

Final Report: GRK Clothing Warehouse

Gary Li, Ronald Tran, Kevin Unarce (Group 16)

December 2018

1 Abstract

Our application attempts to address issues with online shopping through a user friendly website. Customers sometimes have trouble with locating items they would like to purchase, as not every store will have all items in stock at any given time. This causes more searching than we believe is necessary. The website attempts to streamline this process by providing users with access to all types of clothing, giving them a list of stores that carry said clothes, along with the quantity remaining on the item. Our website caters to two different types of users: customers and admins. Customers have the ability to look for clothes and modify their account, while admins have the ability to add, edit, and delete shops, items (and their quantities), and employees.

2 Introduction

GRK Clothing Warehouse is our solution to customer's problems with having to go to several different websites to find who has the items they want in stock. Our website allows customers to locate items of interest quickly and efficiently. Users can either search a particular store to see what items they have in stock, or search for an item that returns the stores that carry the item and the quantity remaining.

3 Project Design

3.1 User Types:

1. The customer:

Customers are people who are looking to purchase items and would like to locate the stores that carry them. At the beginning of our web-page, there are three different bodies: existing/returning users, new customer accounts, and new admins. New users will have to create an account before having the ability to access anything else. After creating a new account, customers will be notified that their account has been successfully added. After successfully creating a new account and clicking continue, a dropdown menu containing all functionalities will appear. Customers have the following options available to them: check inventory of a store, search for an item, and account profile. Other options appear in the dropdown menu, such as: update quantities, add a new shop, add new item, delete employee, add employee and modify employee. If a customer attempts to access these options, however, they will be notified that only admins have the ability to access these options.

If the user decides to search by store inventory, they will be redirected to a new page containing a dropdown menu with all stores currently in the database and a sort function. The customer will be able to choose a store they want to search and sort items by high/low quantities and high/low price (optional). Once satisfied, the customer hits 'submit' and a list of items will appear informing the customer of the store location, the price and the quantity of the item. Once the user is done, they can

click return to go back to the previous page.

Now if the user wants to search for a particular item, they will be directed to a new page with only a search bar. With this you can search for an item that is in the database. Search uses pattern matching, so if the user searches for "s" it will return all results that contain "s" anywhere within its name (Green T-Shirt, Green Pants, Red Socks, etc. will be returned). The result shows the item name and where the item can be found. Once the customer is satisfied they can return to the option screen.

The last thing the customer has access to is their profile, which allows customers to change their password, make a wish list, and choose your form of payment (debit or credit or leave blank). Customers can also log out of their account, returning them to the home page, and delete their account, which asks the customer for confirmation to delete their account. If the customer is satisfied, they can return to the options menu by selecting return.

2. Admin:

New admins can create a new account as well. The only difference between creating a new admin and creating a new customer account is that the admin must know the admin code to create their account, which is admin123. If they do not know the code, they will not be able to create an admin account, which prevents anyone from editing the website. Existing admins can log in by using 'existing account'.

The admin has the same functionality as a customer, but also has the ability to access more options. If an admin wants to update an item quantity, a table will appear with all the items in the database. There is a dropdown menu that allows admins to choose the item and the quantity they want to update. Once done they can return to the option menu.

If an admin wishes to add a new item to the database, there will be four fields that the admin has to fill in. After entering the information and submitting them, the new item will be added to the database. Once done they can return to the option menu.

If an admin wants to manage shops, they have three different options to work with: adding a new shop, edit an existing shop's information, and to update a shops inventory.

For adding a new shop, the admin will have to put a name of a store and the location of that store. For editing existing shops, a dropdown list shows the stores that can be modified. The admin fills in the fields that need to be edited: number of employees, profit and revenue.

