****

**Future University in Egypt**

**Faculty of Computers and Information Technology**

**Department of Computer Science**

CSC 251 (Software Engineering \_1)

**Project Title:**

**Pharmacy system**

**Lab Day/Time:**

**Tuesday /9-11**

|  |  |  |
| --- | --- | --- |
| Prepared by | | |
| Name | **ID** |
| Dina Ayman | 20182996 |
| Joyce Mikha | 20183808 |
| Karim Tarek | 20180208 |
| Mariam Fathy | 20184294 |
| Yassmin Yasser | 20183018 |
| Remon William | 20182074 |

**Supervised by**

**Prof. Ramadan Moawad / Hisham Ahmed**

**TA. Amr Mansour**

|  |
| --- |
|  |

1. **Add product:**

Exist

User’s input

Exist

Request a search query

Product’s Page updated with new product

View

Add product:

Model

Product Database

Controller

Controller will take the input and check if it exists or not

ADD

**Description: -**

1. First, in view the screen will view the interface of the page that the user will use it to enter the name of new product.
2. The controller will take the input and will go to model to request it to retrieve the data of the products from the database of the product and check if the name of the new product is existing or not.
3. The model will response with exist or not to controller.
4. If it responds with exist, the controller will be said to view to appear to user that the product is already exists.
5. Else if the product doesn’t exist, it will continue the process of the product addition.
6. Finally, the new product will be added to the list of products that will appear to the user.
7. **Generate reports:**

Error

Sends action from user

Error

request for data

Display data

View

User clicks view reports

Model

System Database

Controller

Receives order from view then get data from model

view

sales

profit

salary

**Description: -**

1. The view interface will display more than one type of reports (for example: sales report, salary report, profit report) and the view button as well so first you have to choose the type of report you want then click view.
2. The view interface will send the user event to the controller.
3. The Controller will Receives order from view and request for data from model
4. The model has two option either not to find the data and return timeout error to the view
5. Or to find the data as for the sales report, the model will search for number of units sold and the unit price and multiply them.

For the salary report, the model will search for the salary, the working hours.

For profit report, the model will calculate total revenues – total expenses.

1. Finally, the model will display the required report to the view.
2. **Add patient**

Found

User’s Event

Found

Request a search query

By phone no.

Display patient’s page with new patient data

View

Model

Patient Database

Controller

Controller will take the response depending on the event that the user pressed

**Add Patient**

Name:

Address:

PhoneNo:

v

PatientHistory:

v

ADD

**Description: -**

1. Firstly, we have a view window that the pharmacist or administrator will use it to apply the application of new patient.
2. Secondly, Controller will take the event (add) from the user.
3. Then, controller will take the phone No. Of the user and go to patient database and check if it exists or not.
4. If it found patient with the same phone No, pharmacist or admin would got an error.
5. Else if this phone numbers are new; he will be redirected to the patient’s page.
6. Finally, he will be to see the new patient that he added in System.
7. **Patient order:**

User’s Event

Request a search query

About medicine to decriment the order’s quantity.

Receipt updated with the total price of products

View

Model

Product Database

Controller

Controller will take the info of the order and will start to make a search query about the medicine to decrement the quantity of the order from the quantity that stored in inventory

**Order product**

Name:

quantity:

Order

Panadol

2

**Description: -**

1. Firstly, we have a view window that the pharmacist will use it to make the order of the patient this window consist of 2 dropdown lists (the name of the medicine and its quantity).
2. The quantity dropdown list is automatically updated by the quantity in the inventory.
3. Secondly, Controller will take the event (order) from the user.
4. Then, the Controller will take the info of the order and will start to make a search query about the medicine.
5. the model decrements the quantity of the order from the quantity that stored in inventory.
6. Finally, the receipt is printed by the name of the medicine, quantity and the price.

