

CPS 630

Project Report Itr1-2

Team 20

Technical Report for Smart Customer Services (S.C.S) Web application

Team Members:

Name	Student Number
Yasser Ammar Abady	500911995
Saif Ahmed Alvi	500834841
Tusaif Azmat	500660278

Tasks Done By Each Team Member:

Name	Tasks Done	Percentage
Yasser Ammar Abady	Sign in system, About Us/Contact, General debugging, frontend, Technical Report	33.3%
Saif Ahmed Alvi	Service page(a), Service page(b), Service page(c), Shopping cart, Geo location Map and Technical Report	33.3%
Tusaif Azmat	Database maintain mode pages, Technical Report, Backend, Service page(d), general bootstrap styling	33.3%
Total		100%

Objectives:

Over the past few years, the living conditions of everyone have been affected heavily by the carbon emissions caused by industrial waste or transportation. Several solutions have been developed in order to tackle this problem, and they all show promising results. The project applies the “Smart Customer Services” (SCS) Web application methodology as a means of creating potential services to help reduce air pollution. The web application is developed under XAMPP Apache, MySQL, and PHP with MVC architecture. Additionally, the web app includes open libraries such as Google Maps, SASS, and JQuery to assist the system.

Languages:

HTML: used for website content

CSS: used to style whole website

JQuery: used for animations

JavaScript: used to extract data from database

PHP: used to process requests for the forms

SQL: used to create, insert, modify, etc

Tools: XAMPP, W3C, Notepad++

Implementation:

The web application will have 2 different modes: user mode and admin mode. Admin will have all the privileges to view the databases, as well as create, delete and modify records. Users will be able to see their personal information, orders, map, store, products. To distinguish between the 2 modes, the database was set to collect user type. The “user” 1 type is set by default so that everyone with access to the website can make their personal account. The “admin” type, however, can only be set up by the developer/root admin. To differentiate between the user types, we use the SESSION from PHP to record the ‘username’. If the username is admin, we will allow them to have access to “DB maintain” on the top navbar.

A. Design & Implementation of database maintain mode:

In the following sections will include all the implementation of SQL codes for all the transactions.

1. Creating:

```
CREATE TABLE USERS (
    USER_ID INTEGER NOT NULL AUTO_INCREMENT PRIMARY KEY,
    USER_NAME VARCHAR(50),
    USER_TEL VARCHAR(20),
    USER_EMAIL VARCHAR(50),
    USER_ADDRESS VARCHAR(100),
    USER_CITY_CODE VARCHAR(20),
    USER_USERNAME VARCHAR(100),
    USER_PASSWORD VARCHAR(100),
    USER_BALANCE FLOAT,
    USER_ADMIN BOOLEAN
);

CREATE TABLE TRUCK (
    TRUCK_ID INTEGER NOT NULL AUTO_INCREMENT PRIMARY KEY,
    TRUCK_CODE VARCHAR(50),
    AVAILABILITY_CODE VARCHAR(50)
);

CREATE TABLE ITEMS (
    ITEM_ID INTEGER NOT NULL AUTO_INCREMENT PRIMARY KEY,
    STORE_CODE VARCHAR(50),
    ITEM_NAME VARCHAR(50),
    MADE_IN VARCHAR(50),
    DEPART_CODE VARCHAR(50),
    PRICE FLOAT
);
```

```

CREATE TABLE TRIP (
    TRIP_ID INTEGER NOT NULL AUTO_INCREMENT PRIMARY KEY,
    DESTINATION_CODE VARCHAR(100),
    SOURCE_CODE VARCHAR(100),
    DISTANCE FLOAT,
    TRUCK_ID_FK INT,
    ITEM_ID_FK INT,
    PRICE FLOAT,
    FOREIGN KEY (TRUCK_ID_FK) REFERENCES TRUCK (TRUCK_ID),
    FOREIGN KEY (ITEM_ID_FK) REFERENCES ITEMS (ITEM_ID)
);

CREATE TABLE ORDERS (
    ORDER_ID INTEGER NOT NULL AUTO_INCREMENT PRIMARY KEY,
    DATE_ISSUED TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
    DATE_DONE DATE NOT NULL,
    TOTAL_PRICE FLOAT,
    PAYMENT_CODE VARCHAR(100),
    USER_ID_FK INT,
    TRIP_ID_FK INT,
    ITEM_ID_FK INT,
    FOREIGN KEY (USER_ID_FK) REFERENCES USERS (USER_ID),
    FOREIGN KEY (TRIP_ID_FK) REFERENCES TRIP (TRIP_ID),
    FOREIGN KEY (ITEM_ID_FK) REFERENCES ITEMS (ITEM_ID)
);

CREATE TABLE SHOPPING (
    RECEIPT_ID INTEGER NOT NULL AUTO_INCREMENT PRIMARY KEY,
    ORDER_ID_FK INT,
    DATE_ISSUED TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
    DATE_DONE DATE NOT NULL,
    TOTAL_PRICE FLOAT,
    STORE_CODE VARCHAR(100),
    ITEM_ID_FK INT,
    FOREIGN KEY (ORDER_ID_FK) REFERENCES ORDERS (ORDER_ID),
    FOREIGN KEY (ITEM_ID_FK) REFERENCES ITEMS (ITEM_ID)
);

CREATE TABLE PRODUCT_REVIEW (
    REVIEW_ID INTEGER NOT NULL AUTO_INCREMENT PRIMARY KEY,
    R_CONTEXT VARCHAR(200),
    R_SCORE INT,
    ITEM_ID_FK INT,
    FOREIGN KEY (ITEM_ID_FK) REFERENCES ITEMS (ITEM_ID)
);

CREATE TABLE TRUCK_REVIEW (
    TRUCK_ID INTEGER NOT NULL AUTO_INCREMENT PRIMARY KEY,
    R_CONTEXT VARCHAR(200),
    R_SCORE INT,
    TRUCK_ID_FK INT,
    FOREIGN KEY (TRUCK_ID_FK) REFERENCES TRUCK (TRUCK_ID)
);

