

Project Design Document for SealandEntitlements Online Store

COSC 304

Scott Fazackerley, Valentin Koch and Michael Henderson

October 27, 2006

Contents

1	Introduction	2
2	Mission Statement	2
3	Executive Summary	2
4	Domain Assumptions	3
5	Entity Descriptions	3
5.1	Relational Assumptions	3
5.2	Entity Attributes	6
5.2.1	Shipment	6
5.2.2	ShippingMethod	6
5.2.3	Product	6
5.2.4	Review	7
5.2.5	Warehouse	7
5.2.6	Store	7
5.2.7	Manager	7
5.2.8	Customer	8
5.2.9	SavedCard	8
5.2.10	PaymentMethod	8
5.2.11	Orders	8
5.2.12	Category	8
6	Relational Schema	9
7	User Interface and Planned Features	10
7.1	User Interface	10
7.2	Planned Features	11

1 Introduction

SealandEntitlements is combination online and physical store that sells and ships unofficial items that are related to principality of Sealand and is in no way affiliated with the principality of Sealand[1]. The company from here in will refer to SealandEntitlements NARLLC[2]. This document primarily focuses on the design requirement for the online store system for the company with some integration with the inventory system of the physical stores. Using this proposed system, a customer can purchase various items, such as degrees packages, certificates, seals and other related paraphernalia. All transactions and interaction with the company is performed online. The online store front end will be developed with JSP, with the back end running on MySQL and will be collectively referred to as the system.

2 Mission Statement

To build a web-based store that allows users to buy products online and manages inventory and shipping information from the stores. Users should be able to track the status of their orders, search and browse for items to buy, and feel confident and secure in the purchasing process. The system must also support internal use by administrators who can track the volume and type of goods sold, handle customer problems (such as forgotten passwords), manage product inventory levels, and generate sales reports.

3 Executive Summary

All products will be centrally stocked at one main warehouse. The main warehouse is located in Burnaby, British Columbia. The main warehouse will receive all stock directly from its suppliers. The main warehouse will the distribute products to individual stores and process online orders. When a customer places an order with the online store, it will not be processed until the next day. All orders have a cutoff time of 23:59:59(PDT). That is, if an order is received before this time on a given day, it will be processed on the next business day. Business days are Monday through Friday. Orders Received on a Friday or on a weekend or holiday, will be processed on the next business day.

A guest is an un-registered customer. Both a guest and a customer can view products through the online store, where products are grouped by categories and associated with at least one picture. Guests and customers can also view reviews on products from other customers who have purchased a specific product. For a guest to maintain a shopping cart or make a purchase, they are required to have an active account will the online store. That is, they must have registered and created an account. If a customer has forgotten their access information, they can request that the information is sent to them to the email address on record.

Customers can view their previous orders can review products, but a customer can only review a product that they have purchased. A customer can also maintain a shopping cart that will contain items that they have selected. To place an order, a customer must select items to be placed in their shopping cart and the order will be processed from their shopping cart. Once an item has been placed into an order, it will be removed from the shopping cart. Customers can view their order history at any time, and have the ability to cancel orders order items that have not shipped yet. Customers can also shop at a brick and mortar store, but their purchases will not be tracked from these locations.

If an item is out of stock or temporarily unavailable at the time their specific shipment is ready, they can request a partial shipment for the items that are available. Separate shipping charges will apply for spilt shipments. Alternatively, a customer can request to hold the shipment until all items are available. As stock is received, it will be directed to back orders before being made available to customer and other stores.

The company has both store managers and corporate managers. Store managers only have the ability to track stock and request inventory and generate salve reports. A corporate manager has the same permissions

as a manager in addition to: A corporate manager has the ability to recover the system by using a (SQL) script to create tables and load initial data. Additionally, a corporate manager can remove, or update item currently for sale. All administrative and manager functions are secured by a user id and password.