**Diagram, engineering drawing

Description automatically generated**

**Diagram

Description automatically generated**

**Unit** **Test** **Case1: decrementproduct(product** **pr,quantity)**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Input | | | | | | | Expected Output | Actual Output | Test Case Result |
| Product pr | | | | | | quantity |
| actualName | formula | code | price | quantity | expirationDate |
| panadol | b32 | 777 | 50 | 10 | 29/11/2020 | 10 | 0 | 0 | Success |
| comtrex | b442 | 737 | 10 | 50 | 29/11/2020 | 60 | Insufficient amount in storage | Insufficient amount in storage | Success |
| panadol | b32 | 777 | 50 | 10 | 29/11/2020 | ten | 0 | Error | Fail |

**Unit** **Test** **Case2:** **SendAlert (** **)**

|  |  |  |  |
| --- | --- | --- | --- |
| Input | Expected Output | Actual Output | Test Case Result |
|  | False | No output | Fail |
|  | true | No output | Fail |

**Unit** **Test** **Case3:** **view\_alert(** **)**

|  |  |  |  |
| --- | --- | --- | --- |
| Input | Expected Output | Actual Output | Test Case Result |
|  | False | No output | fail |
|  | your inventory level is under normal level | your inventory level is under normal level | Success |

**Unit** **Test** **Case4:** **addpatientdata (patient** **a)**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Input | | | | | | | | Expected Output | Actual Output | Test Case Result |
| a | | | | | | | |
| genericid | patienthistory | <product> | username | password | address | Phone number | name |
| 1 | “Headache” | ez | “mero” | 22 | “A512” | 012 | mariam | Patient added successfully | Patient added successfully | Success |
| 2 | “covid-19” | eez | “joya” | 11 | “co-19” | 012 | joyce | Patient added successfully | Patient added successfully | Success |
| 1 | “toothache” | ezz | “kimo” | 33 | “A6-14” | 015 | karim | Patient is already exist. | Patient is already exist. | Success |
| 2.1 | “zahymar” | ezzz | “remo” | 44 | “cr-7” | 012 | remon | Patient added successfully | Error | Fail |

**Unit** **Test** **Case5:** **addproduct(product** **a)**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Input | | | | | | Expected Output | Actual Output | Test Case Result |
| Product a | | | | | |
| actualName | formula | code | price | quantity | exirationDate |
| “Comtrex” | “h20+co2” | “cr” | 20 | 10 | 31 | "product was added successfully" | "product was added successfully" | Success |
| “Folmux” | “h20+co2” | “cr” | 20 | 10 | 31 | "product was added successfully" | “Product was already exist” | Fail |
| “Comtrex” | “h2o” | “cr” | 20 | 10 | 21 | "product was added successfully" | “Product was already exist” | Fail |
| 123 | “zz” | “gg” | 20 | 10 | 21 | "product was added successfully" | Error | Fail |

**Unit** **Test** **Case6:** **searchproduct(product** **a)**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Input | | | | | | Expected Output | Actual Output | Test Case Result |
| Product a | | | | | |
| actualName | formula | code | price | quantity | expirationDate |
| “Comtrex” | “h20+co2” | “cr” | 20 | 10 | 31 | Product Was Found | Product Was Found | Success |
| “Folmux” | “h20+co2” | “cr” | 20 | 10 | 31 | Product Was Found | Product Not Found | Fail |
| “Comtrex” | “h2o” | “cr” | 20 | 10 | 21 | Product Was Found | Product Not Found | Fail |
| “123” | “h20+co2” | “cr” | 20 | 10 | 21 | Product Was Found | Product Not Found | Fail |

**Unit** **Test** **Case7:** **deleteproduct(product** **a)**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Input | | | | | | Expected Output | Actual Output | Test Case Result |
| Product a | | | | | |
| actualName | formula | code | price | quantity | expirationDate |
| “Comtrex” | “h20+co2” | “cr” | 20 | 10 | 31 | product was deleted successfully | product was deleted successfully | Success |
| "cataflam" | "gg" | "cf" | 20 | 20 | 31 | product was deleted successfully | product was deleted successfully | Success |
| “123” | “ooo” | “fff” | 20 | 10 | 21 | "product not found" | "product not found" | Success |
| “123” | “h20+co2” | “cr” | 20 | 10 | 21 | product was deleted successfully | “Product Not Found” | Fail |