```

2. Adding

```

/**
 * Populate Tables with some data
 */
USE SCS_CPS630_PROJECT;

insert into USERS (USER_NAME, USER_TEL, USER_EMAIL, USER_ADDRESS, USER_CITY_CODE,
    USER_USERNAME, USER_PASSWORD, USER_BALANCE, USER_ADMIN) values
('Username 1', '3814553378', 'Username1@email.com', '150 Sunnybrook Road,
    Toronto', 'M4C 5K5', 'afortune0', MD5('m614oAR4'), 500.00, false);

insert into TRUCK (TRUCK_CODE, AVAILABILITY_CODE) values ('TRUCK_01', 'true');

insert into ITEMS (STORE_CODE, ITEM_NAME, PRICE, MADE_IN, DEPART_CODE)
values ('Store_01', 'Furniture_01', 3000.00, 'China', 'Furniture');

insert into TRIP (DESTINATION_CODE, SOURCE_CODE, DISTANCE, TRUCK_ID_FK, ITEM_ID_FK, PRICE)
values ('42 Landsdown Street', '318 Marker Circle', 16, 2, 1, 3022.46);

insert into ORDERS (DATE_DONE, TOTAL_PRICE, PAYMENT_CODE, USER_ID_FK, TRIP_ID_FK, ITEM_ID_FK)
values ('2020-09-05', 1186.34, '50385067829721017', 1, 1, 19);

insert into PRODUCT_REVIEW (R_CONTEXT, R_SCORE, ITEM_ID_FK)
values ('Distributed high-level support', 4.1, 21);

```

3. Deleting:

```
public function deleteUser ($id) {  
    $sql = "DELETE FROM Users WHERE USER_ID = '$id'";  
    $query = mysqli_query($this->conn, $sql) or die (mysqli_error($this->conn));  
    return $query;  
}
```

```
public function deleteItem($id) {  
    $sql = "DELETE FROM Items WHERE ITEM_ID = '$id'";  
    $query = mysqli_query($this->conn, $sql) or die (mysqli_error($this->conn));  
    return $query;  
}
```

4. Modifying:

```
public function updateById($id, $storecode, $price, $madein, $name, $departcode) {  
    $sql = "UPDATE Items  
    SET STORE_CODE='$storecode', PRICE='$price', MADE_IN='$madein' , ITEM_NAME='$name' , DEPART_CODE='$departcode'  
    WHERE FLOWER_ID = '$id'";  
    $query = mysqli_query($this->conn, $sql) or die (mysqli_error($this->conn));  
    return $query;  
}
```

```
public function updateById($id, $code, $avail) {  
    $sql = "UPDATE Truck  
    SET Truck_CODE= '$code', AVAILABILITY_CODE='$avail'  
    WHERE Truck_ID = '$id'";  
    echo $sql;  
    $query = mysqli_query($this->conn, $sql) or die (mysqli_error($this->conn));  
    return $query;  
}
```

