**B.Sc. Computing / Higher Certificate**

**Second Year Project**

**Semester Four**

**Section 1:**

**User Requirements Specification**

**Leon Walker O’Brien X00126692**

**TABLE OF CONTENTS**

**Part One**

**Interview of key users**

Interview 1 3

Interview 2 4

**Results**

Evaluate Results 5

**part Two**

**User Requirements and narrative**

List of Requirements 6

System Narrative 7

**part Three**

**Use Case**

Use Case Diagram 8

Use Case Descriptions 9

**part Four**

**Use Case Model**

Class Diagram 16

**Interview 1:**

Q. What sort of items would you like the website to have?

A. I would like to see t-shirts, hoodies, patches, CDs and vinyl.

Q. What color scheme would you think would look good for a heavy metal online store?

A. I would say that darker colors would look better as they go with the music. You could also balance it out with lighter colors.

Q. As a user, what would be most important thing to you?

A. It would be for the system to be easy to use and good security I guess.

Q. Do you think a news section would be good for the website?

A. Yes! That would be great!

Q. What way would you like to interact with the system?

A. Computer/phone.

Q. What way would you like the purchasing of items to be?

A. A basket would be nice so I could keep shopping or possibly keep reading headlines.

Q. How often would you like to see the website being updated with new products and articles?

A. I’d say maybe every month.

Q. How do you think we can develop in the future?

A. Maybe move the business into custom work and keep bringing in new fresh products.

**Interview 2:**

Q. What sort of items would you like the website to have?

A. denim jackets, patches, t-shirts, leather jackets.

Q. What color scheme would you think would look good for a heavy metal online store?

A. Maybe classic black and white.

Q. As a user, what would be most important thing to you?

A. Simplistic layout.

Q. Do you think a news section would be good for the website?

A. Yes, I actually like to read the news and see what’s going on.

Q. What way would you like to interact with the system?

A. Keyboard and mouse.

Q. What way would you like the purchasing of items to be?

A. As long as it’s easy to follow, I don’t mind.

Q. How often would you like to see the website being updated with new products and articles?

A. I would like to see updates weekly so it feels fresh.

Q. How do you think we can develop in the future?

A. You could start selling clothing that relate to different music genres.

**Evaluation Of Results:**

After having a short question and answer session with a small amount of people who I thought would be perfect candidates and would visit this site, I concluded that there are a few points that must be taken into account when in development the stage. I aim to create a systemic website that pushes out the latest news and sells the best products possible relating to the music genre of heavy metal. This has been a long time passion of mine and has also been a passion of the people in the interviews just so I could get the best results.

One point made clear during the questionnaires’ was that it is important to keep the system simple and easy to use. This point has become one of my main priorities and I will definitely implement it when in development. It is a vital part of the system, as it will allow people of all ages to enjoy the experience. Older folk find the Internet daunting at the best of times so we want to make it better for them. Usability is key when it comes to the general public and it is very important to take the users opinions on board.

Aesthetics was also a big part of the interviews. Both interviewees said they would like to see dark colors on the website as it goes with the darkness and the beauty of the heavy metal genre. They also mention balancing that specific lack of color with some brightness or even the classic black and white. I completely agree with the color scheme and will try my best to make it a reality. The scheme of the colors are important as it can attract more people to the website. It just happens to be that the people who are interested in dark music also have a preference for the dark colors as they tend create an atmosphere.

Security was mentioned briefly but is probably the most important point of all. We will enforce accounts with password. Different types of user will continue to different parts of the site to ensure safety. Admins will login to a part of the site where they can edit and monitor the site and customers will go to a part of the site where they can add items to their basket and buy the items they love. Security will always play a big part as it keeps peoples information safe and keeps the integrity of the system.

**List of Requirements:**

* A home page with information containing links to a news page, a shop, registry and a login page.
* Clean layout for each clickable page.
* Dark colors to represent the darkness of the music.
* If not logged in, the news and store links can be clicked but the information will only be displayed but you can’t buy anything.
* Click register to put in information and sign in to become a customer.
* Click login to either login as admin or customer.
* Admin can delete, add or edit either news articles from the news page or products from the store. They can also monitor every user on the system.
* Customers can add products to their basket and buy items. The news page can only be viewed by customers and not changed. Customers can also view a profile page which will contain information about themselves.
* The news page has links that bring you to specific news relating to a sub genre of music. All news is the default.
* Black metal news.
* Thrash metal news.
* Death metal news.
* Power Metal news.
* The shop also has categories relating to what sort of merchandise you want.
* Patches.
* Hoodies.
* T-shirts.

* The logout link will bring you back to the login page.

**System Narrative:**

The first thing that users will see is the home page that contains information on what started this project and what it’s based on. It also goes through the heavy metal genres to educate people if needed. The last thing on the home page is contact information just in case of any inquiries. The navigation bar across the top of the website has a number of links that will take the user where they want to go. The links are store, news, register and login.