**Unit** **Test** **Case8:** **add\_staff\_data(staff** **s)**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Input | | | | | | | | Expected Output | Actual Output | Test Case Result |
| s | | | | | | | |
| username | password | address | phonenumber | name | W\_hours | salary | id |  |  |  |
| “mariam” | “123” | “Heliopolis” | “012” | “mariam” | 1 | 20 | 3 | " worker data is added successfully" | "worker data is added successfully" | Success |
| “karim” | “123” | “Heliopolis” | “0122” | “karim” | 3 | 20 | 1 | " worker data is added successfully" | " worker data is added successfully" | Success |
| “dina” | “122” | “maadi” | “01220” | “dina” | 2 | 40 | 1 | "this worker is already exist" | "this worker is already exist" | success |
| “yassmin” | “124” | “nasr city” | “012204” | “yassmin” | 1 | 20 | two | " worker data is added successfully" | Error | Fail |

**Unit** **Test** **Case9:** **addproducttoorder (product** **a)**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Input | | | | | | Expected Output | Actual Output | Test Case Result |
| a | | | | | |
| actualName | formula | code | price | quantity | expirationDate |
| “Comtrex” | “h20+co2” | “cr” | 20 | 10 | 31 | product was added to your order | product was added to your order | Success |
| "cataflam" | "zx" | "cfd" | 20 | 20 | 31 | product was added to your order | Product doesn’t exist | Fail |

**Unit** **Test** **Case10:** **totalprice (product** **a)**

|  |  |  |  |
| --- | --- | --- | --- |
| Input | Expected Output | Actual Output | Test Case Result |
|  | "Total price is calculated is =200” | "Total price is calculated is =200” | success |
|  | "Total price is calculated is =300” | "Total price is calculated is =300” | success |
|  | "Total price is calculated is =500” | Error | Fail |

**Unit** **Test** **Case11:** **pay(paymentmethod** **payment)**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Input | | | | | | | Expected Output | Actual Output | Test Case Result |
| Paymentmethod payment | | | | | | |
| choice | name | cardnumber | cvv | expirydate | email | password |
| 1 | “mariam” | “99999” | “220” | 29/12/2 020 | “s20@gmail.com ” | “2222” | paid by credit card successful | You paid by credit card successful | Success |
| 2 | “mariam” | “99999” | “220” | 29/11/2020 | “s20@gmail.com ” | “2222” | You paid by PayPal successful | You paid by PayPal successful | Success |
| 3 | “dina” | “33333” | “221” | 29/3/  2020 | “r0@gmail.com” | “4444” | Wrong choice | Wrong choice | success |
| 2 | “yasmin” | “44444” | “222” | 20/11/2 020 | y0@gmail.com | 4555 | You paid by PayPal successful | Error | Fail |

**Unit** **Test** **Case12:view\_report(String** **sname)**

|  |  |  |  |
| --- | --- | --- | --- |
| Input | Expected Output | Actual Output | Test Case Result |
| sname |
| profit | 190 | 190 | Success |
| sales | 3000 | 3000 | Success |
| salary | 1200 | 1200 | Success |
| easy | Not found option | Not found option | Success |

**Unit** **Test** **Case13:update(String** **name,int** **expenses,int** **revenue,int** **w\_hour,int** **salaryperhour,int** **unitnumber,int** **price)**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Input | | | | | | | Expected Output | Actual Output | Test Case Result |
| name | expenses | revenues | workinghours | Salary\_per\_ho ur | units\_number | price |
| profit | 100 | 1000 | 65 | 6 | 15 | 30 | 900 | 900 | Success |
| sales | 100 | 1000 | 65 | 6 | 15 | 30 | 450 | 450 | Success |
| salary | 100 | 1000 | 65 | 6 | 15 | 30 | 390 | 390 | Success |
| price | 100 | 1000 | 65 | 6 | 15 | 30 | Not Valid | Not found option | Fail |
| salary | 100.0 | 1000 | 65 | 6 | 15 | 30 | 390 | Error | Fail |