5. Searching:

```
public function getSpecificTruck($id){  
    $sql = "SELECT * FROM Truck WHERE Truck_ID in ($id)";  
    $query = mysqli_query($this->conn, $sql) or die (mysqli_error($this->conn));  
    return $query;  
}
```

```
public function getSpecificItem($id){  
    $sql = "SELECT * FROM Items WHERE ITEM_ID in ($id)";  
    $query = mysqli_query($this->conn, $sql) or die (mysqli_error($this->conn));  
    return $query;  
}
```

```
public function search($search, $id){  
    $sql = "SELECT * FROM Orders INNER JOIN Items on Orders.Item_ID_FK = Items.Item_ID WHERE Order_ID=$id AND USER_ID_FK=$search";  
    $query = mysqli_query($this->conn, $sql) or die(mysqli_error($this->conn));  
    return $query;  
}
```

B. Design & layout of application interfaces UI:

1. Front end:

Each page will contain certain information, functionality that assists the user. All pages will navigate through a Navbar component.

The website will contain the following pages:

- **Home page**
 - **About us**
 - **Contact us**
 - **Sign up**
 - **Reviews**
 - **Shopping cart**
 - **Type of Service**
-
- **The Home page:** where all the information about the web is displayed. This will give a quick overview of the page along with its features. On this page, other content will be included such as the "About Us" page - a brief bio description of Team 20 members as well as their contribution to the site, and the form "Contact us".
 - **The Signup page:** will lead the user to a form where they could fill up their information for signing up for an account. After they have successfully created an account, they can log in to the system using the Login form.
 - **The Shopping Cart** is where the user can see their current orders. This page will 5 summarize all the prices along with all items the customer has ordered. By default, this page will show an empty cart. However, once the user has selected items on the "Type of service" page, the interface will be changed accordingly to the orders.
 - **The Type of Service** will toggle users between services: Furniture service or Truck Delivery driving service. The user can use drag and drop to add items to their cart. Once the user decides to process a payment, they will be redirected to the order page or their "Cart".
 - **The Review page:** Users can read over the reviews for a product or a Truck driver, however, they will not be able to make any review/ rating unless they have previously ordered/got delivered from that specific driver or bought the specific flower type. They will also not be able to use the review page if they have not logged in.

2. Back end:

There are 2 main components that made up the back-end, which are the relational database system and the MVC back-end.

For the relational database system, we have decided to create the following table

Table	Description	Attributes	# of Records
Users	Information about customers, including user name and hashed password	ID, name, telephone, email, address, city code, username, password, balance, admin (To check for users)	10
Items	Information about the store and price of the product	ID, Store code, Item name, Made in, Department code, price	22
Orders	Information about customer orders	ID, date issued, date done, total price, payment code, User ID, trip ID, Truck ID	10

Truck	Information about the store delivery trucks and its availability	ID, Truck code, Availability	10
Trip	Information about the trip	ID, destination code, source code, distance, Truck id, price	10
Product Review	All the reviews on product Items	ID, context, score, Item ID	5
Truck Review	All the reviews on delivery truck driver	ID, context, score, Truck ID	5

:

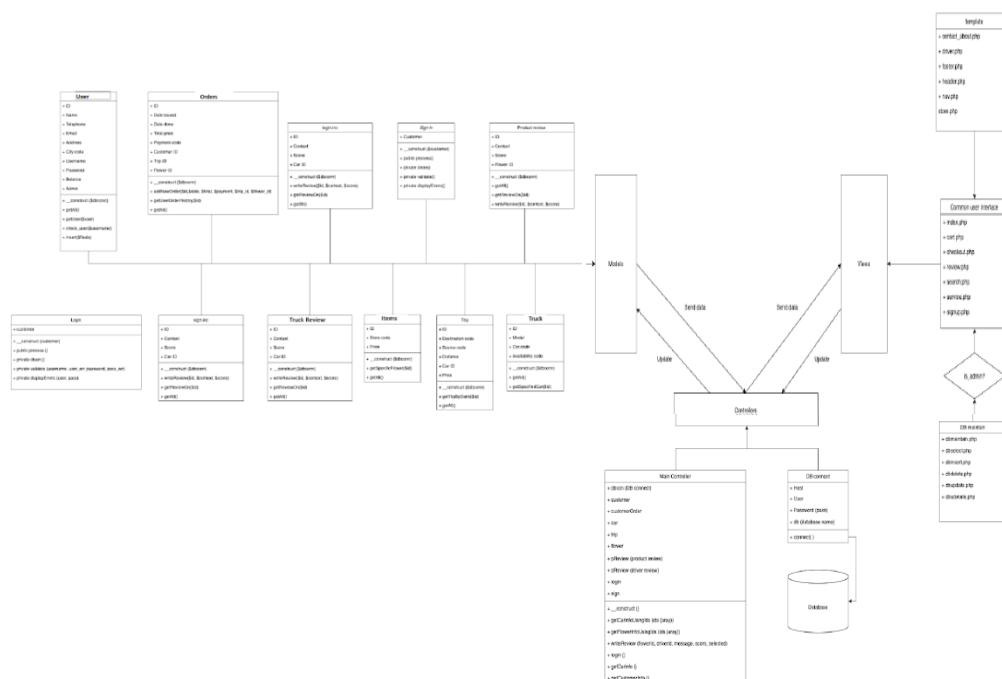
There will be separated scripts to generate the tables according to the design board above, most of the data are consistent, however, the orders total price is inconsistent with the add up of trip price and the product type, since they were generated for demo purpose. However, once a new user is created, the new records will be much more consistent comparing to existed ones.