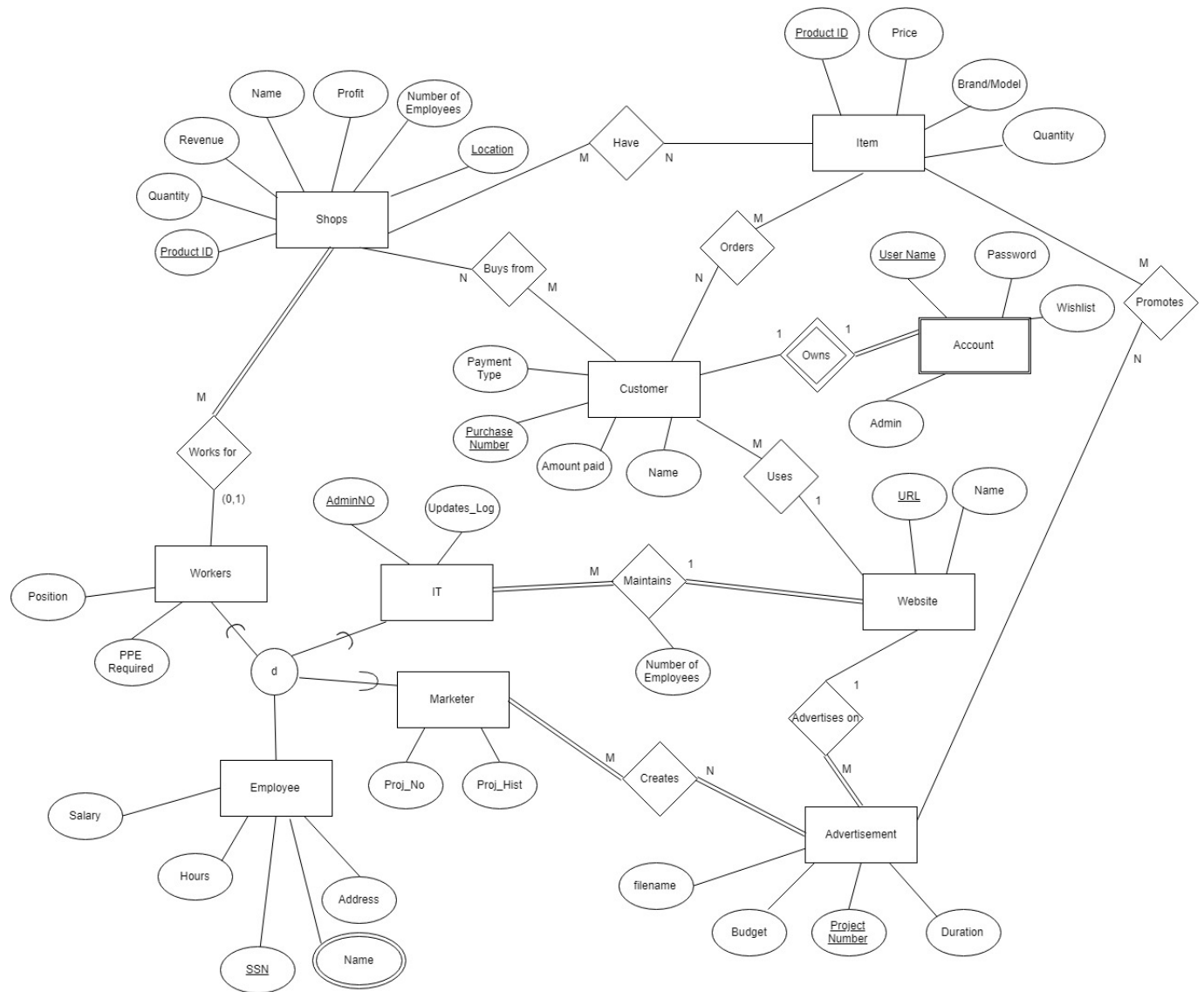
For updating shop inventory, the admin will choose which store they want to put the item in, and entering the item's product id. Once the admin is complete, and wishes to go back, click return.

If the admin wishes to add a new employee, they will see seven fields: ssn, first name, last name, salary, home address, hours, and job type. The admin will fill in the information of this employee. Once hitting submit, the new employee will be added to the database. If the admin wishes to return to options, click return.

If the admin wants to delete an employee, they will select the ssn of the employee they wish to remove from the dropdown menu and select delete. Once delete is selected, the employee will be removed from the database.

If the admin wants to modify an employee, the admin chooses an ssn from the dropdown menu and fills in with the information and submit after fields are filled in. Once completed, return brings the admin back to the option screen.

