Use Case Diagrams

Guest Activity

Guest

Staff

Staff

Use Case Descriptions

|  |  |
| --- | --- |
| Use Case 1 | **Log In** |
| Actor Action | System Response |
| **1.** User enters a username and password then clicks *Log In* | **2.** The system will check if these details are valid or not. If they’re valid they will be brought to a home screen. |
|  | Alternative System response |
|  | The details are invalid, an error message will appear notifying the user that they have entered invalid details. |

**Non-Functional Requirement**

Security: All user data should be encrypted with a hashed password so that their details cannot be accessed by malicious third parties.

|  |  |
| --- | --- |
| Use Case 2 | **Log Out** |
| Actor Action | System Response |
| **1.** User clicks button with the text “Log Out”. | **2.** When clicked the user will be logged out of their account and returned to the home screen as a guest. |

**Non-Functional Requirement**

Usability: Users can be logged out with one click no matter what the page.

Security: There must be no trace of the user after they log out.

|  |  |
| --- | --- |
| Use Case 3 | **Cancel Order** |
| Actor Action | System Response |
| **1.** Customer clicks *Cancel Order*. | **2.** System asks *Are you sure?* |
| **3.** Click *Yes* | **4.** The order will be deleted from the customer’s list of current orders. The money that was taken out the customer’s account will be added back to the account. |

**Non-Functional Requirement**

Security: System must use trusted provider to transfer money into customers account.

**Pre-Condition**

Customer must have at least one unconfirmed order

|  |  |
| --- | --- |
| Use Case 4 | **Check Orders** |
| Actor Action | System Response |
| **1.** Customer clicks *Orders*. | **2.** System displays separated list of pending and confirmed orders. |
|  | Alternative System response |
|  | The Customer has no orders associated with them |

**Non-Functional Requirement**

Performance: System should retrieve all orders up to date.

|  |  |
| --- | --- |
| Use Case 6 | **Confirm Order** |
| Actor Action | System Response |
| **1.** Customer clicks *Confirm Order* on a specific Order. | **2.** Displays a confirmation message *You have received this product?* and displays the corresponding product name. |
| **3.** Customer clicks *Yes.* | **4.** The Customer is returned to the Orders page with the order now in the confirmed orders. |

**Non-Functional Requirement**

Performance: System should retrieve and display these orders within 5 seconds of the button being clicked.

|  |  |
| --- | --- |
| Use Case 7 | **Report problem with Order** |
| Actor Action | System Response |
| **1.** Customer clicks *Report* on a specific order. | **2.** Displays a text box titled *Problem.* |
| **3.** Customer fills in textbox with details of what was wrong with the order. Customer clicks *send* to finish. | **4.** The problem and the associated product and customer details are stored for a member of staff to review later. |

|  |  |
| --- | --- |
| Use Case 8 | **Search for Parts** |
| Actor Action | System Response |
| **1.** User clicks *Search*. | **2.** A search screen is displayed with fields for product details such as Name, Part, Make, Model and Price. |
| **3.** User enters as many details as they need to. They then click *Search Products.* | **4.** Any Products that match the fields searched for are displayed. |
|  | Alternative System response |
|  | There are no products matching the search and a message is displaying |

**Non-Functional Requirement**

Performance: System should retrieve and display the products within 5 seconds

Usability: There should be drop-down lists so that the user can pick which details they want in case they do not know.

|  |  |
| --- | --- |
| Use Case 8 | **Redeem Code** |
| Actor Action | System Response |
| **1.** Customer clicks *Redeem Code*. | **2.** A screen is displayed with a text box. |
| **3.** Customer fills in text box with voucher code. | **4.** The voucher code is validated and the value associated with it is added to the user’s account. |
|  | Alternative System response |
|  | The voucher code is invalid and the user is returned to the enter voucher screen. |

**Non-Functional Requirement**

|  |  |
| --- | --- |
| Use Case 9 | **Rate Product** |
| Actor Action | System Response |
| **1.** Customer clicks *Rate Product* on a specific Order. | **2.** A screen with a dropdown list of numbers 1 to 5 is displayed. |
| **3.** Customer picks a number from 1 to 5 depending on their happiness with the product. The Customer then clicks *Submit.* | **4.** The rating is stored to list of ratings for that product. |

**Non-Functional Requirement**

|  |  |
| --- | --- |
| Use Case 10 | **Create Comment** |
| Actor Action | System Response |
| **1.** Customer clicks *Comment on Product* on a specific Product. | **2.** A screen is displayed with a text box. |
| **3.** The Customer writes a comment on the product. The Customer then clicks *Submit.* | **4.** The comment is stored to list of comments for that product. |