Additionally, customer passwords will be hashed once they are recorded in the database. Once the user decided to log in to the system, the database will decipher the hashed code and check for matching credentials. If the record exists, the system will populate accordingly to the type of the account (Since there will be a dashboard system for admin).

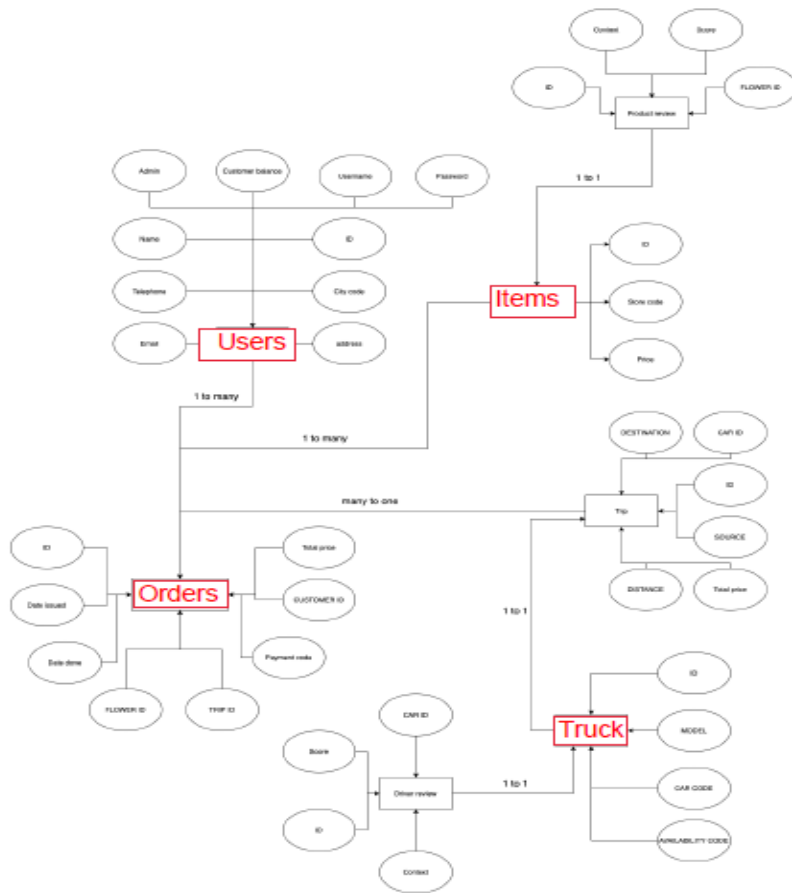
As for the back-end system, it was designed and implemented by the MVC pattern and object-oriented architecture. There will be 7 different classes with different methods to implement to the web application. These 7 classes match all the tables from the database and will be used as our Model. These models will be called in the main Controller, which will respond accordingly based on the parameters. Afterwards, the front-end (View) will be updated whenever it makes a request to the back-end side.

DESIGN DIAGRAM:

The following diagrams are the overall design for our system

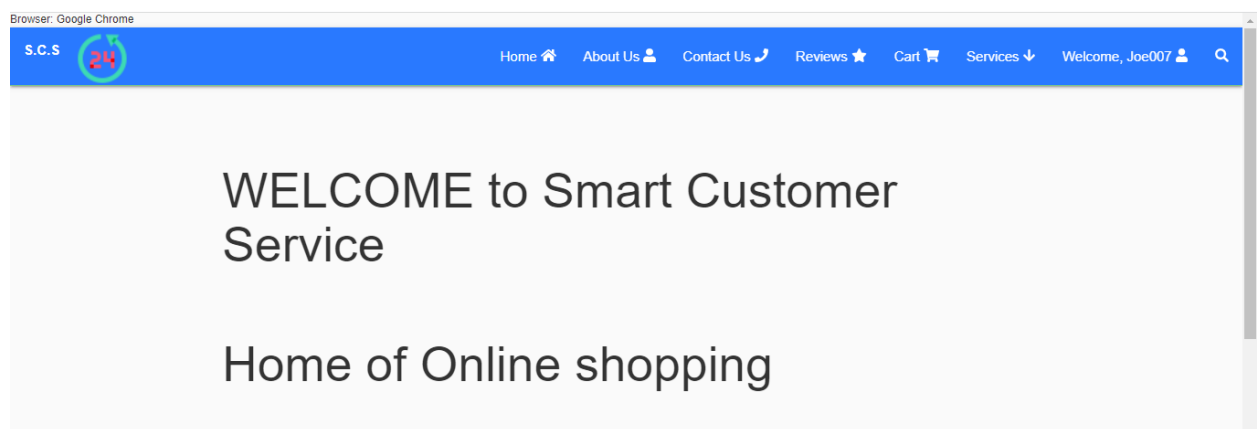


Database Diagram:










RESULTS:

These are some of the images of our web application:




Browser: Google Chrome









S.C.S 

Home  About Us  Contact Us  Reviews  Cart  Services  Sign out
profile

WELCOME to Smart Customer Service

Browser: Google Chrome



S.C.S 

Home  About Us  Contact Us  Reviews  Cart  Services  Welcome, Joe007  

You're Logged Off









© 2022 - 2022 S.C.S Smart Customer Services - All Rights Reserved.
Last Updated : 03/15/2022 20:33:37

Browser: Google Chrome

S.C.S  Home 

Please Log in or Sign Up to use the system

Warning: This document is for internal use only. It is not to be distributed outside the project. Project: c:\pvt\30_final_project\main\1_1_2\template\new.php on line 2

Home  About Us  Contact Us  Reviews  Cart  Services  Sign Up  

Login

Username

Password

Sign Up


First Name Last Name






Address City Code



Username

Telephone

Browser: Google Chrome

S.C.S 

Home  About Us  Contact Us  Reviews  Cart  Furniture
Appliances
Service A
Service B

Welcome, Joe007  

WELCOME to Smart Customer Service

Home of Online shopping

Browser: Google Chrome

S.C.S. Home About Us Contact Us Reviews Cart Services Welcome, Joe007

Furniture Store



Source location

Destination location

SHOW MAP

Choose a Truck For Delivery:

Choose a Truck

Item code	Price
 Store_01	2000\$
 Store_02	3000\$

Browser: Google Chrome

S.C.S. Home About Us Contact Us Reviews Cart Services Welcome, Joe007

Furniture Store

Source

Etobicoke, ON M9C, Canada


Destination

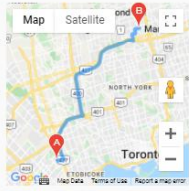
Richmond Hill, ON L4B, Canada

SHOW MAP

Choose a Truck For Delivery:

TRUCK_01

Item code	Price
 Store_01	2000\$





Browser: Google Chrome

S.C.S. Home About Us Contact Us Reviews Cart Services Welcome, Joe007

Your Cart

Shopping Cart

Empty Cart

Name	Code	Quantity	Unit Price	Price	Remove
 Furniture_item_03	Furniture_name_03	1	\$ 1700.00	\$ 1,700.00	
Total:		1	\$ 1,700.00		

PROCEED TO CHECKOUT

Browser: Google Chrome

S.C.S



localhost says
Order Placed

OK

Your Cart

Shopping Cart

Empty Cart

Name	Code	Quantity	Unit Price	Price	Remove
 Furniture_item_03	Furniture name_03	1	\$ 1700.00	\$ 1,700.00	
Total		1	\$ 1,700.00		

PROCEED TO CHECKOUT

Browser: Google Chrome

S.C.S

Home Db maintain About Us Contact Us Reviews Cart Services Welcome, Joe007

WELCOME to Smart Customer Service

Home of Online shopping

Browser: Google Chrome

S.C.S

Home Db maintain About Us Contact Us Reviews Cart Services Welcome, Joe007

Data Base Administration Operations

SELECT

Tables

Truck

Submit

© 2022 - 2022 S.C.S Smart Customer Services - All Rights Reserved.
Last Updated : 03/15/2022 20:42:57

Browser: Google Chrome

S.C.S

Home Db maintain About Us Contact Us Reviews Cart Services Welcome, Joe007

Data Base Administration Operations

SELECT

INSERT

DELETE

UPDATE

© 2022 - 2022 S.C.S Smart Customer Services - All Rights Reserved.
Last Updated : 03/15/2022 20:42:57