***Unit*** ***Test*** ***Case14:*** ***incrementproduct(product*** ***a)***

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Input | | | | | | Expected Output | Actual Output | Test Case Result |
| a | | | | | |  | | |
| actualName | formula | code | price | quantity | expirationDate |
| comtrex | h20+co2 | cr | 20 | 10 | 31 | quantity is =20 product was Incremented Successfully | quantity is =20 product was Incremented Successfully | Success |
| Folmux | h20+co2 | cr | 20 | 30 | 31 | quantity is =40 product was Incremented Successfully | Product Not Found | Fail |
| Comtrex | h2o | cr | 20 | 40 | 21 | quantity is =50 product was Incremented Successfully | Product Not Found | Fail |
| Comtrex | h20+co2 | cr | 20 | 80 | 21 | quantity is =100  product was Incremented Successfully | quantity is =90 product was Incremented Successfully | Success |
| Comtrex | h2o+co2 | cr | twenty | 10 | 31 | quantity is =20 product was Incremented Successfully | Product Not Found | Fail |
| Comtrex | h2o+co2 | cr | 20 | 10.9 | 31 | quantity is =20.9 product was Incremented Successfully | quantity is =20 product was Incremented Successfully | Fail |

**Unit Test Case15: getproduct()**

|  |  |  |  |
| --- | --- | --- | --- |
| Input | Expected Output | Actual Output | Test Case Result |
|  | Products will be displayed | Panadol b32 777 2  profen bh32 997 2  comtrex h20+co2 cr 10  cataflam gg cf 20 | Success |

**Unit** **Test** **Case16:calculate\_salary(String** **sname,int** **expenses,int** **revenues,int** **workinghours,int** **salary\_per\_hour,int** **units\_number,int** **price)**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Input | | | | | | | Expected Output | Actual Output | Test Case Result |
|  | | | | | | |
| sname | expenses | revenues | working hours | salary\_per \_hour | units\_number | price |
| salary | 10 | 200 | 30 | 40 | 50 | 60 | salary report = 1200.0 | salary report = 1200.0 | Success |
| sala | 10 | 200 | 30 | 40 | 50 | 60 | salary report = 1200.0 | not Found option | fail |

**Unit** **Test** **Case18:calculate\_profit(String** **sname,int** **expenses,int** **revenues,int** **workinghours,int** **salary\_per\_hour,int** **units\_number,int** **price)**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Input | | | | | | | Expected Output | Actual Output | Test Case Result |
|  | | | | | | |
| sname | expenses | revenues | working hours | salary\_per \_hour | units\_number | price |
| profit | 10 | 200 | 30 | 40 | 50 | 60 | profit report = 190.0 | profit report = 190.0 | Success |
| sala | 10 | 200 | 30 | 40 | 50 | 60 | profit report = 190.0 | not Found option | fail |

**Unit** **Test** **Case17:calculate\_sales(String** **sname,int** **expenses,int** **revenues,int** **workinghours,int** **salary\_per\_hour,int** **units\_number,int** **price)**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Input | | | | | | | Expected Output | Actual Output | Test Case Result |
|  | | | | | | |
| sname | expenses | revenues | working hours | salary\_per \_hour | units\_number | price |
| sales | 10 | 200 | 30 | 40 | 50 | 60 | sales report = 3000.0 | sales report = 3000.0 | Success |
| sala | 10 | 200 | 30 | 40 | 50 | 60 | sales report = 3000.0 | not Found option | fail |

**System Testing**

**Project Name: Pharmacy System**

**Add product Test Case**

**Test Case ID:** 1.1 **Test Designed By:**

**Test Priority (Low/medium/Large):** Med **Test Designed Date:**

**Module Name:** add product module **Test Executed By:**

**Test Title:** product added successfully **Test Execution Date:**

**Pre-Conditions:** login to the system

**Dependencies:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Step | Test Steps | Test Data | Expected result | Actual result | Status (Pass/Fail) | Notes |
| 1 | Navigate to product page | Press on product button | pharmacist must navigate to products page | pharmacist navigated to products page | pass |  |
| 2 | Pharmacist enter new product data that wasn’t in the system before | Pharmacist enter actual name of the product, formula, code, price, quantity and expiration date then click on add button | Message: this product is added successfully | Product was added successfully | Pass |  |
| 3 | Pharmacist enter new product data that was in the system | Pharmacist enter actual name of the product, formula, code, price, quantity and expiration date then click on add button | Message:  product was already exists | product was already exists | pass |  |