The user can click the store link and a page containing all the products in the database will load. From this point, the user cannot buy any products unless they login or register. The page will have a simple layout with each product one after the other containing information such as name, category, description and price. On the side, there will be a list of categories of products, which are clickable to show specific items in those categories. The default is all products ordered by ID numbers.

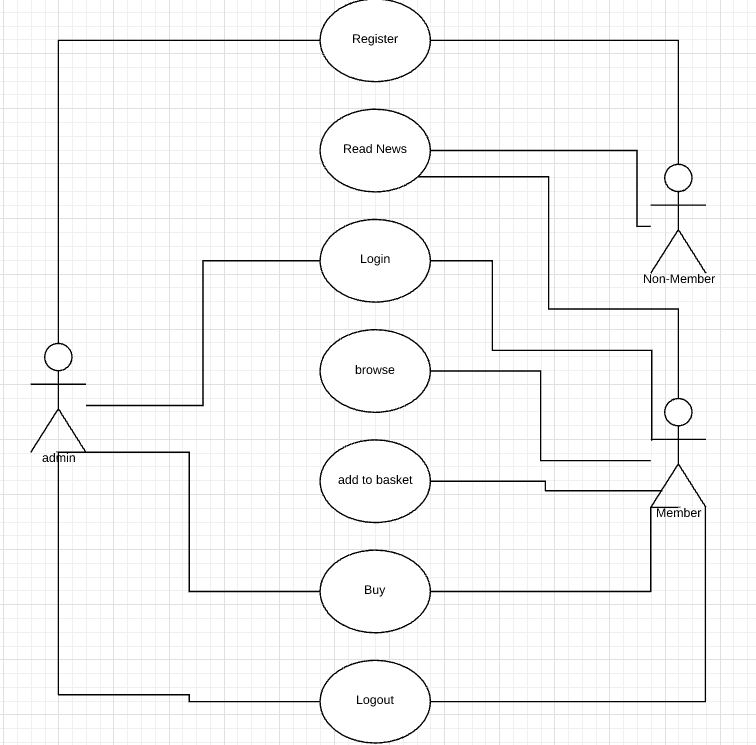
They can then move onto the news page, which is primarily made for users and customers to read only. Each article will be displayed with a heading, a genre of music, which the story relates to, and the story of what the article is about. There will also be a list on the side that allows you to narrow the articles down to a specific genre of the users liking. The default is also all articles ordered by Id numbers.

New customers can then register if they want to buy products simply by clicking the register link and this will take them to a form. They enter their personal information and submit the form. They can then login like other customers by clicking the login link and entering the correct email and password. If either the password or email is entered incorrectly, the system will tell you that it is wrong and tells the user to try again. Once logged in, the user can then start to add things to their basket and buy items.

Admins will login and be directed to a different part of the system, which will allow them to add, delete or edit either articles or products from their specific pages. It will also allow them to monitor all the users that have accounts by clicking the users link. This allows admins to see if accounts have been added to the system.

The logout link will replace the login and resister links once the user is logged in. This can be clicked when the user is done and it will return the user to the login page.

**Use Case Diagram:**



**Use Case Descriptions:**

|  |  |
| --- | --- |
| Use Case: | Register |
| Actor(s): | Admin, non-Member |
| Goal: | To successfully register the user and make them a member. |
| Overview: | User clicks the register button in the corner. A form requesting the user to put in details comes up. The user types in their details including an email and password. The user clicks submit and the form is then checked and authenticated. The user is then ready to login and buy items. |
| Pre- Condition: | The user is not a member and has not registered. |
| Post- Condition: | The user is a member and can now buy items. |
| Successful Scenario: | * User clicks register. * A form requesting details come up. * User enters their correct details. * Form is then submitted. * Authentication is done. |
| Alternative Scenario(s): | * User clicks register. * A form requesting details come up. * User enters in details. * Submit is clicked but details are wrong. * Email is already in the system. * Try another email. |

|  |  |
| --- | --- |
| Use Case: | Read News |
| Actor(s): | Non-member, Member |
| Goal: | For users to successfully read news articles |
| Overview: | User can either login or stay logged out while hitting the news link and can then read the news. Each article can be filtered so the user can find what they want by typing in the search bar. |
| Pre- Condition: | User has not read the news. |
| Post- Condition: | User has read the news. |
| Successful Scenario: | * User goes on the website. * They click the news link. * They search in the search bar for a particular article. * The article comes up. * News is now known. |
| Alternative Scenario(s): | * User goes on the website. * User logs into account. * They click on the news link. * They search in the search bar for a particular article. * The article is not there. * Continue reading others. |

|  |  |
| --- | --- |
| Use Case: | Login |
| Actor(s): | Admin, Member |
| Goal: | To successfully login a user. |
| Overview: | Members need to login to fully take advantage of all the website. Members click the login link in the corner. A form comes up requesting your email and password. The user puts their particular information in and the website gives them access to buying. If information is wrong, the website will ask the user to try again. |
| Pre- Condition: | User is not logged in. |
| Post- Condition: | User is logged in. |
| Successful Scenario: | * User clicks on the login link. * A form comes up requesting information. * They enter the correct information. * They are then allowed access. |
| Alternative Scenario(s): | * User clicks on the login link. * A form comes up requesting information. * They enter the wrong information * They are not allowed into the website * They have to try again. |

