Coupon Crazy

Final Report

November 29th, 2017

Group Members:

Christian Bush, Avan Patel,

Cameron Platt, Nathan Thomas

Group: H

Lab Instructor:

Ben Wiggins

1. Team Member Responsibilities

1.1. Project Requirement Phase

Avan Patel: Responsible for 4 special requirements and Guest Interface

Cameron Platt: Responsible for 3 Special requirements and Initial Task and Roles

Christian Bush: Responsible for 4 special requirements and overall outlook of document

Nathan Thomas: Responsible for 4 special requirements

1.2. Design Phase

Avan Patel: Responsible for Overview Class Diagram and 2 Detailed Class Diagram and 1 Sequence Diagram

Cameron Platt: Responsible for 1 Detailed Class Diagram and 1 Sequence Diagram and Task & Role

Christian Bush: Responsible for 2 Detailed Class Diagram and 1 Sequence Diagram and Overall outlook of document

Nathan Thomas: Responsible for 1 Detailed Class Diagram and 1 Sequence Diagram and Database Design

1.3. Implementation Phase

*Avan Patel***:** Responsible for overall outlook of all the pages, html forms for: 3 settings pages for different users, login/signup page, Implementation report.

Cameron Platt: Responsible for the production data, Implementation Report

Christian Bush: Responsible for all front end scripts and design.

Nathan Thomas: Responsible for all the Database Design and PHP validation, making sure database can communicate with all pages.

2. Group Pluto Account Username and Password

Username: cbb330

Password: couponcrazy

3. Group Pluto MySQL Account Password

Username: cbb330

Password: ab1234

4. Login Info

4.1. Administrator

UserName: Avan

Email: avan@yahoo.com

password: Avan123

user: ADMIN

UserName: Christian

Email: chris@gmail.com

password: 1234CHRIS

user: ADMIN

UserName: Cameron

Email: cameron@outlook.com

password: Cameron

user: ADMIN

UserName: Nathan

Email: nathan@microsoft.com

password: 098Nathan

user: ADMIN

4.2. Store Owner

UserName: Bobby

Email: bobby@gmail.com

password: bob123

user: Store Owner

UserName: Alice

Email: Alice@yahoo.com

password: Alice123

user: Store Owner

UserName: Jimmy

Email: jimmy@gmail.com

password: password

user: Store Owner

4.3. User

UserName: User1

Email: user1@gmail.com

password: password1

user: User

UserName: User2

Email: user2@gmail.com

password: password2

user: User

UserName: User3

Email: user3@gmail.com

password: password3

user: User

5. Final Database Design

6. Directory Names of all Source Files

DataAccess:

    CCdb.php

    dblogin.php

FrontEnd:

    adminSetting.php

    checkout.php

    index.php

    login.php

    logout.php

    RedemptionSim.php

    resetPassword.php

    signup.php

    storeOwnerSetting.php

    userSetting.php

Models:

    Admin.php

    CCExecption.php

    Coupon.php

    Store.php

    StoreOwner.php

    User.php

7. Requirements Implemented

7.1. Required:

Add Coupon To Card

Add New Inventory

Add Store

Adjust Coupon Quantity

Browse Inventory

Card Swipe Simulation

Check Out

Login

Logout

Reset Password

Sign up

View Coupon

7.2. Medium:

Delete Account

Remove Store

8. Requirements Not Used or Changed

Search Inventory: We did not implement a search inventory in our final design. This was a medium requirement that we decided was not necessary for our overall design. This would have taken a lot of extra work to implement that would only add a minor convenience to our final project. Thus we decided to exclude it from our final project as the coupons would all be present from the start instead of needing to be searched.

9. Design Phase Deviations

User, StoreOwner, Admin, Coupon, Store: Added validation functions for all input. Added CCException objects as attributes to save validation errors.

Coupon, Store: Added argument bool new to constructor to check whether data is new data. Default true.

Admin: Added argument bool hashed to see if password already hashed. Unused. Default false.

CCException: Added a new class, CCException, which inherits from PHP's built in Exception class. Stores an associative array of error messages, where the key is a general and arbitrary exception type, such as "username" for a username error. Can add errors, get all errors, get whether there are any errors, and get a printable string of errors.

CCdb: Renamed passwordIsReset() (no arguments) to passwordsReset(). Removed returns for all non-getting functions. Removed changeUsername(). Removed query(str query_str, str arr args). Added hasCoupon(str username, str coupon). Added sanitize(str input).

Database: Changed all fields to be case-sensitive. It was unspecified before, so the default was case-insensitive.

10. Extra Information

Reset Password: Once a user requests to reset a password. Admin is notified of the change in the settings page, and can reset the password to 1234. Once that is done the admin can send an email to the user with updated password. The user can use this password to change the password on User Settings page. We did not implement the part where we send email to the user automatically. The Admin must physically send email to the User with the reset password.

Login: Login is case-sensitive, so please use exact information provided.

Add Coupon/Redemption Page: Consider Redemption Page as Physical Store. The items listed on the redemption page represent the items or inventory that each store COULD BE selling. This is a hardcoded list. If a Store Owner were to add a coupon to his store, he would have to make sure it matches with an inventory item that is being sold by said store. If the Coupon Name you add as a Store Owner does not match exactly with the Coupon Name in your store, a user will not be able to redeem the Coupon with that item. If a store owner attempts to create a coupon with a name that matches a preexisting coupon,

it will throw an error. Below is the list of Coupons you can add that will match your stores hardcoded inventory which do not already exist in the database:

Charmin Toilet Paper

Nutella

Glad Trash Bags

Swanson Broth

Planters

Reynolds Wrap Foil