may buy any product; however, membership levels will have various products at a significant discount, or for free.

- There are seven different types of products: exam access, bathrooms, parking, lounge, bonus accessories, bonus time, and life lines.

- Products are type-identified by their product code (Product.pcode).
    - Exam Access: EX
    - Bathrooms: BR
    - Parking: PK

- Lounge: LO

- Bonus Accessories: BA

- Bonus Time: BT

- Life Lines: LL

- ReservedParking extends Product.

- When a reserved spot is purchased, its reserved attribute changes to true.

- If all ReservedParking has been reserved, then an error message appears to let a customer know that the team is in the process of acquiring more spots.

- ReservedParking is the only item sold on the website which has limited availability.

- Single exams are not products; a customer buys access to a type of exam.

3 ENTITY DESCRIPTION

3.1 UML DIAGRAM
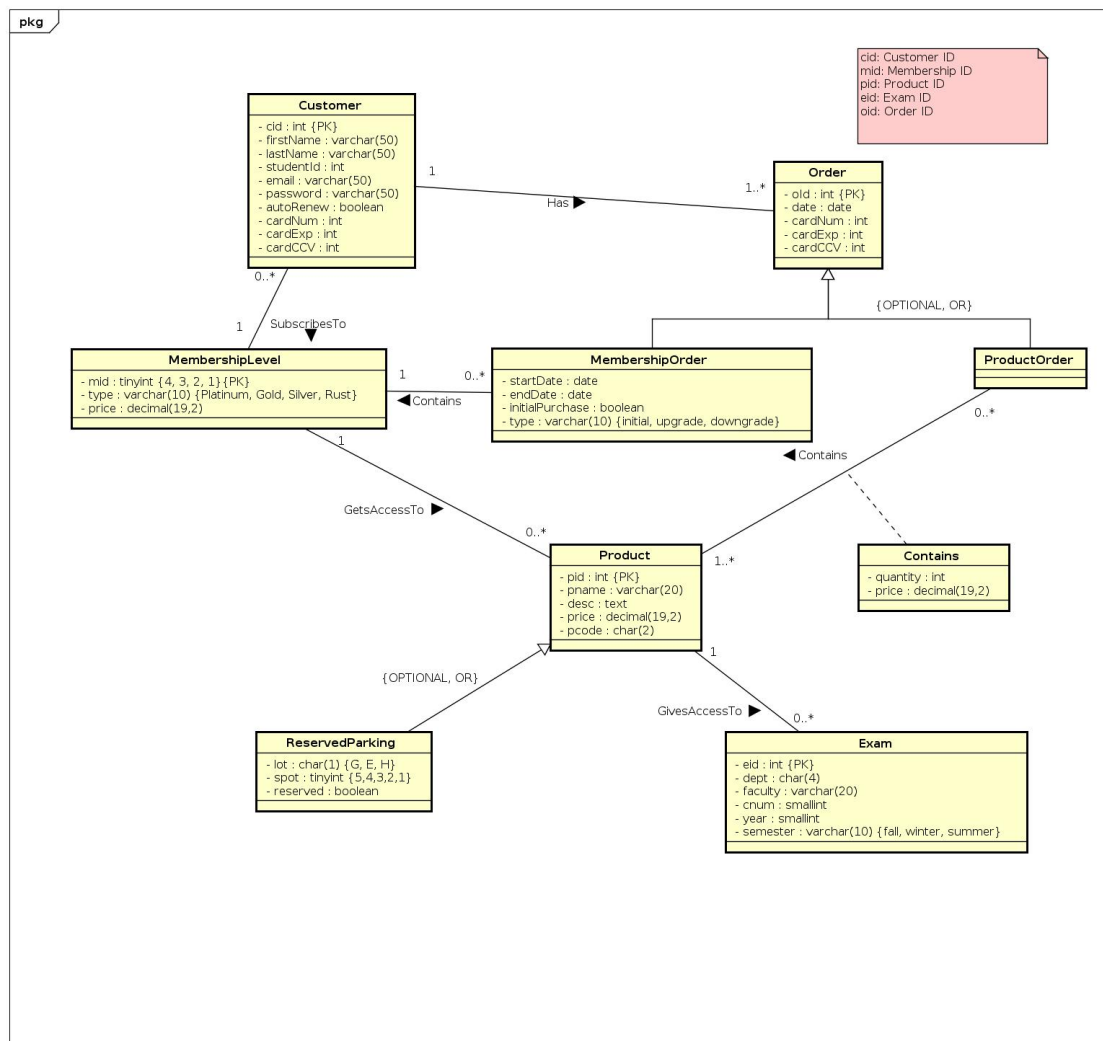
3.2 UML Diagram: Brief Description

3.2.1 Customer

A customer is a person who has signed up for one of the membership plans we offer through our website. They must be a UBC student, and enter their first name, last name, UBC student ID, email, and password to successfully create an account with us. If they chose any of the non-free plans, they may choose to have their plan auto-renewed each month. They may also choose to register their credit card information with us.