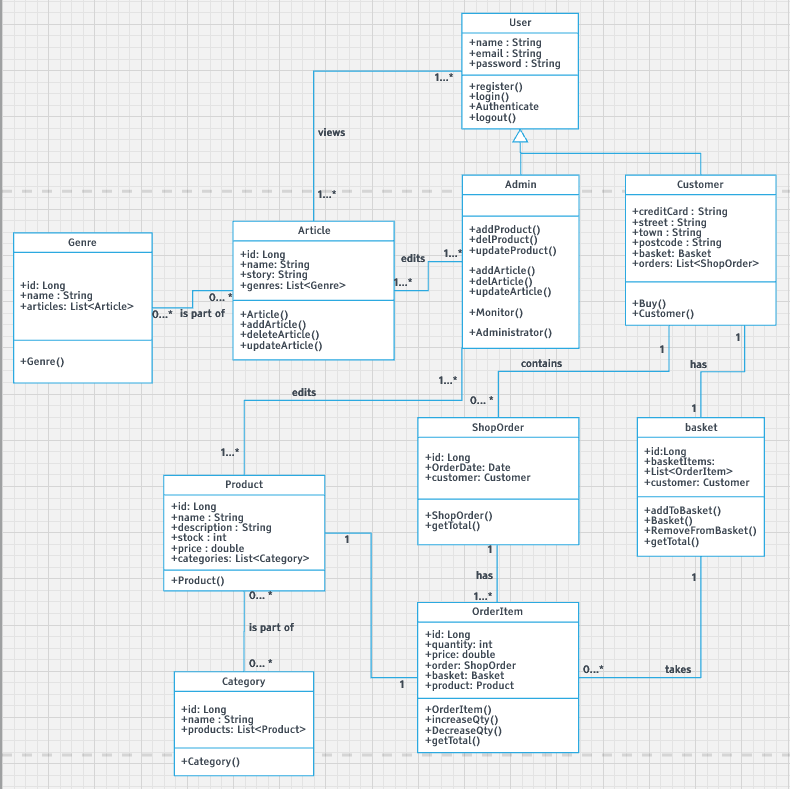
|  |  |
| --- | --- |
| Use Case: | browse |
| Actor(s): | Member |
| Goal: | For a member to successfully view all products in the store |
| Overview: | Member logs into their account. They then click the store link. They can then either scroll down manually or click the search bar, which will allow them to search for particular items. They can also use the category links along the side. |
| Pre- Condition: | Customer has not seen all products. |
| Post- Condition: | Customer has seen all products. |
| Successful Scenario: | * Member log in. * They go to the store link. * They scroll through the items * They find the item they’re seeking. |
| Alternative Scenario(s): | * Member log in. * They go to the store link. * They use the Category links on the side. * They then use the search bar. * They don’t find the item they’re seeking. |

|  |  |
| --- | --- |
| Use Case: | Add To Basket |
| Actor(s): | Member |
| Goal: | To successfully add an item to a customers basket |
| Overview: | Customers browse through the shops items and see things they like. They then press the little shopping basket icon and it will add that particular item to their basket. They can continue to do this until they’re ready to pay. |
| Pre- Condition: | Customer has no items in their basket. |
| Post- Condition: | Customer has one or more items in their basket. |
| Successful Scenario: | * User browses through items. * User spots something they want. * They press the icon. * Item is now in the basket. * Done. |
| Alternative Scenario(s): | * User browses through items. * User spots something they want. * They press the icon. * The item is now in the basket. * They continue to shop. * They add more items to the basket. |

|  |  |
| --- | --- |
| Use Case: | Buy |
| Actor(s): | Admin, Member |
| Goal: | For the Customer to successfully buy items. |
| Overview: | Customer adds things to their basket. They then click the buy link. It brings them to a order confirmation with their shipping address and the items they ordered. |
| Pre- Condition: | Customer has not bought anything. |
| Post- Condition: | Customer has bought items. |
| Successful Scenario: | * Customer clicks the buy link. * Order confirmation comes up with delivery address. * Order is o it’s way. |
| Alternative Scenario(s): | * Customer clicks the buy link. * Wrong address on order confirmation. * Contact the website company by email. * Information changed. |

|  |  |
| --- | --- |
| Use Case: | Logout |
| Actor(s): | Admin, Member |
| Goal: | To successfully logout a member. |
| Overview: | After users are done reading news, browsing and buying items, users click the logout link in the corner and this returns them to the home page where they can shut down. |
| Pre- Condition: | User is still logged in. |
| Post- Condition: | User is logged out. |
| Successful Scenario: | * User is done shopping. * User clicks the logout link * They then shut down. * Done. |
| Alternative Scenario(s): | * User is done shopping. * User clicks the logout link * They then log back in for more shopping and repeat. * They then shut down. |