4 Domain Assumptions

The following domain assumptions are made in the design of this system:

- A customer will be uniquely identified by their email address and maintain a password.
- A manager will be uniquely identified by their id and maintain a password.
- All operations and shipments are based on pacific daylight/standard time as appropriate.
- All shipments will be with Fedex. Customers have three shipping choices
 - **Express** (Priority overnight or the best available option for the customer shipping region. Not available for all shipping regions.)
 - **Ground** (3 to 7 days or longer. Not available for all shipping regions.)
 - **International** (only available to customers living outside North America.)
- Payment methods accepted by the company are Visa and MasterCard. Other payment methods may come available in the future with increased customer base.
- A shipment can be of for different status options:
 - **not shipped** - the order has not shipped yet or is being processed.
 - **hold for completion** - the order is being held pending receipt of backordered items.
 - **partial shipment** - part of the order has shipped without backordered items. As backorders are received, item will ship. If the order has multiple backordered items that are received on different days, each item will ship separately.
 - **complete** - the entire order has shipped.

5 Entity Descriptions

The SealandEntitlements Entity Relationship model is presented in figure 1.

5.1 Relational Assumptions

The following relational assumptions are made about the system:

The **Shipment** relation contains all necessary information needed to ship one, more, or all products of an order. This relationship is shown in the ShippedProduct relation.

The **ShippedProduct** relation represents the relationship between entities Shipment and Product. The relationship shows the quantity of products that are contained in a shipment. A tuple in this relation belongs to a specific Shipment and a specific Product.

The **Product** relation describes an actual product that a customer want to buy. The information inside the relation is used for the customer to look up the product, and for an order to calculate the total price. A shipment contains one or multiple units of a Product. This relationship is shown in the ShippedProduct relation.