**Search Product Test Case**

**Test Case ID:** 1.2 **Test Designed By:**

**Test Priority (Low/medium/Large):** Med **Test Designed Date:**

**Module Name:** search product module **Test Executed By:**

**Test Title:** product found successfully **Test Execution Date:**

**Pre-Conditions:** product must be added to the system

**Dependencies:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Step | Test Step | Test Data | Expected Result | Actual Result | Status(Pass/Fail) | Notes |
| 1 | Navigate To product page | Press on product button | pharmacist must navigate to products page | pharmacist navigated to products page | pass |  |
| 2 | Search on a product that in the system | Pharmacist enter product code, name and formula in text area and click on search button | Pharmacist will find the product | Pharmacist found the product | pass |  |
| 3 | Search on a product that not in the system | Pharmacist enter product code and formula in text area and click on search button | Pharmacist will not find the product | Product not found | pass |  |

**Delete Product Test Case**

**Test Case ID:** 1.3 **Test Designed By:**

**Test Priority (Low/medium/Large):** Med **Test Designed Date:**

**Module Name:** delete product module **Test Executed By:**

**Test Title:** product deleted successfully **Test Execution Date:**

**Pre-Conditions:** product must be found in the system

**Dependencies:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Step | Test Steps | Test Data | Expected Result | Actual Result | Status(pass/Fail) | Notes |
| 1 | Navigate To product page | Press on product button | pharmacist must navigate to products page | pharmacist navigated to products page | pass |  |
| 2 | Delete product from the system | Pharmacist enter product code and then click on delete button | Message: product deleted successfully | Product deleted successfully | pass |  |
| 3 | Delete product from the system | Pharmacist enter product code that not in the system and  click on  delete button | Message: Error product not found | Product not found | pass |  |

**Add patient Test Case**

**Test Case ID:** 1.4 **Test Designed By:**

**Test Priority (Low/medium/Large):** Med **Test Designed Date:**

**Module Name:** add patient module **Test Executed By:**

**Test Title:** add patient to the system **Test Execution Date:**

**Pre-Conditions:** login to the system

**Dependencies:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Step | Test Steps | Test Data | Expected Data | Actual Result | Status(Pass/Fail) | Notes |
| 1 | Navigate to Patient Page | Press on Patient Button | Pharmacist must navigate to patient page | Pharmacist navigated to patient page | pass |  |
| 2 | Pharmacist enter patient data | Pharmacist enter name of patient, address, phone, username, password, number,  patient history, id then click on add button | Message: patient added successfully to the system | Patient added successfully to the system | pass |  |

**Add Staff Test Case**

**Test Case ID:** 1.5 **Test Designed By:**

**Test Priority (Low/medium/Large):** Med **Test Designed Date:**

**Module Name:** add Staff module **Test Executed By:**

**Test Title:** add Staff to the system **Test Execution Date:**

**Pre-Conditions:** login to the system

**Dependencies:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Step | Test Steps | Test Data | Expected Data | Actual result | Status (pass/Fail) | Notes |
| 1 | Navigate to Administrator Page | Press on administrator button | Administrator navigate to administrator page | Administrator navigated to administrator page | pass |  |
| 2 | Administrator enter staff data | Administrator enter name of the staff, username, password, phone number , address , worked hours , salary , id and then click on add staff button | Message: Staff Data is Added Successfully | Staff Data Added Successfully | pass |  |

