**Project Title**

**Pran Distributor Product Delivery Management System**

**Rakibul Alam (00155394)**

**Daffodil International Academy**

**Abstract**

In modern era, People convert their work from manual to automated system in remarkably. It is used to do people work easy and saving time for them. Pran Distributor Product delivery management is a desktop application which will productive for user and time worthy for him.

For developing the system, I used a OOP language called JAVA. I have used Netbeans IDE and agile methodologies and relational database concept are implemented in the projects. MoSCoW prioritization techniques are used to throughout the development process.

This project objective is lessened distributor work load when he delivering and counting sells product and the system also help him to make decision in his business perspective.

Contents

[Acknowledgement 4](#_Toc497119736)

[Project Initiation 5](#_Toc497119737)

[Introduction: 5](#_Toc497119738)

[The way system developed: 5](#_Toc497119739)

[Background Study: 5](#_Toc497119740)

[Problem with current system: 6](#_Toc497119741)

[Solution of the problems: 6](#_Toc497119742)

[Project Description: 6](#_Toc497119743)

[Alternative Option: 6](#_Toc497119744)

[Chosen solution: 7](#_Toc497119745)

[Justification of methods and frameworks: 7](#_Toc497119746)

[The methodology I have chosen: 8](#_Toc497119747)

[The Project aims and objectives: 8](#_Toc497119748)

[Feasibility study 9](#_Toc497119749)

[Economic Feasibility: 9](#_Toc497119750)

[Technical Feasibility: 9](#_Toc497119751)

[Operational Feasibility: 9](#_Toc497119752)

[Analysis 10](#_Toc497119753)

[Requirements speciation: 10](#_Toc497119754)

[Functional Requirements: 10](#_Toc497119755)

[Non- functional Requirements: 10](#_Toc497119756)

[Requirements Prioritization: 11](#_Toc497119757)

[Must Have: 11](#_Toc497119758)

[Should Have: 11](#_Toc497119759)

[Could Have: 12](#_Toc497119760)

[Won’t Have: 12](#_Toc497119761)

[Use case Diagram: 13](#_Toc497119762)

[Architecture: 14](#_Toc497119763)

[Design 15](#_Toc497119764)

[Activity diagram: 16](#_Toc497119765)

[Normalization: 17](#_Toc497119766)

[Data Dictionary: 18](#_Toc497119767)

[Entity Relationship Diagram (ERD): 18](#_Toc497119768)

[Implementation 20](#_Toc497119769)

[The choice of programming language: 20](#_Toc497119770)

[Development architecture to system architecture: 20](#_Toc497119771)

[Training: 20](#_Toc497119772)

[Testing 21](#_Toc497119773)

[Functionality those are going to be tested: 21](#_Toc497119774)

[Unit Testing: 21](#_Toc497119775)

[Integration testing: 26](#_Toc497119776)

[Usability testing: 27](#_Toc497119777)

[Test environment: 29](#_Toc497119778)

[Test execution: 29](#_Toc497119779)

[Critical evaluation 41](#_Toc497119780)

[System Strength: 41](#_Toc497119781)

[System Weakness: 41](#_Toc497119782)

[Problems Identified & Resolved 41](#_Toc497119783)

[Further Improvement: 42](#_Toc497119784)

[Conclusion: 42](#_Toc497119785)

[References 43](#_Toc497119786)

[Appendices 44](#_Toc497119787)

[Requirements Catalogue: 44](#_Toc497119788)

[Use case Description: 44](#_Toc497119789)

[Test Script: 45](#_Toc497119790)

[User Guide: 45](#_Toc497119791)

[System Code: 46](#_Toc497119792)

# Acknowledgement

To complete the project, I am lucky to have support, help and encouragement from lot of people. For their cooperation, I would like to acknowledge them.

First and foremost, I deeply thankful Md. Azizur Rahman, my project advisor for wonderful guideline during project with support and knowledge. I also thankful for his advice, assistance, continuous feedback and encouragement. His hardworking attitude and wide knowledge inspire me deeply to progress my life.

Finally, I dedicate the project to my parents, help and inspiration throughout the project.

# Project Initiation

## **Introduction:**

The importance of automation system is growing up in the modern system remarkably. So, many organization are shift from manual system to automated system. For developing a system, it is important to have good knowledge how current system is working. That’s why I have included project background. It is also essential to know how the system will be developed and what frameworks and tools will be used and the aims and objective of the project.

## **The way system developed:**

## Background Study:

Pran is a well-known organization which sells their products through a distributor in every region. The distributor of Pran using a manual system to calculate his delivery and selling products. He used to various order collection memos to delivery his products and then also counts a returning product from the market. Pran sells its product dividing in groups. For this reason, salesmen collect their order from retailer and wholesaler using different memos. Then salesmen collect products following these memos. Distributor delivery the products using these memos. After selling these product, when salesman come to the distributor with return some products, then distributor need to collect these products and he generally minus collecting products from delivering products to find out actual selling product at daily. In this case, the distributor two group of products of Pran. These are:

1. Noodles
2. Culinary

## **Problem with current system:**

1. It is difficult to calculate a large amount of delivery and return products data to find out the actual sells. It is time consuming and costly process.
2. It is difficult to realize market situation using the current system.
3. It is difficult to know which product is mostly sells in a week, month or year which is important to progress the business situation. This is totally less productive.