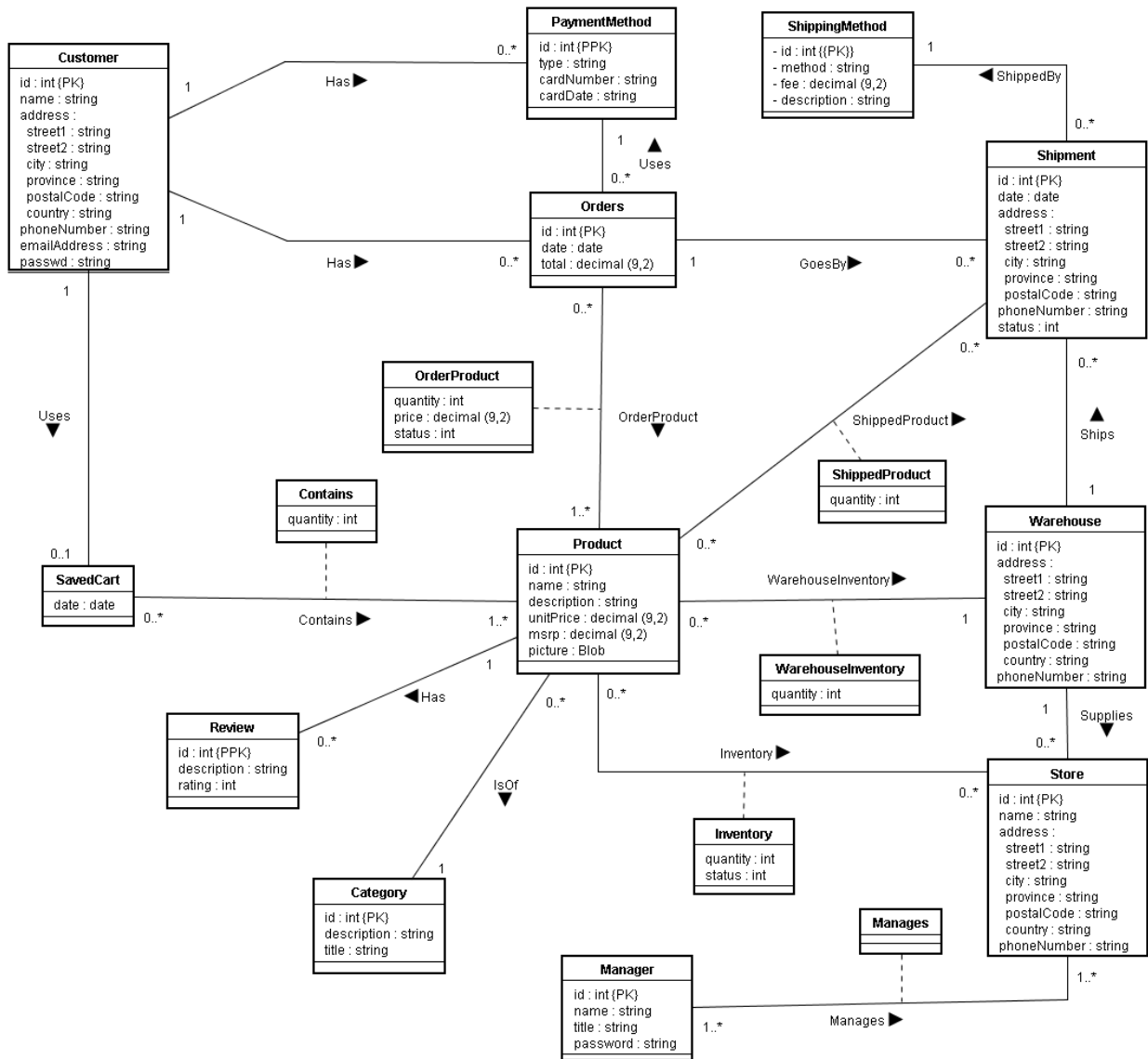


Figure 1: Sealand Entitlement Entity Relationship Model

The **Review** relation represents a comment by a customer about a Product that was bought. A Product can have multiple reviews.

The **ShippingMethod** relation contains the information about a specific shipment method. A Shipment is shipped by one of the shipping methods. A shipping method can be used by several shipments.

The **Warehouse** relation represents a warehouse. All Shipments originate from one warehouse. The warehouse can make multiple shipments. The warehouse contains products and the amount of each is stored in the WarehouseInventory relation. The warehouse supplies each store with its inventory.

The **Store** relation represents a physical store. Each store sells products. The amount of a product in a store is shown in the Inventory relation. Each store has 1 or more managers.

The **Manager** relation represents a company manager. A manager can manage 1 or more stores.

The **Manages** relation keeps track of which manager is managing which store.

The **WarehouseInventory** relation keeps track of which products are supplied from the warehouse and the amount of each product in the warehouse.

The **Inventory** relation keeps track of which products are sold at each store. It also keeps track of the amount of each product available at the stores.

Each **Customer** can have zero or more different methods of payments, but each **PaymentMethod** can only be associated with a single Person.

Each **Customer** can have zero or more different **Orders**, but each order can only be associated with a single **Customer**.

Each **Customer** can have zero or one **SavedCart**, but a **SavedCart** can only be associated with a single **Customer**.

Each order must be associated with only one **PaymentMethod**, but a **PaymentMethod** can be associated with zero or more **Orders**.

Each **SavedCart** can contain one or more products, and each **SavedCart** can contain zero or more products. An associated entity contains allows that tracking of the quantity of each product placed in the **SavedCard**.

Orders contains one or more **Product** and are related by **OrderProduct**, with the additional attributes of quantity and price. A **Product** can be associated with zero or more **Orders**.

Orders can go by zero ore more shipments but a **Shipment** must contain only 1 order.

A **Customer** does not need a **SavedCart** to shop for items, but in order to purchase items, they must put items into a **SavedCart** before proceeding to place the order. Additionally, a **Customer** can only maintain, at most, one **SavedCart**.

A **Product** contained in a **SavedCart**, will not be associated with a saved price as the price of a product may fluctuate with time, as the price shown in the interface will reflect the most recent price.

A **Customer** can have multiple payment method's for a specific **Customer** as a **Customer** may have multiple charge cards that the use, but they can only use one specific method for a given order.

Each **Product** is related to only one **Category** type, but a **Category** may contain zero or more **Products**.

5.2 Entity Attributes

Entities have been modeled to represent the required components for the system as:

5.2.1 Shipment

Attribute	Description
id {PK}	A shipment is identified by a unique id. Hence, id is the primary key of Shipment.
date	Once the shipment leaves the warehouse, a shipment date is recorded in the date attribute.
street1	the first line of the street address for the shipment address
street2	the second line of the street address for the shipment address
lemcity	the city for this address
province	the province for this address
postalCode	the postal code of the shipment
phoneNumber	If there is a problem with a shipment, the customer service might want to call the receiver of the shipment. The phoneNumber is therefore stored as a string.
shippingFee	Since shipment costs may vary, a shippingFee is stored as a decimal.
status	The status attribute indicates the present state of the shipment.