3.2 ER Diagram

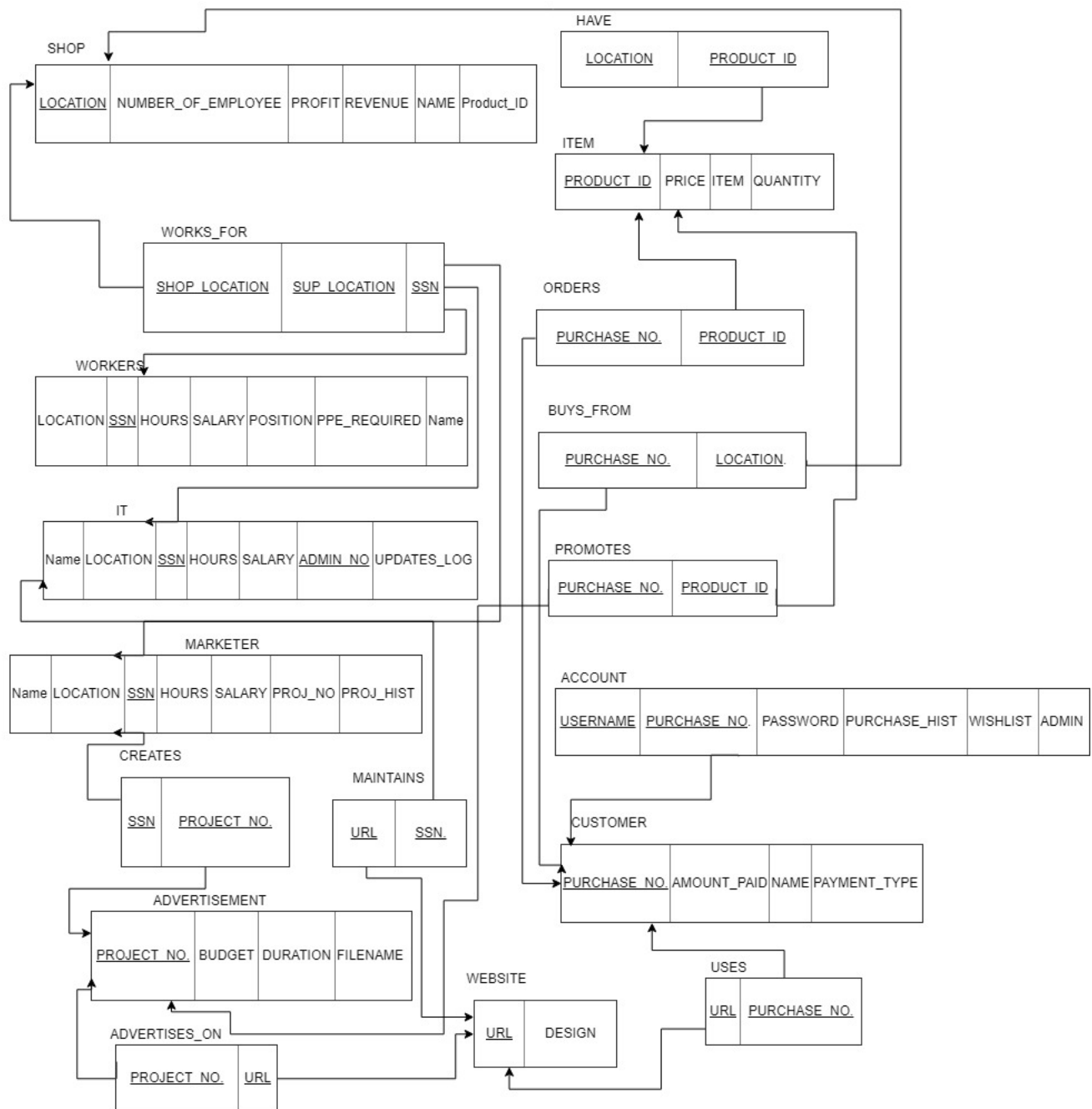


General Notes:

- We removed the entity "Supplier" and its relationship with "Shops" as we did not end up using it in our final application.
- Minor tweaks were adding more attributes to entities (Admin in "Account", filename in "Advertisement")

4 Implementation

4.1 RM Diagram



General Notes:

- We decided to keep "Account" as a weak entity rather than using it as an attribute in "Customer" in the end because we felt as though Accounts would not exist without Customers, and using Account as an entity allowed us to streamline some of the features within our application.

4.2 DBMS

We used phpMyAdmin as our database for our website. Our database contains the entities and relationships we used for our application: account, advertisement, buy, employee, have, items, maintains, promotes, shops, website, works_for. All SQL transactions we used will be on the following pages.