3.2.2 MembershipLevel

Our store offers four different membership levels for customers to subscribe to. Our most expensive plan is Platinum, which offers a full access package to all of our products, and the plans decrease in price and access to Gold, Silver, and then Rust. Rust is the free plan we offer, and customers must pay individually for all products.

3.2.3 Order

Order is a superclass for MembershipOrder and ProductOrder. It contains the unique order ID, the date the order was made, and the credit card information

3.2.4 MembershipOrder

Before a customer is able to access any products, or purchase any individual products, they must complete a membership order. Memberships are monthly subscriptions, and so the start and end date of each order is recorded. Customers may upgrade or downgrade their memberships at the beginning of each billing cycle.

3.2.5 ProductOrder

This table stores all of the different products a customer has purchased at one time through our website.

3.2.6 Contains

This table stores the quantity and total price for each individual product in an order.

3.2.7 Product

This table stores all the information about the products we offer to our customers. Users visiting our website may view all of our product data and descriptions before deciding to sign up for a membership plan.

3.2.8 ReservedParking

The reserved parking that we offer to our customers requires more attributes to accurately describe it, so it is a sub-class of product. The lot location, the spot location, and whether that spot has already been reserved, is necessary data we must track.

3.2.9 Exam

This table stores all the information of all the exams we offer to our customers. There are only three exam products which, once purchased, give our customers access to all the exams for that particular access package. The exams are sorted by their respective faculty, department, semester, and course number.

3.3 Entity Attribute Description

3.3.1 Customer

Attribute	Description
cId: INT PRIMARY KEY	Unique Customer identification number.
firstName: VARCHAR(50)	The customer's first name.
lastName: VARCHAR(50)	The customer's last name.
studentId: INT	The customer's student identification number for UBC.
email: VARCHAR(50)	The customer's login email address.
password: VARCHAR(50)	The customer's login password.
autoRenew: BOOLEAN	Indicates if the customer wants to be automatically rebilled for their membership each month.
cardNum: INT	The customer's credit/debit card number.
cardExp: INT	The customer's credit/debit card expiration date.
cardCCV: INT	The customer's credit/debit card CCV number.

3.3.2 MembershipLevel

Attribute	Description
mId: TINYINT	Number which identifies membership level. 4=Platinum, 3=Gold, 2=Silver, 1=Rust
type: VARCHAR(10)	Identifies membership level.
price: DECIMAL(19,2)	The monthly price of chosen membership level.

3.3.3 Order

Attribute	Description
oId: INT PRIMARY KEY	A number which uniquely identifies a product order.
date: DATE	A date indicating the date an order was placed.
cardNum: INT	The credit/debit card number associated with a order.
cardExp: INT	The credit/debit card expiration date associated with the credit/debit card for the order.
cardCCV: INT	The credit/debit card CCV associated with an order.

3.3.4 MembershipOrder

Attribute	Description
startDate: DATE	Date when membership was purchased
endDate: DATE	Date when membership expires
initialPurchase: BOOLEAN	Indicates if this is the customer first membership purchase
type:VARCHAR(10)	Indicates whether this is the customer's initial purchase, an upgrade, or a downgrade in membership.

3.3.5 ProductOrder

This entity does not contain any attributes unique to it.

3.3.6 CONTAINS

Attribute	Description
quantity: INT	A number indicating the amount of an individual product in a single order.
price: DECIMAL(19,2)	A number indicating the total price of the amount of an individual product in an order.

3.3.7 PRODUCT

Attribute	Description
pId: INT PRIMARY KEY	Number that uniquely identifies a product.
pName: VARCHAR(20)	Product name.
desc: TEXT	A detailed description of the product.
price: DECIMAL(19,2)	The price of the product.
pCode: CHAR(2)	A description of the product category.