**Class Diagram:**

****

**B.Sc. Computing / Higher Certificate**

**Second Year Project**

**Semester Four**

**Section 2:**

**Design Specification**

**Leon Walker O’Brien X00126692**

**TABLE OF CONTENTS**

**Part one**

**Screen Designs**

Input Screens + Output Screens 19

**part Two**

**Database Design**

ERD 23

Table/Record Layout 23

Column data types 23

Primary Keys/Foreign Keys 23

**part Three**

**Program Design**

Pseudocode 26

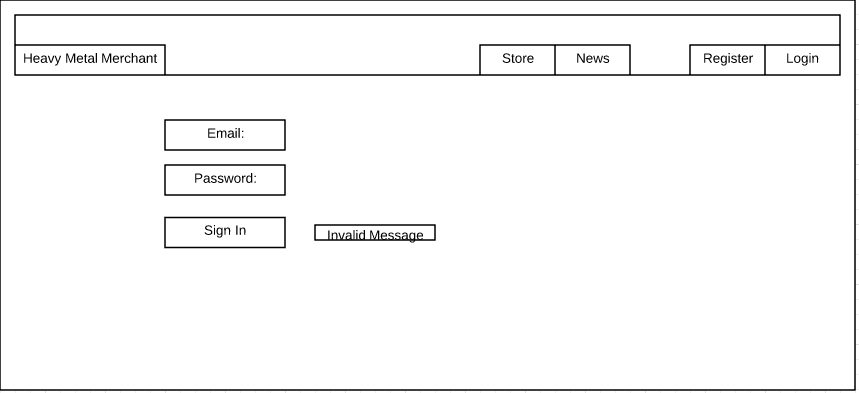
**part Four**

**Test Design**

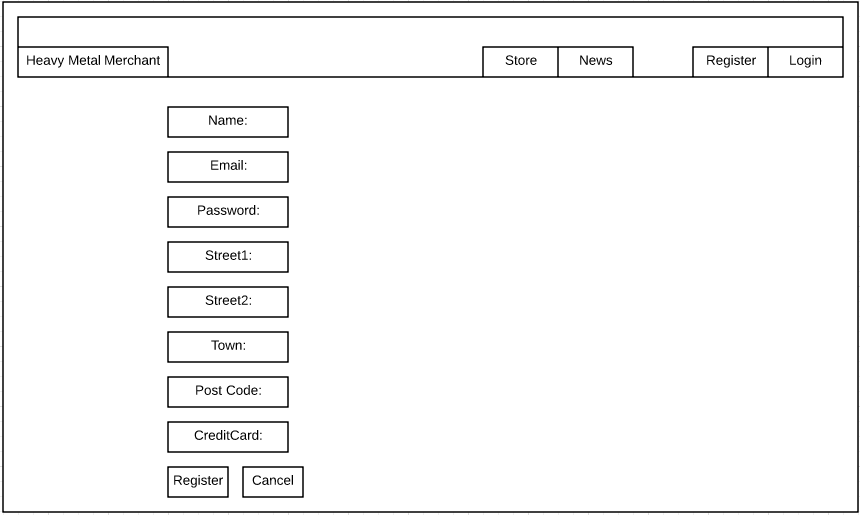
Test Case Matrix 33

**Input and Output Screens:**

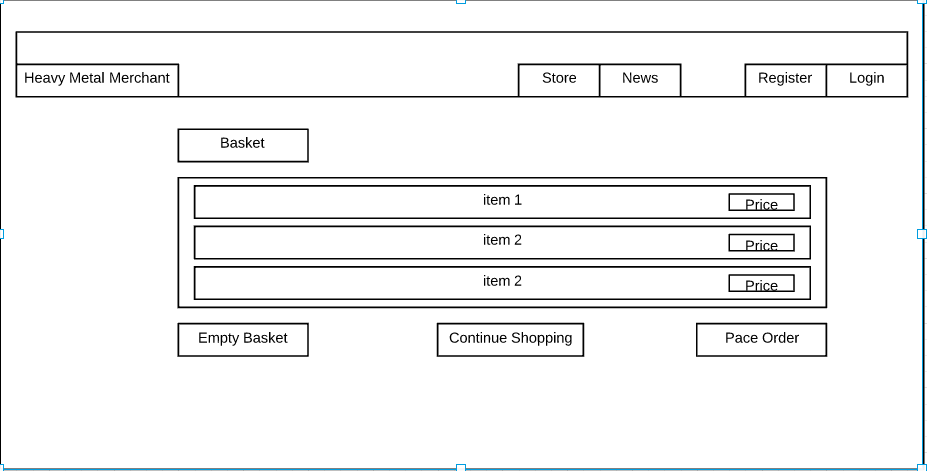
**Login screen**



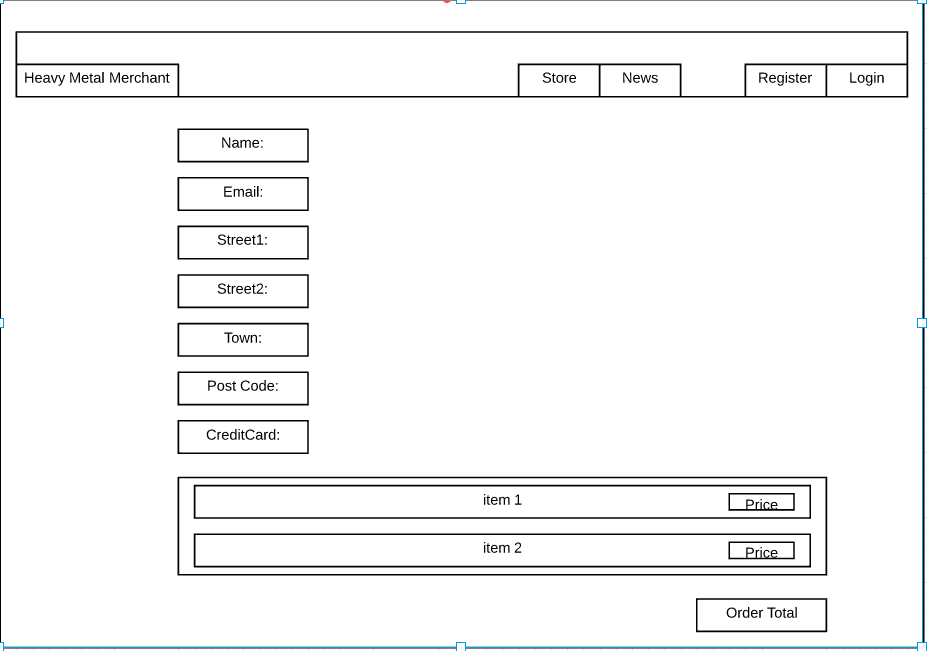
**Register Screen**



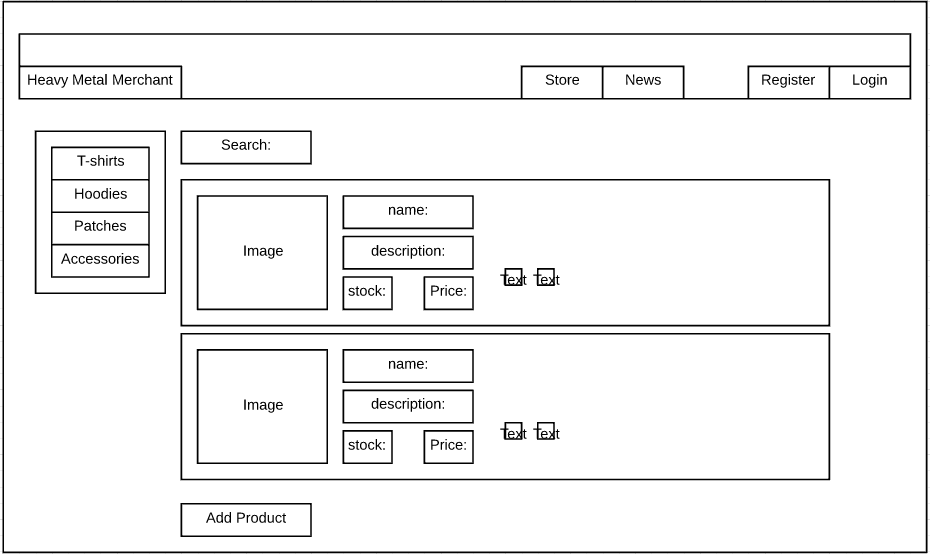
**Basket Screen**



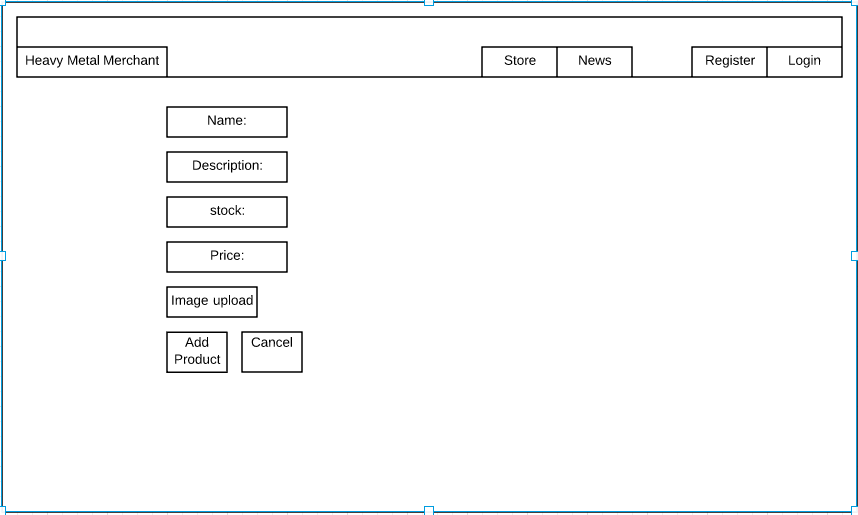
**Order Confirmed Screen**

****

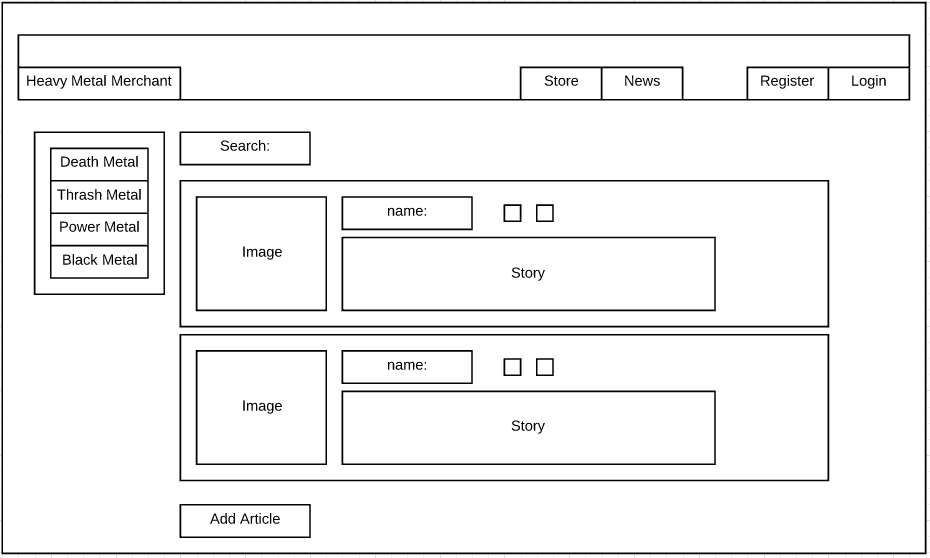
**List Product Screen**

****

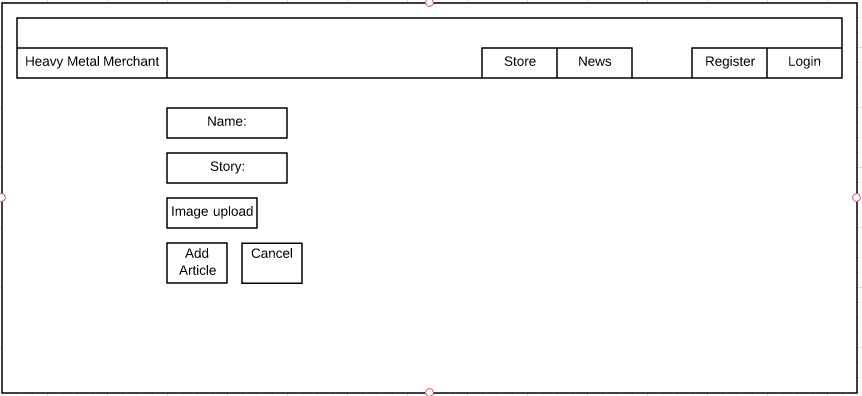