5.2.2 ShippingMethod

Attribute	Description
id {PK}	The id is the unique identifier of the relation. It is the primary key of ShippingMethod.
method	The method is a string that describes the shipping method in one word. Examples are Courier, Express, Normal, etc.
fee	The fee is the price a customer has to pay if he selects the ShippingMethod.
description	The description contains detailed information about the shipping method. As an example, the estimated shipping time could be contained in the description.

5.2.3 Product

Attribute	Description
id {PK}	Every product is identified by a unique integer id. This is the primary key of the relation.
name	The product name is mainly used by the customer, to identify the product. The name is a simple string.
description	The description of a product gives a customer a more detailed explanation of a product than the bare name. The description is a string.
unitPrice	The unitPrice is the company's cost for one unit of a product.
msrp	The suggested retail price of the product.
picture	The picture attribute is a Blob that contains an image of the product.

5.2.4 Review

Attribute	Description
pid {PPK}	The pid is a foreign key to the Product for which the review was made. Together with the id, it forms the primary key of the relation.
id	The id of the review together with the pid form the primary key of the relation.
description	A string that contains the actual review for the product.
rating	The rating is a value that represents the customer satisfaction about the product on a numerical scale. It is an integer field.

5.2.5 Warehouse

Attribute	Description
id {PK}	The warehouse is identified by unique integer id.
street1	The first line of the street address of the warehouse.
street2	The second line of the street address of the warehouse.
city	The city in which the warehouse is located.
province	The province in which the warehouse is located.
postalCode	The postal code of the warehouse.
country	The country in which the warehouse is located.
fax	The fax number of the warehouse

5.2.6 Store

Attribute	Description
id {PK}	The store is identified by unique integer id.
name	The name of the store
street1	The first line of the street address of the store.
street2	The second line of the street address of the store.
city	The city in which the store is located.
province	The province in which the store is located.
postalCode	The postal code of the store.
country	The country in which the store is located.
phoneNumber	The phone number of the store.

5.2.7 Manager

Attribute	Description
id {PK}	A manager is identified by unique integer id.
name	The name of the manager.
title	The current title of the manager.
password	The manager's password for accessing the management section of the website.

5.2.8 Customer

Attribute	Description
id {PK}	a unique number that identifies a customer.
name	a customer's complete name.
street1	the first line of the street address for this address.
street2	the second line of the street address for this address
city	the city for this address.
province	the province for this address.
postalCode	the postal code of this address.
country	the country of this address.
phoneNumber	the phone number for this customer.
emailAddress	the email address for this customer

5.2.9 SavedCard

Attribute	Description
date {PPK}	The date the shopping cart for a customer was last saved and is a PPK with customer id

5.2.10 PaymentMethod

Attribute	Description
id {PPK}	a unique number, that in conjunction with a Customer's id uniquely identifies a payment method
type	the method of payment chosen such as Visa, MasterCard, etc.
cardNumber	the number that identifies the account of the type of payment method.
cardDate	the date that the payment method expires on or is invalid past.

5.2.11 Orders

Attribute	Description
id {PK}	a unique number, that in conjunction with a Customer's id uniquely identifies an order.
date	the date on which the order was taken.
total	the total amount of all items of OrderProducts, including tax, but excluding shipping fee's.

5.2.12 Category

Attribute	Description
id {PK}	a unique number, that uniquely identifies a category.
title	the title for a given category.
description	A description of what type of items are in a category.