**Non-Functional Requirement**

|  |  |
| --- | --- |
| Use Case 11 | **Create PC** |
| Actor Action | System Response |
| **1.** Customer clicks *Create PC.* | **2.** A screen is displayed with fields for all the different parts of the PC (Monitor, Processor, RAM etc.). |
| **3.** The Customer adds the product they want for each part of the PC. The Customer clicks *Finish.* | **4.** An order is placed for all the parts the Customer wanted. |

**Non-Functional Requirement**

|  |  |
| --- | --- |
| Use Case 12 | **Enter Card Details** |
| Actor Action | System Response |
| **1.** Customer clicks *Enter Details.* | **2.** A screen is displayed with fields asking for your credit card number |
| **3.** The Customer clicks *Finish.* | **4.** You’re returned to the home screen. |
|  | Alternative System response |
|  | The credit card is invalid and the user is returned to the enter voucher screen. |

**Non-Functional Requirement**

**Security:** System should implement a defence to stop any chance of data being

|  |  |
| --- | --- |
| Use Case 13 | **View Product** |
| Actor Action | System Response |
| 1. User goes into the webpage and selects what specific model they wish to view | 2. The system will then redirect the user into which model they have requested and display the items available to purchase. |

|  |
| --- |
| Alternative system response  **2a.** Should the specific item the user selects happens to be sold out. The system will display an appropriate message informing the user of the issue. |

|  |  |
| --- | --- |
| Use Case 14 | **Register as User** |
| Actor Action | System Response |
| 1. User clicks into the register tab on the page | 2. Once clicked, the system will redirect him into the registration page that will grant them access to become a user. |
| 3. User will enter credentials that the system asks them and will click enter | 4. The system will check the details that the user has provided and will send an email of confirmation to the users email address. |
| 5. The user will open the email then and finally confirms that they are a user. |  |

|  |
| --- |
| Alternative system response  **4a.** User does not receive an email of confirmation.   * User did not provide proper email address |

**Non-Functional Requirement  
Security**: Process of registration must be secured so that no personal information will be leaked onto another party. Problems like these can put the user at risk.

|  |  |
| --- | --- |
| Use Case 15 | **Add Item** |
| Actor Action | System Response |
| 1. Admin clicks into ‘Add Item’. | 2. Displays page in regards to adding a new product. |
| 3. Admin provides details of new product and then clicks ‘Finish’. | 4. System is updated and saves newly added product on webpage |

**Non-Functional Requirement**

**Scalability:** New items should be able to be added to increase the range of products available.

|  |  |
| --- | --- |
| Use Case 16 | **Edit Product details** |
| Actor Action | System Response |
| 1. Admin clicks into product and then clicks on ‘Edit’ | 2. Displays page where admin can change product information. |
| 3. Admin edits information and clicks ‘Finish’ | 4. The system is updated and the new product information is now displayed. |

**Non-Functional Requirement  
Extensibility:** Should be able to modify all attributes of product no matter what attributes they include especially if there’s new products on the market.

|  |  |
| --- | --- |
| Use Case 17 | **Apply discount** |
| Actor Action | System Response |
| 1.Admin clicks into product and then clicks into ‘Edit’ | 2. Displays page where the admin can edit product information |
| 3. Admin ticks on the box that grants a discount, enters the value of discount and clicks ‘Finish’. | 4. The system will update accordingly and display new prices with discounts applied accordingly. |

|  |  |
| --- | --- |
| Use Case 18 | **Remove Discount** |
| Actor Action | System Response |
| 1. Should the product already have a discount linked to it, go into it and click ‘Edit’ | 2. Displays page where the admin can edit product information |
| 3. Admin then un-ticks the box that grants the discount and clicks ‘Finish’. | 4. The system will update accordingly and display the correct price without discount applied. |

|  |  |
| --- | --- |
| Use Case 19 | **Remove Comment** |
| Actor Action | System Response |
| 1. Admin clicks into desired product and clicks into ‘Edit’ | 2. System will redirect to the edit page of the chosen product |
| 3. Admin scrolls down the comments, pick a specific one and clicks on ‘Delete’ | 4. System will then remove comment from page |
| 5. Admin clicks on ‘Finish’ | 6. System will update accordingly with selected comment now permanently removed |

**Non-Functional Requirement**

**Security**: Admin must be able to maintain order in the web page, makes sure that messages do not consist of inappropriate behaviour and/or spam.

**NFR:**

Usability

Maintainability

Scalability

Extensibility

Availability

Security

Portability

**REFERENCES:**

http://creately.com/diagram/example/hpczuu4u/Manufacturing%20company