**Add Product Screen**

****

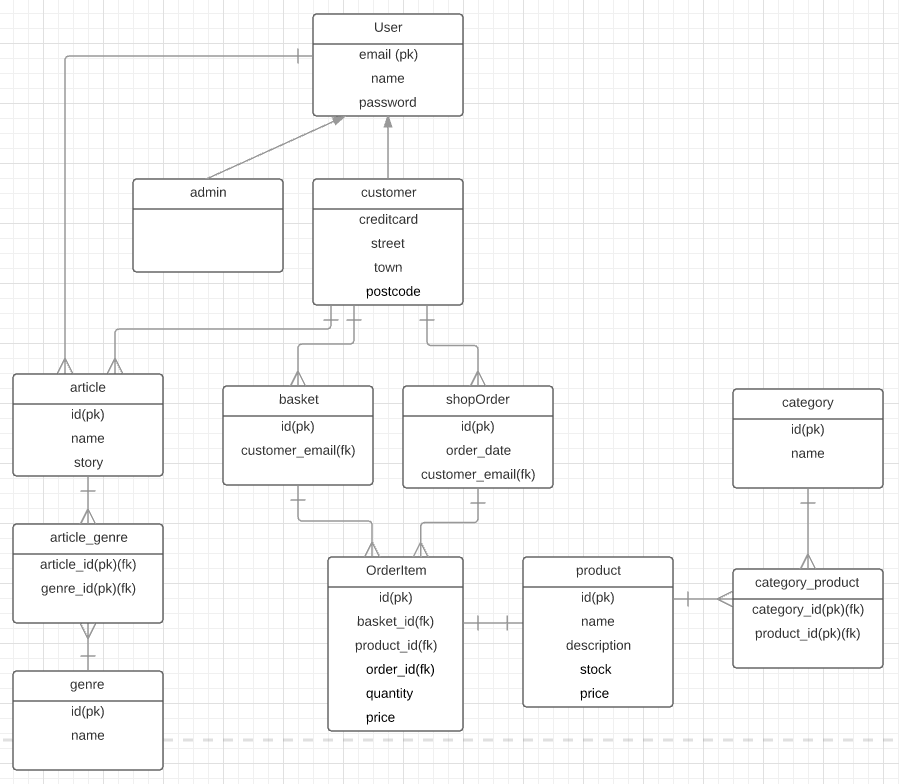
**List Article Screen**

****

**Add Article Screen**

****

**Entity Relationship Diagram:**

****

**Record Layouts/Column Data Types/Primary and Foreign keys:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **email (PK)**  VARCHAR | **Name**  VARCHAR | **password**  VARCHAR | **street1**  VARCHAR | **street2**  VARCHAR | **Town**  VARCHAR | **Postcode** VARCHAR | **creditCard**  VARCHAR |

**User**

|  |  |
| --- | --- |
| **ID(PK)**  number | **Customer\_email (FK)**  VARCHAR |

**basket**

|  |  |  |
| --- | --- | --- |
| **ID(PK)**  number | **order\_date**  date | **Customer\_email(FK)**  VARCHAR |

**shopOrder**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID(PK)**  number | **Basket\_ID(FK)**  number | **Product\_ID(FK)**  number | **Order\_ID(FK)**  number | **Quantity**  number | **Price**  decimal |

**OrderItem**

**Product**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID(PK)**  number | **Name**  VARCHAR | **Description**  VARCHAR | **Stock**  number | **Price**  decimal |

**Category\_product**

|  |  |