6 Relational Schema

The relational schema for this system is represented as SQL DDL:

```
CREATE TABLE Customer (
    id INTEGER NOT NULL,
    name VARCHAR(30) NOT NULL,
    street1 VARCHAR(50) NULL,
    street2 VARCHAR(50) NULL,
    city VARCHAR(30) NOT NULL,
    province VARCHAR(30) NOT NULL,
    postalCode VARCHAR(30) NOT NULL,
    country VARCHAR(30) NOT NULL,
    phoneNumber VARCHAR(30) NOT NULL,
    passwd VARCHAR(30) NOT NULL
);
ALTER TABLE Customer ADD CONSTRAINT Customer_PK
PRIMARY KEY (id);
CREATE TABLE PaymentMethod (
    id INTEGER NOT NULL,
    cid INTEGER NOT NULL,
    type VARCHAR(30) NOT NULL,
    cardNumber VARCHAR(30) NOT NULL,
    cardDate DATE NOT NULL
);
ALTER TABLE PaymentMethod ADD CONSTRAINT PaymentMethod_PK
PRIMARY KEY (id,cid);
CREATE TABLE Orders (
    id INTEGER NOT NULL,
    cid INTEGER NOT NULL,
    date DATE NOT NULL,
    total DECIMAL(9,2) NOT NULL,
    pm_cid INTEGER NOT NULL,
    pmid INTEGER NOT NULL,
    sid INTEGER NOT NULL
);
ALTER TABLE Orders ADD CONSTRAINT Orders_PK
PRIMARY KEY (id);
CREATE TABLE SavedCart (
    cid INTEGER NOT NULL,
    date DATE NOT NULL
);
ALTER TABLE SavedCart ADD CONSTRAINT SavedCart_PK
PRIMARY KEY (cid);
CREATE TABLE Product (
    id INTEGER NOT NULL,
    name VARCHAR(30) NOT NULL,
    description VARCHAR(2000) NULL,
    unitPrice DECIMAL(9,2) NOT NULL,
    msrp DECIMAL(9,2) NOT NULL,
    picture BLOB NULL,
    ctid INTEGER NOT NULL
);
ALTER TABLE Product ADD CONSTRAINT Product_PK
PRIMARY KEY (id);
CREATE TABLE Contains (
    cid INTEGER NOT NULL,
    pid INTEGER NOT NULL,
    quantity INTEGER NOT NULL
);
ALTER TABLE Contains ADD CONSTRAINT Contains_PK
PRIMARY KEY (cid,pid);
CREATE TABLE OrderProduct (
    oid INTEGER NOT NULL,
    pid INTEGER NOT NULL,
    quantity INTEGER NOT NULL,
    price DECIMAL(9,2) NOT NULL,
    status INTEGER NOT NULL
);
ALTER TABLE OrderProduct ADD CONSTRAINT OrderProduct_PK
PRIMARY KEY (oid,pid);
CREATE TABLE ShippingMethod (
    id INTEGER NOT NULL,
    method VARCHAR(30) NOT NULL,
    fee DECIMAL(9,2) NOT NULL,
    description VARCHAR(2000) NOT NULL
);
ALTER TABLE ShippingMethod ADD CONSTRAINT ShippingMethod_PK
PRIMARY KEY (id);
CREATE TABLE Shipment (
    id INTEGER NOT NULL,
    date DATE NOT NULL,
    street1 VARCHAR(50) NULL,
    street2 VARCHAR(50) NULL,
    city VARCHAR(30) NOT NULL,
    province VARCHAR(30) NOT NULL,
    postalCode VARCHAR(30) NOT NULL,
    phoneNumber VARCHAR(30) NOT NULL,
    status INTEGER NOT NULL,
    smid INTEGER NOT NULL,
    wid INTEGER NOT NULL,
    oid INTEGER NOT NULL
);
ALTER TABLE Shipment ADD CONSTRAINT Shipment_PK
PRIMARY KEY (id);
CREATE TABLE ShippedProduct (
    sid INTEGER NOT NULL,
    pid INTEGER NOT NULL,
    quantity INTEGER NOT NULL
);
ALTER TABLE ShippedProduct ADD CONSTRAINT ShippedProduct_PK
PRIMARY KEY (sid,pid);
CREATE TABLE Warehouse (
    id INTEGER NOT NULL,
    street1 VARCHAR(50) NULL,
    street2 VARCHAR(50) NULL,
    city VARCHAR(30) NOT NULL,
    province VARCHAR(30) NOT NULL,
    postalCode VARCHAR(30) NOT NULL,
    country VARCHAR(30) NOT NULL
);
ALTER TABLE Warehouse ADD CONSTRAINT Warehouse_PK
PRIMARY KEY (id);
CREATE TABLE WarehouseInventory (
    wid INTEGER NOT NULL,
    pid INTEGER NOT NULL,
    quantity INTEGER NOT NULL
);
ALTER TABLE WarehouseInventory ADD CONSTRAINT
WarehouseInventory_PK PRIMARY KEY (wid,pid);
CREATE TABLE Store (
    id INTEGER NOT NULL,
    name VARCHAR(30) NOT NULL,
    street1 VARCHAR(50) NULL,
    street2 VARCHAR(50) NULL,
    city VARCHAR(30) NOT NULL,
    province VARCHAR(30) NOT NULL,