3.3.8 RESERVEDPARKING

Attribute	Description
lot: CHAR(1)	A single letter Indicating which parking lot the spot is in.
spot: TINYINT	A number indicating which spot of the chosen lot the parking spot is.
reserved: BOOLEAN	Indicates whether the chosen spot has already been reserved.

3.3.9 EXAM

Attribute	Description
eId: INT PRIMARY KEY	A number which uniquely identifies an exam.
dept: CHAR(4)	Four letters that indicate which department and exam belongs too.
faculty: VARCHAR(20)	Indicates which faculty an exam belongs to.
cnum: SMALLINT	A number that indicates what course number, of the department an exam is from.
year: SMALLINT	A number that indicates what year an exam is from.
semester: VARCHAR(10)	Indicates which semester the exam is from. (Fall, Winter, Spring).

3.4 RELATIONAL ASSUMPTIONS

Relationship	Description
Customer SubscribesTo MembershipLevel	This is a one-to-many relationship that associates a customer with their membership level. A customer must have a membership level, but a membership level might not have any customers.
Customer Has Order	This is a one-to-many relationship that associates a customer with an order for a product(s). As there are two types of orders (membership orders and product orders), a customer necessarily must have at least one order (for membership level).
MembershipOrder Contains MembershipLevel	This is a one-to-many relationship that associates an order for a membership level to the level that was bought. A membership order may only be for one membership level at a time, but membership levels may have never been purchased or may have been purchased multiple times.
MembershipLevel GetsAccessTo Product	This is a one-to-many relationship that associates various membership levels to the products that they are allowed to access for free. A membership level might have access to either no products (Rust) or many products, but a product is only associated with one membership level. This is because if it is associated with, for example, Silver (MembershipLevel.mid = 2), then every membership level with MembershipLevel.mid >=2 will have access to it.
Order Contains ProductOrder	This is a many-to-many relationship that associates an order with the various products that are being purchased on the order. A customer may buy any product at any time, despite their membership level. If a customer has access to products via their membership level, accessing these products would not create an order. There are two attributes stored in the relationship: quantity, an integer greater than zero, and a price, which is differentiated from the price attribute in product so that customers can be offered discounts, etc. A product may never have been ordered, but an order must contain at least one product.
Product GivesAccessTo Exam	This is an one-to-many relationship that associates a product with the exams that said product gives access to. A product may not give access to any exams (e.g. reserved parking). There are three products that give access: final exams, midterm exams, and previous exams.