**Calculate profit Test Case**

**Test Case ID:** 1.6 **Test Designed By:**

**Test Priority (Low/medium/Large):** Med **Test Designed Date:**

**Module Name:** Calculate profit module **Test Executed By:**

**Test Title:**  calculation of profit **Test Execution Date:**

**Pre-Conditions:**  check whether the account is for administrator or pharmacist

**Dependencies:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Step | Test Steps | Test Data | Expected Data | Actual Result | Status(Pass/Fail) | Notes |
| 1 | Navigate to Administrator Page | Press on administrator button | Administrator  navigate to administrator page | Administrator navigated to administrator page | pass |  |
| 2 | Administrator calculate profit | Administrator enter name, revenues, expenses and worked hour, unit number, price and salary per hour then click on calculate button | Profit will be calculated | Profit calculated successfully | pass |  |

**Increment Product Test Case**

**Test Case ID: 1.7 Test Designed By:**

**Test Priority (Low/Medium/Large): Med Test Designed Date:**

**Module Name: Increment Product module Test Expected Data:**

**Test Title: Product incremented in stock Test Execution Data:**

**Pre-Conditions: Check product code, name, formula.**

**Dependencies:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Steps | Test Steps | Test Data | Expected Result | Actual  Result | Status  (Pass/Fail) | Notes |
| 1 | Navigate to product page | Press on product button | Pharmacist must navigate to products page | Pharmacist navigated to products page | Pass |  |
| 2 | Pharmacist enter products data of a product in the system | Pharmacist enter the product code, name, formula and then click on increment button | Message: The product incremented successfully | Pharmacist increment product in the stock | Pass |  |
| 3 | Pharmacist enter products data of a product doesn’t exist in the system | Pharmacist enter the product code, name, formula and then click on increment button | Message: The product wasn’t exist | Pharmacist did not find the product | pass |  |

**Total price Test Case**

**Test Case ID: 1.8 Test Designed By:**

**Test Priority (Low/Medium/Large): Med Test Designed Date:**

**Module Name: Total price module Test Expected Data:**

**Test Title: Total Price of the Order Test Execution Data:**

**Pre-Conditions: Patient order a product**

**Dependencies:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Steps | Test Steps | Test Data | Expected Result | Actual  Result | Status  (Pass/Fail) | Notes |
| 1 | Navigate to patient page | Press on patient button | patient must navigate to patient page | patient navigated to patient page | Pass |  |
| 2 | patient enter order data | patient enter the products that he wanted then click on calculate button | Message: Total price calculated successfully | total price was being calculated | Pass |  |

**Pay Test Case**

**Test Case ID: 1.9 Test Designed By:**

**Test Priority (Low/Medium/Large): Med Test Designed Date:**

**Module Name: pay module Test Expected Data:**

**Test Title: The way of payment Test Execution Data:**

**Pre-Conditions: Patient order a product**

**Dependencies:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Steps | Test Steps | Test Data | Expected Result | Actual  Result | Status  (Pass/Fail) | Notes |
| 1 | Navigate to order page | Press on order button | Patient must navigate to order page | Patient navigated to order page | Pass |  |
| 2 | Patient choose payment method | Patient select the choice by credit card or paypal | Message: The patient choose method successfully | Patient choose payment method | Pass |  |

**send alert Test Case**

**Test Case ID: 2.0 Test Designed By:**

**Test Priority (Low/Medium/Large): Med Test Designed Date:**

**Module Name: send alert Test Expected Data:**

**Test Title: Alert to increase products Test Execution Data:**

**Pre-Conditions: Product less than or equal 20%**

**Dependencies:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Steps | Test Steps | Test Data | Expected Result | Actual  Result | Status  (Pass/Fail) | Notes |
| 1 | System will check quantity of product order by order to see if it wants extension with the shortage product or not | Check automatically | Product quantity was checked | Product quantity was checked | Pass |  |
| 2 | May need extension for the product that will be under the normal level | Check automatically | Alert will be sent | Alert will be sent | Pass |  |
| 3 | May not need extension for the product that will be under the normal level | Check automatically | Nothing won’t happen just decrement product | Nothing won’t happen just decrement product | pass |  |