    postalCode VARCHAR(30) NOT NULL,
    country VARCHAR(30) NOT NULL,
    phoneNumber VARCHAR(30) NOT NULL,
    wid INTEGER NOT NULL
);
ALTER TABLE Store ADD CONSTRAINT Store_PK
PRIMARY KEY (id);
CREATE TABLE Inventory (
    stid INTEGER NOT NULL,
    pid INTEGER NOT NULL,
    quantity INTEGER NOT NULL,
    status INTEGER NOT NULL
);
ALTER TABLE Inventory ADD CONSTRAINT Inventory_PK
PRIMARY KEY (stid,pid);
CREATE TABLE Review (
    id INTEGER NOT NULL,
    pid INTEGER NOT NULL,
    description VARCHAR(2000) NOT NULL,
    rating INTEGER NOT NULL
);
```

```

ALTER TABLE Review ADD CONSTRAINT Review_PK
PRIMARY KEY (id,pid);
CREATE TABLE Manager (
  id INTEGER NOT NULL,
  name VARCHAR(30) NOT NULL,
  title VARCHAR(30) NOT NULL,
  password VARCHAR(30) NOT NULL
);
ALTER TABLE Manager ADD CONSTRAINT Manager_PK
PRIMARY KEY (id);
CREATE TABLE Manages (
  mid INTEGER NOT NULL,
  stdid INTEGER NOT NULL
);
ALTER TABLE Manages ADD CONSTRAINT Manages_PK
PRIMARY KEY (mid,stdid);
CREATE TABLE Category (
  id INTEGER NOT NULL,
  title VARCHAR(30) NOT NULL,
  description VARCHAR(2000) NOT NULL
);
ALTER TABLE Category ADD CONSTRAINT Category_PK
PRIMARY KEY (id);

ALTER TABLE SavedCart ADD CONSTRAINT Uses FOREIGN KEY (cid) REFERENCES Customer (id) ON DELETE CASCADE;
ALTER TABLE PaymentMethod ADD CONSTRAINT Has FOREIGN KEY (cid) REFERENCES Customer (id) ON DELETE SET NULL;
ALTER TABLE Orders ADD CONSTRAINT Has FOREIGN KEY (cid) REFERENCES Customer (id) ON DELETE SET NULL;
ALTER TABLE Orders ADD CONSTRAINT Uses FOREIGN KEY (pm_cid,pmid) REFERENCES PaymentMethod (cid,id) ON DELETE SET NULL;
ALTER TABLE Shipment ADD CONSTRAINT GoesBy FOREIGN KEY (oid) REFERENCES Orders (id) ON DELETE SET NULL;
ALTER TABLE Shipment ADD CONSTRAINT ShippedBy FOREIGN KEY (smid) REFERENCES ShippingMethod (id) ON DELETE SET NULL;
ALTER TABLE Shipment ADD CONSTRAINT Ships FOREIGN KEY (wid) REFERENCES Warehouse (id) ON DELETE SET NULL;
ALTER TABLE ShippedProduct ADD CONSTRAINT SP1 FOREIGN KEY (sid) REFERENCES Shipment (id) ON DELETE CASCADE;
ALTER TABLE ShippedProduct ADD CONSTRAINT SP2 FOREIGN KEY (pid) REFERENCES Product (id) ON DELETE CASCADE;
ALTER TABLE OrderProduct ADD CONSTRAINT R_11 FOREIGN KEY (oid) REFERENCES Orders (id) ON DELETE CASCADE;
ALTER TABLE OrderProduct ADD CONSTRAINT R_12 FOREIGN KEY (pid) REFERENCES Product (id) ON DELETE CASCADE;
ALTER TABLE Contains ADD CONSTRAINT R_13 FOREIGN KEY (cid) REFERENCES SavedCart (cid) ON DELETE CASCADE;
ALTER TABLE Contains ADD CONSTRAINT R_14 FOREIGN KEY (pid) REFERENCES Product (id) ON DELETE CASCADE;
ALTER TABLE Review ADD CONSTRAINT Has FOREIGN KEY (pid) REFERENCES Product (id) ON DELETE SET NULL;
ALTER TABLE WarehouseInventory ADD CONSTRAINT R_16 FOREIGN KEY (pid) REFERENCES Product (id) ON DELETE SET NULL;
ALTER TABLE WarehouseInventory ADD CONSTRAINT R_17 FOREIGN KEY (wid) REFERENCES Warehouse (id) ON DELETE CASCADE;
ALTER TABLE Store ADD CONSTRAINT Supplies FOREIGN KEY (wid) REFERENCES Warehouse (id) ON DELETE SET NULL;
ALTER TABLE Inventory ADD CONSTRAINT R_20 FOREIGN KEY (pid) REFERENCES Product (id) ON DELETE SET NULL;
ALTER TABLE Inventory ADD CONSTRAINT R_21 FOREIGN KEY (stdid) REFERENCES Store (id) ON DELETE CASCADE;
ALTER TABLE Manages ADD CONSTRAINT R_22 FOREIGN KEY (stdid) REFERENCES Store (id) ON DELETE CASCADE;
ALTER TABLE Manages ADD CONSTRAINT R_23 FOREIGN KEY (mid) REFERENCES Manager (id) ON DELETE CASCADE;
ALTER TABLE Product ADD CONSTRAINT contains FOREIGN KEY (ctid) REFERENCES Category (id) ON DELETE SET NULL;
COMMIT;

