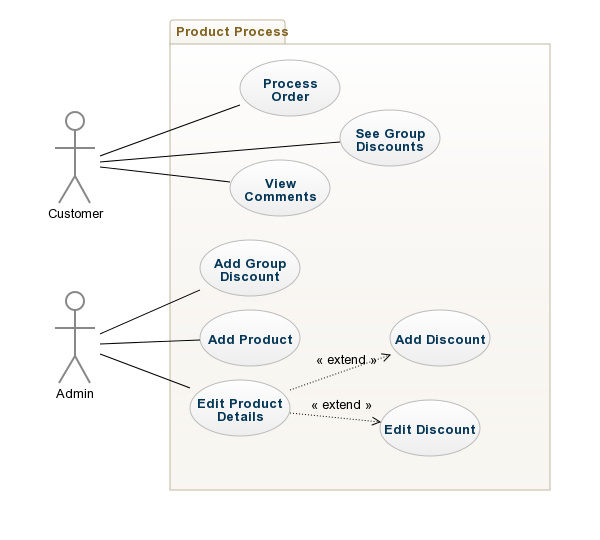
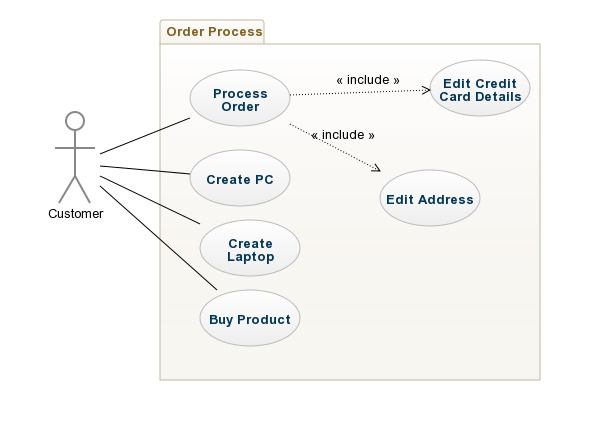
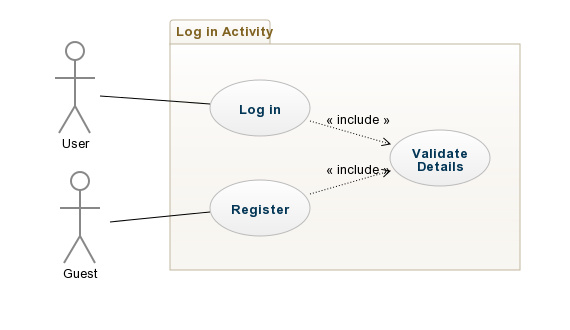
Use Case Diagrams







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 4 | **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**

**Security:** Users should only be able to make comments for bought products.

|  |  |
| --- | --- |
| Use Case 5 | **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**

**Usability:** Customer should easily be allowed to find the parts they’re looking for.

|  |  |
| --- | --- |
| Use Case 6 | **Create Laptop** |
| Actor Action | System Response |
| **1.** Customer clicks *Create Laptop.* | **2.** A screen is displayed with fields for all the different parts of the Laptop (Processor, RAM etc.). |
| **3.** The Customer adds the product they want for each part of the Laptop. The Customer clicks *Finish.* | **4.** An order is placed for all the parts the Customer wanted. |

**Non-Functional Requirement**

**Usability:** Customer should easily be allowed to find the parts they’re looking for.

|  |  |
| --- | --- |
| Use Case 7 | **Buy Product** |
| Actor Action | System Response |
| **1.** Customer selects Product and clikcs *Buy.* | **2.** It is added to the Customer’s cart. |

**Non-Functional Requirement**

**Usability:** Customer should easily be allowed to find the product they’re looking for.

|  |  |
| --- | --- |
| Use Case 8 | **Edit Card Details** |
| Actor Action | System Response |
| **1.** Customer is submitting an order. | **2.** A screen is displayed with fields asking for your credit card number |
| **3.** The Customer enters details *Proceed.* | **4.** You’re taken to the next screen |
|  | Alternative System response |
|  | The credit card is invalid the user is not allowed to proceed. |

**Non-Functional Requirement**

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

|  |  |
| --- | --- |
| Use Case 9 | **Edit Address Details** |
| Actor Action | System Response |
| **1.** Customer is submitting an order. | **2.** A screen is displayed with fields asking for your address. |
| **3.** The Customer enters details *Proceed.* | **4.** You’re taken to the next screen |

**Non-Functional Requirement**

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

|  |  |
| --- | --- |
| Use Case 10 | **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 11 | **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 12 | **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. |

|  |  |
| --- | --- |
| Use Case 13 | **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 clicks the product and enters the value of discount and clicks ‘Finish’. | 4. The system will update accordingly and display new prices with discounts applied accordingly. |

|  |  |
| --- | --- |
| Use Case 14 | **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 clicks the product and clicks *Remove*. | 4. The system will update accordingly and display the correct price without discount applied. |

|  |  |
| --- | --- |
| Use Case 15 | **Add Group Discount** |
| Actor Action | System Response |
| 1. Click *Add Group Discount.* | 2. Displays page where the admin can add Discount. |
| 3. Admin then selects their products and enters there appropriate Discount. | 4. The system will update accordingly and add the new Group Discount to the database. |

|  |  |
| --- | --- |
| Use Case 16 | **See Group Discounts** |
| Actor Action | System Response |
| 1. Customer clicks *View Group Discount.* | 2. Displays page where the all Group Discounts are displayed. |

**Non-Functional Requirement**

**Performance:** Should find all Group Discounts within 5 seconds.

|  |  |
| --- | --- |
| Use Case 17 | **View Comments** |
| Actor Action | System Response |
| 1. Customer selects Product and clicks *View Comments.* | 2. Displays page where the all Comments for selected product are displayed. |

**Non-Functional Requirement**

**Performance:** Should find all Comments within 5 seconds.

**NFR:**

Usability

Maintainability

Scalability

Extensibility

Availability

Security

Portability

**REFERENCES:**

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