Create Customer Account:

```
INSERT INTO CUSTOMER
VALUES (Username AS U, Password, Fname, Lname, Payment_Type(OPTIONAL), Wishlist =
NULL, isAdmin = 0)
WHERE NOT EXISTS (SELECT Username
                   FROM Accounts AS A
                   WHERE U.Username <> A.Username)
```

This is used to create a new customer account.

Create Admin Account:

```
INSERT INTO CUSTOMER
VALUES (Username AS U, Password, Fname, Lname, Payment_Type(OPTIONAL), Wishlist =
NULL, isAdmin = 1)
WHERE NOT EXISTS (SELECT Username
                   FROM Accounts AS A
                   WHERE U.Username <> A.Username)
```

This is used to create a new admin account. Notice the isAdmin attribute here is 1, whereas the customer isAdmin value is 0. This distinguishes whether or not the account has admin privileges.

Delete Account:

```
DELETE FROM CUSTOMER
WHERE Username = Input.Username
```

This is used to delete an account from the database.

Modify Account:

```
UPDATE CUSTOMER
SET Password, Fname, Lname, Payment_Type
WHERE Password <> NULL, Fname <> NULL, Lname <> NULL
```

This is used when an account's attributes are being updated.

Sort:

```
SELECT *
FROM ITEMS AS I , SHOPS AS S, HAVE AS H
WHERE S.name = I.storeName, S.location = H.location, H.product_id = I.product_id
ORDER BY I.quantity OR I.price [ASC|DESC]
```

This is used when a search is being sorted by quantity or price (ascending and descending)

Search:

```
SELECT *  
FROM ITEMS AS I , SHOPS AS S, HAVE AS H  
WHERE I.name LIKE %INPUT% AND I.product_id = H.product_id, H.location = S.location
```

This is used when searching for an item.

Shops with Item

```
SELECT S.name, I.name, S.location  
FROM SHOPS AS S, ITEMS AS I, HAVE AS H  
WHERE SHOPS.ProductID = ITEM.ProductID
```

This is used when returning a shop's inventory.

Update Wishlist:

```
INSERT INTO CUSTOMER (Wishlist)  
VALUES Input.productID
```

This is used when updating the wishlist of a customer.

Add Employee:

```
INSERT INTO EMPLOYEE  
VALUES (Ssn, Fname, Lname, Address, Salary)
```

This is used to add a new employee to the database.

Delete Employee:

```
DELETE FROM EMPLOYEE  
WHERE EMPLOYEE.ssn = INPUT.ssn
```

This is used to delete an employee from the database.

Modify Employee:

```
UPDATE EMPLOYEE  
SET Fname, Lname, Address, Salary  
WHERE Fname <> NULL, Lname <> NULL, Address <> NULL, Salary <> NULL
```

This is used to update employee attributes.

Update Quantity:

UPDATE ITEMS

SET ITEMS.quantity = INPUT.quantity + ITEM.quantity

WHERE INPUT.ProductID = ITEM.ProductID

This is used when updating the quantity of an item.

New Item:

INSERT INTO ITEM

VALUES name, product_id AS PID, price, quantity

WHERE NOT EXISTS (SELECT *

FROM ITEM

WHERE PID.product_id = ITEM.product_id)

This is used to add a new item to the database.

Add Shop:

INSERT INTO SHOPS

VALUES (Address, Name)

WHERE Address <> NULL, Name <> NULL

This is used to add a new shop to the database.

Modify Shop:

UPDATE SHOPS

SET Revenue, Name, Profit, Number of Employees

This is used to modify a shop's attributes.

Advertisement:

SELECT filename

FROM ADVERTISEMENT AS A

WHERE A.projectNumber = INPUT.projectNumber

This is used to determine the ad that will be displayed on a page. Note that the input is a randomly generated number.

4.3 User Manual

Will appear at the end of the file, after the "References" section.

5 References

Images

All images should be free for noncommercial use. All rights reserved to the owners of said images.

- Green T-Shirt
<http://pngimg.com/download/5454>
- Red Socks
<https://pxhere.com/en/photo/1329167>
- Black Underwear
<https://www.flickr.com/photos/1-6-scale-doll-clothes/25151457504>
- Green Pants
https://commons.wikimedia.org/wiki/File:Cargo_pants_001.jpg
- Blue Hat
<https://www.maxpixel.net/Cap-Hat-Blue-Headgear-Felt-Hat-Womens-Hat-3286880>
- Purple Shorts
https://commons.wikimedia.org/wiki/File:Kit_shorts_concepcion95h.svg