**View alert Test Case**

**Test Case ID: 2.1 Test Designed By:**

**Test Priority (Low/Medium/Large): Med Test Designed Date:**

**Module Name: View alert Test Expected Data:**

**Test Title: Alert to increase products Test Execution Data:**

**Pre-Conditions: Alert was sent**

**Dependencies:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Steps | Test Steps | Test Data | Expected Result | Actual  Result | Status  (Pass/Fail) | Notes |
| 1 | System will check the send alert process | Check automatically | Product quantity was checked | Product quantity was checked | Pass |  |
| 2 | Product quantity was under normal level | Product quantity was checked | Message: your inventory level is under normal level | View alert message your inventory level is under normal level | Pass |  |
| 3 | Product is in normal level | Product quantity was checked | Nothing will send | Nothing will send | pass |  |

**Decrement Product Test Case**

**Test Case ID: 2.2 Test Designed By:**

**Test Priority (Low/Medium/Large): Med Test Designed Date:**

**Module Name: Decrement Product module Test Expected Data:**

**Test Title: Product decremented in stock Test Execution Data:**

**Pre-Conditions: check order list.**

**Dependencies:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Steps | Test Steps | Test Data | Expected Result | Actual  Result | Status  (Pass/Fail) | Notes |
| 1 | Navigate to order page | Press on order button | patient must navigate to order page | patient navigated to order page | Pass |  |
| 2 | patient enter product data of an order in the system  if the product quantity is more than or equal the quantity of order needed | patient enter the product code, name, formula and the quantity needed then click on decrement button | Product quantity will decrement automatically | Product quantity will decrement automatically | Pass |  |
| 3 | patient enter product data of an order in the system  if the product quantity is less than the quantity of order needed | patient enter the product code, name, formula and the quantity needed then click on decrement button | Message: Insufficient amount in storage | Insufficient amount in storage | pass |  |

**Calculate salary Test Case**

**Test Case ID:** 2.3 **Test Designed By:**

**Test Priority (Low/medium/Large):** Med **Test Designed Date:**

**Module Name:** Calculate sales module **Test Executed By:**

**Test Title:**  calculation of salary **Test Execution Date:**

**Pre-Conditions:**  check whether the account is for administrator or pharmacist

**Dependencies:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Step | Test Steps | Test Data | Expected Data | Actual Result | Status(Pass/Fail) | Notes |
| 1 | Navigate to Administrator Page | Press on administrator button | Administrator  navigate to administrator page | Administrator navigated to administrator page | pass |  |
| 2 | Administrator calculate salary | Administrator enter name, revenues, expenses and worked hour, unit number, price and salary per hour then click on calculate button | salary will be calculated | salary calculated successfully | pass |  |

**Calculate sales Test Case**

**Test Case ID:** 2.4 **Test Designed By:**

**Test Priority (Low/medium/Large):** Med **Test Designed Date:**

**Module Name:** Calculate sales module **Test Executed By:**

**Test Title:**  calculation of sales **Test Execution Date:**

**Pre-Conditions:**  check whether the account is for administrator or pharmacist

**Dependencies:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Step | Test Steps | Test Data | Expected Data | Actual Result | Status(Pass/Fail) | Notes |
| 1 | Navigate to Administrator Page | Press on administrator button | Administrator  navigate to administrator page | Administrator navigated to administrator page | pass |  |
| 2 | Administrator calculate sales | Administrator enter name, revenues, expenses and worked hour, unit number, price and salary per hour then click on calculate button | sales will be calculated | sales calculated successfully | pass |  |

**Get product Test Case**

**Test Case ID:** 2.5 **Test Designed By:**

**Test Priority (Low/medium/Large):** Med **Test Designed Date:**

**Module Name:** Get product module **Test Executed By:**

**Test Title:** Get product **Test Execution Date:**

**Pre-Conditions:** add product

**Dependencies:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Step | Test Steps | Test Data | Expected result | Actual result | Status (Pass/Fail) | Notes |
| 1 | Navigate to product page | Press on product button | patient must navigate to products page | patient navigated to products page | pass |  |
| 2 | patient get product data that was added | Check the product data that was added automatically | Show all the product that was added | Show all the product that was added | Pass |  |