| --- | --- |
| **Category\_ID (PK)(FK)**  number | **Product\_ID (PK)(FK)**  number |

**Category**

|  |  |
| --- | --- |
| **ID (PK)**  number | **Name**  VARCHAR |

**Article**

|  |  |  |
| --- | --- | --- |
| **ID (PK)**  number | **Name**  VARCHAR | **Story**  VARCHAR |

**Article\_genre**

|  |  |
| --- | --- |
| **Article\_ID (PK)(FK)**  number | **Genre\_ID (PK)(FK)**  number |

**Genre**

|  |  |
| --- | --- |
| **ID (PK)**  number | **Name**  VARCHAR |

**Pseudocode:**

**Register**

**Variables:**

name: String = “John”;

email: String = “email”;

password : String = “password”;

street1: String = “123 street”;

street2: String = “Yada”;

town: String = “Dublin”;

postCode: String = “Dublin 17”;

creditCard: String = “1234”

nameIn: String;

emailIn: String;

passwordIn : String;

emailIn: String;

passwordIn : String;

street1In: String;

street2In: String;

townIn: String;

postCodeIn: String;

creditCardIn: String;

**Begin**

1. User enters name, email, password, street1, street2, town, post code and credit card into assigned slot on the form.
2. The user then clicks register and if all data types match they are then registered

The user can then login.

3. If data types do not match, they will be asked to type the correct information.

**End.**

**Login**

**Variables:**

email: String = “email”;

password : String = “password”;

emailIn: String;

passwordIn : String;

**Begin**

1. User enters emailIn and passwordIn into the login form.
2. If (emailIn = email and passwordIn = password)

The user gets logged into the system;

1. Else if(emailIn != email or passwordIn != password)

Display(“Invalid username or password, please try again”);

1. User will re-enter emailIn and passwordIn;

**End.**

**Add To Basket**

**Variables:**

Product p = new Product();

Customer customer = CurrentLoggedIn().getEmail(“email”);

**Begin**

If(customer.Basket==null){

1. sets the customers basket to new if there’s nothing in it.