3.5 RELATIONAL SCHEMA: SQL DDL

```sql
CREATE TABLE Customer (
        cId INT                   AUTO_INCREMENT,
        firstName VARCHAR(50)     NOT NULL,
        lastname VARCHAR(50)      NOT NULL,
          studentId INT                   NOT NULL,
        email VARCHAR(50)               NOT NULL,
        password VARCHAR(50)      NOT NULL,
        autoRenew BOOLEAN         NOT NULL,
        cardNum INT,
        cardExp INT,
        cardCCV INT,
        mId INT   CHECK(mid in (4,3,2,1)) NOT NULL,
        PRIMARY KEY(cid),
        FOREIGN KEY(mid) REFERENCES MembershipLevel(mid)
                ON DELETE NO ACTION
                ON UPDATE NO ACTION
);

CREATE TABLE MembershipLevel (
        mId TINYINT               CHECK (mid in (4,3,2,1)),
        type VARCHAR(10) CHECK (type in ('Platinum', 'Gold', 'Silver', 'Rust')) NOT NULL,
        price DECIMAL(19,2),
          PRIMARY KEY (mId)
);

CREATE TABLE MembershipOrder (
        oId INT,
        startDate DATE      NOT NULL,
        endDate DATE        NOT NULL,
        initialPurchase BOOLEAN          NOT NULL,
        type VARCHAR(10) CHECK (type in ('Initial', 'Upgrade', 'Downgrade') NOT NULL,
        mId INT             NOT NULL,
        PRIMARY KEY (oid),
        FOREIGN KEY (mId) REFERENCES MembershipLevel(mId)
                ON DELETE NO ACTION
                ON UPDATE NO ACTION
);

CREATE TABLE Order (
        oId INT                   AUTO_INCREMENT,
        date DATE           NOT NULL,
        cardNum INT,
        cardExp INT,
        cardCCV INT,
        cId INT    NOT NULL,
        PRIMARY KEY (oid),
        FOREIGN KEY (cId) REFERENCES Customer (cId)
                ON DELETE NO ACTION
                ON UPDATE NO ACTION
);

CREATE TABLE Product (
        pId INT                   AUTO_INCREMENT,
        pname VARCHAR(20)       NOT NULL,
        desc TEXT                     NOT NULL,
        price DECIMAL(19,2)             NOT NULL,
        pCode CHAR(2)     CHECK (pCode in ('EX', 'BR', 'PK', 'LO', 'BA', 'BT'))       NOT NULL,
        mId INT   CHECK (mId in (4,3,2,1)) NOT NULL,
        PRIMARY KEY (pId),
        FOREIGN KEY (mId) REFERENCES MembershipLevel (mId)
                ON DELETE NO ACTION
                ON UPDATE NO ACTION
);

CREATE TABLE ProductOrder (
        oId INT,
        PRIMARY KEY (oId),
        FOREIGN KEY (oId) REFERENCES Order(oid)
                ON DELETE NO ACTION
                ON UPDATE NO ACTION
);

CREATE TABLE Exam (
        eId INT,
        dept CHAR(4)       NOT NULL,
        faculty VARCHAR(20)        NOT NULL,
        cNum SMALLINT              NOT NULL,
        year SMALLINT              NOT NULL,
        Semester VARCHAR(10)       CHECK (Semester in('Spring', 'Winter', 'Summer')) NOT NULL,
        pId INT   NOT NULL,
        PRIMARY KEY (eId),
        FOREIGN KEY (pId) REFERENCES Product(pId)
                ON DELETE NO ACTION
                ON UPDATE NO ACTION
);

CREATE TABLE Contains (
        pId INT    ,
        oId INT    ,
        quantity INT CHECK(quantity >0)       NOT NULL,
        price DECIMAL(19,2)                   NOT NULL,
        PRIMARY KEY (pId, oId),
        FOREIGN KEY (pId) REFERENCES Product(pId)
                ON DELETE NO ACTION
                ON UPDATE NO ACTION
        FOREIGN KEY (oId) REFERENCES ProductOrder(oId)
                ON DELETE NO ACTION
                ON UPDATE NO ACTION
);

CREATE TABLE ReservedParking (

        lot CHAR(1)        CHECK(lot in('G','E','H')) NOT NULL,
        spot TINYINT       CHECK(spot in (5,4,3,2,1)) NOT NULL,
        reserved BOOLEAN        NOT NULL,
        PRIMARY KEY (rpId),
        FOREIGN KEY (pId) REFERENCES Product(pId)
                ON DELETE NO ACTION
                ON CASCADE NO ACTION
);
```
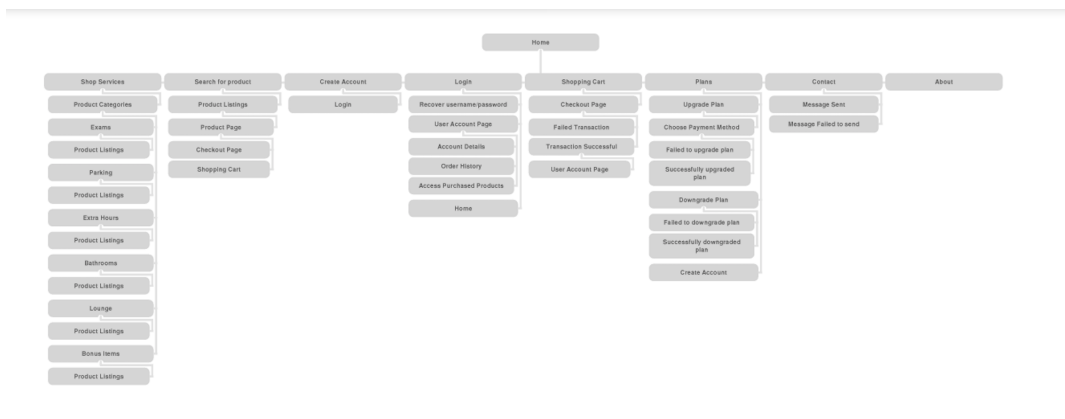
4 SITE MAP

4.1 List of Features

- Browse for products by keyword via the search bar in the fixed header.

- Filter products by category and subcategory in the product list page.

- View all the products in list form with their picture and description.

- View the previous orders with the quantity and price of the products order.

- Add or remove products from session shopping cart.

- View and update products in session cart.

- Checkout their session cart and receive order invoice.

- Create an account and purchase a membership.

4.2 Interface Description

This is a brief description of the intended user interaction, the interface, and the functionality of the *UBCVIP* webstore as it stands at this time. Concepts and designs are, however, subject to change before the final launch.