```

7 User Interface and Planned Features

7.1 User Interface

The main page of the web store will consist of a list of product categories along with a search bar and a link to the login page. It might also include some featured products. Each category listed on the main page is a link to a list of all the products that belong in that category. The product list will be broken up into multiple pages to prevent the list of products on any page from becoming too long.

The search bar will allow the user to select either all products or a specific category of products to search. The user will enter keywords and the search will display the results over multiple pages with a maximum number of results per page.

Each product page will display a picture of the item along with a title, description and customer reviews for the item. The user can then add the product to the online shopping cart or navigate to another page on the site. The user can also add any product from the search results or category pages to the online shopping cart.

At any time the customer will be able to click the *checkout* link to order the items in the shopping cart. The user will have to be logged in to checkout an order. At this point the user will enter a shipping address and select a payment method. The payment method can either be a previously saved payment method or a new one the user will enter during the checkout.

A *Sign In* link on every page will allow the user to log into the website using the login page. Once logged on the customer can navigate to the *My Account* page where the user will be able to track orders, add payment methods or review any product that the user has ordered.

Managers in the company will be allowed access through the main web page of the site. The manager page will allow managers to run functions associated with their specific roll and will only access to functions that they are entitled to perform, such as running sales reports.

7.2 Planned Features

Based on the planned user interface, the following features will be offered to guests and customers through the online store. Not all items are available to guests that do not have an active account.

- Browse for products by category
- Search for products by
 - Category
 - Description
- View at least one picture for each product and read any associated review.
- Maintained a shopping cart
- Checkout and place an order
- View status of orders placed
- Account creation and maintenance

Managers will have access to additional features related to the operation of the company. Not all managers will have access to all the items.

- Recover the system.
- Add, remove, or update item currently for sale.
- Track inventory levels of items in central warehouse and modify sales status
- Generate simple sales reports including frequent items, sales volume, etc.

References

- [1] Sealand, *The Principality of Sealand*, <http://www.sealandgov.org>
- [2] SealandEntitlements NARLLC , not a real limited liability company, not even close.