Customer.setBasket(newBasket());

}

1. Customer then adds a product to their basket.

Customer.getBasket().add product(p);

Customer.update();

It then renders the basket page to view items!

**End.**

**addOne**

**variables:**

long itemID;

OrderItem item = OrderItem(itemID);

**Begin**

Item.increaseQty();

1. This increases the item count by 1.

Item.update();

1. This updates the basket.

This then redirects you to the basket page.

**End.**

**removeOne**

**variables:**

long itemID;

OrderItem item = OrderItem(itemID);

Customer c;

**Begin**

c.getBasket().removeItem(item);

1. This decreases the item count by 1.

c.getBasket.update();

1. This updates the basket.

This then redirects you to the basket page.

**End.**

**EmptyBasket**

**Variables:**

Customer c;

shopOrder order = new shopOrder();

**Begin**

1. Use method to remove all items.

c.getBasket.removeAll();

1. update.

c.getBasket.update();

You are then returned to the empty basket.

**End.**

**PlaceOrder**

**Variables:**

Customer c;

shopOrder order = new shopOrder();

**Begin**

1. map customer to this order

order.setCustomer(c);

1. copy basket to order.

Order.setItems(c.getBasket().getBasketItems());

3. Save order and print.

Order.save();

For(OrderItem 1: order.getItems())

setOrder, setBasket and update inside loop!

Order.update();

1. clear basket by setting to null.

c.getBasket().setBasketItems(null);

you are then redirected to the order confirmation!

**End.**

**AddProduct**

**Variables:**

Form<Product> addProductForm;

**Begin**

1. AddProductForm is loaded.
2. Admin inputs name, description, stock and price into slots and image is selected.
3. AddProductSubmit is then called.
4. If(authentification==true) then product is added.

You are then redirected to the listProducts page

**End.**

**updateProduct**

**Variables:**

String saveImg

Form<Product> updateProductForm;

**Begin**

1. UpdateProductForm is loaded.
2. Admin inputs whatever information needed.
3. UpdateProductSubmit is then called.
4. If(authentification==true) then product is updated.

You are then redirected to the listProducts page

**End.**

**DelProduct**

**Variables:**

Long id;

**Begin**

1. Reference the id of the product to delete.

Product.find.ref(id).delete();

Output(“Success”);

You are then redirected to the listProducts page

**End.**

**AddArticle**

**Variables:**

Form<Article > addArticleForm;

**Begin**

1. AddArticleForm is loaded.
2. Admin inputs name and story into slots and image is selected.
3. AddArticleSubmit is then called.
4. If(authentification==true) then Article is added.

You are then redirected to the listArticles page

**End.**

**updateArticle**

**Variables:**

String saveImg

Form<Article > updateArticleForm;

**Begin**

1. UpdateArticleForm is loaded.
2. Admin inputs whatever information needed.
3. UpdateArticleSubmit is then called.
4. If(authentification==true) then Article is updated.

You are then redirected to the listArticles page

**End.**

**DelArticle**

**Variables:**

Long id;

**Begin**

1. Reference the id of the product to delete.

Article.find.ref(id).delete();

Output(“Success”);

You are then redirected to the listArticles page

**End.**

**Test Case Matrix:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case Test Case No** | **Test Case Name** | **Input** | **Expected Outcome** | **Actual Outcome** | **Result (Pass/Fail)** |
| 1 | Login | Valid Email and Password entered | Progress to Store | Progress to Store | Pass |
| 2 | Login | Valid Email and Invalid password entered | Invalid login error message displayed | Invalid login error message displayed | Pass |
| 3 | Login | Invalid Email and valid password entered | Invalid login error message displayed | Invalid login error message displayed | Pass |
| 4 | Login | Invalid Email and invalid password entered | Invalid login error message displayed | Invalid login error message displayed | Pass |
| 5 | Register | Valid inputs entered | Customer is register into the system and can login | Customer is register into the system and can login | Pass |
| 6 | Register | Invalid inputs entered | Customer is asked to enter again | Customer is asked to enter again | Pass |
| 7 | Add To Basket | Customer clicks shopping basket icon | Specific item is copied to their basket | Specific item is copied to their basket | Pass |
| 8 | Add One | Customer clicks plus icon to increase the item by one | Item is increased by one | Item is increased by one | Pass |
| 9 | Remove One | Customer clicks minus icon to decrease the item by one | Item is decreased by one | Item is decreased by one | Pass |
| 10 | Remove all | Remove all link is clicked and all items are removed from the basket | Items are removed | Items are removed | Pass |
| 11 | Place Order | Place order is clicked on the basket page | Order is bought and a Order confirmation is received | Order is bought and a Order confirmation is received | Pass |
| 12 | Product | Specific Product is searched for in the search bar | Product is found | Product is found | Pass |
| 13 | Article | Specific Article is searched for in the search bar | Article is found | Article is found | Pass |