4.2.1 Header

Every page on *UBCVIP* is equipped with a fixed header that provides accessibility to shop services, membership plans, the *UBCVIP* contact page, and the about us page; these are listed in a horizontal list to the left of the website logo. It also allows the user to search the store, change the currency, direct themselves to the login page, and view their session cart.

4.2.2 Main

The main page, or the index page, of the website gives the user direct access to the customer sign in feature, where they are able to either, make an account or log in to an existing account. If the user decides to proceed in the first time user sign up, they can select the sign up button on the left and input their information into the customer required fields on the right. If the user has a pre-existing account, they can select the login button on the left and enter their email and password into the designated fields on the right, at which point they will be directed to the product category page. The login credential fields are shown by default upon loading the page.

4.2.3 Product Category Page

The product category page is layed out with a 2x3 matrix of product category panels. If a user selects a category they are redirected to the product list page that will be pre-filtered according to the category they have selected.

4.2.4 Product List Page

The product list page shows a list of products with each element in the list containing a picture of the product, the product name, the product description, and the price for one unit. It also supports user filtering with a vertical panel located on the left of the page. If the user selects a product from the list they are redirected to the specific product page.

4.3 SELECTED SITE IMAGES
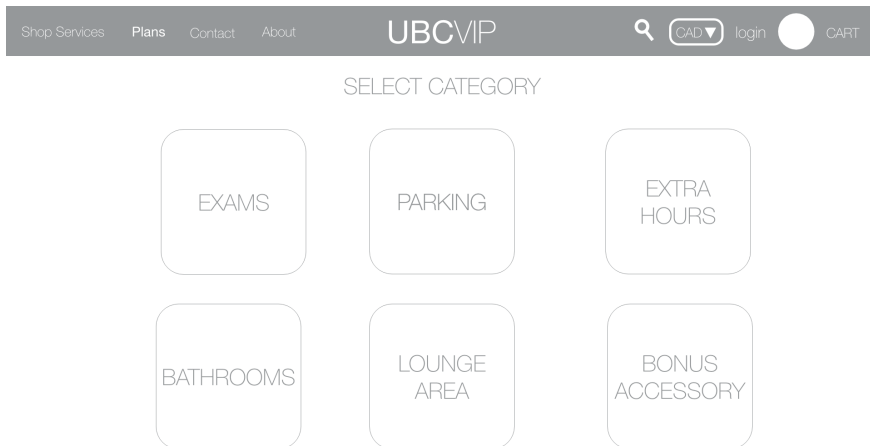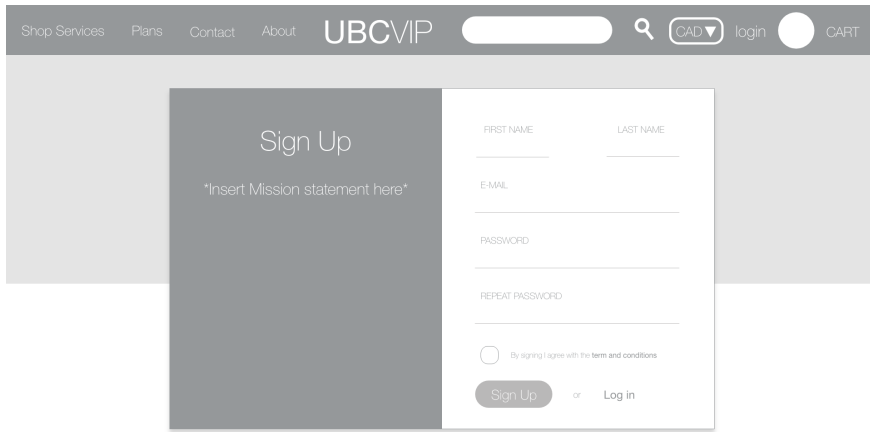
5 GOING FORWARD

We are confident in our ability to fulfill this design outline. However, the following describes our limitations and projected problems.

5.1 LIMITATIONS

- *UBCVIP* does not actually have access to the UBC database, therefore we cannot validate student IDs.

- *UBCVIP* unfortunately does not have access to UBC final exams.

- *UBCVIP* does not actually receive payments nor will any orders be fulfilled.

- We do not support administrative accounts on the website.

5.2 PROJECTED PROBLEMS

- Showing user recommended products.

- Displaying the most popular products on products page.

- Emailing invoices to users after a purchase is made.

- Creating a secure login page.