## **Solution of the problems:**

## Project Description:

My proposed system has only one user who is the distributor for a specific region. The user can log in and select product and sell easily using the system. He can manage product efficiently using the system.

## Alternative Option:

The system can be developed as mobile, web or desktop based system.

1. In case of mobile and tab apps, all data will be stored in remotely. To run the application, it need to install in particular device.
2. For web application, need a remote server host to store data. It is mandatory to use internet to access the data.
3. For the desktop application, data stored in a local server. Here, not mandatory of internet connection. But have no accessibility of data from remote places.

## Chosen solution:

Here, I have used desktop based system because it is for a specific distributor to manage his work. Here need not to access data from remote. Also, using desktop application have less security impacts.

## **Justification of methods and frameworks:**

A software or system development methodology is a software development framework which is using the process of developing of an information system in order to structure, plan and control the development.

There are many different methodologies are used during development the project.

* Agile
* Crystal Methods
* Dynamic Systems Development Model (DSDM)
* Extreme Programming (XP)
* Feature Driven Development (FDD)
* Joint Application Development (JAD)
* Lean Development (LD)
* Rapid Application Development (RAD)
* Rational Unified Process (RUP)
* Scrum
* Spiral
* Systems Development Life Cycle (SDLC)
* Waterfall (a.k.a. Traditional)

(IT Knowledge Portal, 2017)

Agile is a conceptual framework for developing a software project. It is using iterative, incremental development process for faster delivery of the software.

There are a number of agile methodology. There are:

* Crystal Methods
* Dynamic Systems Development Model (DSDM)
* Scrum

## **The methodology I have chosen:**

Agile is effective for small/medium project. I have to complete my project within one month. That’s why I have chosen DSDM Arten agile methodology.

## **The Project aims and objectives:**

**Aims:**

* To make the delivery system easier
* To make the calculation system efficient
* To improve productivity of the business
* To store all product data in a database
* To improve decision making for the distributor

Objectives:

* Easy to maintain delivery
* Auto calculate of selling products
* Product managing facilities
* Facilities of generating monthly, yearly product selling report.
* Option to print immediate selling product information.
* Option to Distributor easily find each category products.

# **Feasibility study**

A feasibility study is an analysis of how successfully a project can be completed, accounting for factors that affect it such as economic, technological, legal and scheduling factors. Project managers use feasibility studies to determine potential positive and negative outcomes of a project before investing a considerable amount of time and   
money on it.

(investopedia, 2017)

## **Economic Feasibility:**

Using the system, the user can easily manage his product. It is time saving and cost worthy. Its interface is so easy, user needn’t take training to use it. As, it is desktop application user need not any domain-hosting cost and also need not internet for using the application.

## **Technical Feasibility:**

The proposed system is not design as platform independent. It is developed for windows operating system platform. The system runs with validation and shows actual result.

## **Operational Feasibility:**

The system is not fully operationally successful. Because all the function is not meets the requirements. Such as. It is not able to categorized product and generated report. But calculation and manage product accurately.

# **Analysis**

## **Requirements speciation:**

There have functional and not function requirements to develop a project. Then I am going to prioritize the requirements by using the technique- MoSCoW prioritization. Here I have also included ERD, Use case diagram.

## **Functional Requirements:**

Statements of services the system should provide, how the system should react to particular inputs and how the system should behave in particular situations.

(Sommerville, n.d.)

The functional requirements of my system are:

1. Manage Products
2. Calculate product price
3. Generate table to holds product information
4. Calculate tables value to find out final information

## **Non- functional Requirements:**

Constraints on the services or functions offered by the system such as timing constraints, constraints on the development process, standards are non-functional requirements.

The non-functional requirements of my system are:

1. Validation
2. Verification
3. Reliability
4. Response time
5. System efficiency
6. Accessibility

## Requirements Prioritization:

During software development, requirements need to prioritize because of having budgetary constraints, tight deadlines. The prioritization techniques are:

* Ranking
* Numerical Assignment
* MoScoW techniques
* Bubble sort techniques
* Analytic Hierarchy Process (AHP)

I am using DSDM Artern methodology so, DSDM Artern preferred MoSCoW prioritization requirements. I am utilizing MoSCoW prioritization for the project. MoSCoW prioritization stands for should have, must have, would have and could like but won’t get.

### **Must Have:**

These requirements must be implemented in the project:

1. User log in facilities
2. User new product adding facilities
3. User editing product facilities
4. Input delivering and return product facilities
5. Calculate selling product facilities

### **Should Have:**

These requirements need to implemented in the project:

1. Categorized product

2. Generated report

3. Store selling information by date

4. Using graphical chat by understanding selling information

### **Could Have:**

1. Searching product by name
2. Searching product by category
3. Every product with image

### **Won’t Have:**

1. Multiple user to edit products

## **Use case Diagram:**

A use case diagram show how user interacts with system. Description is included in appendix.

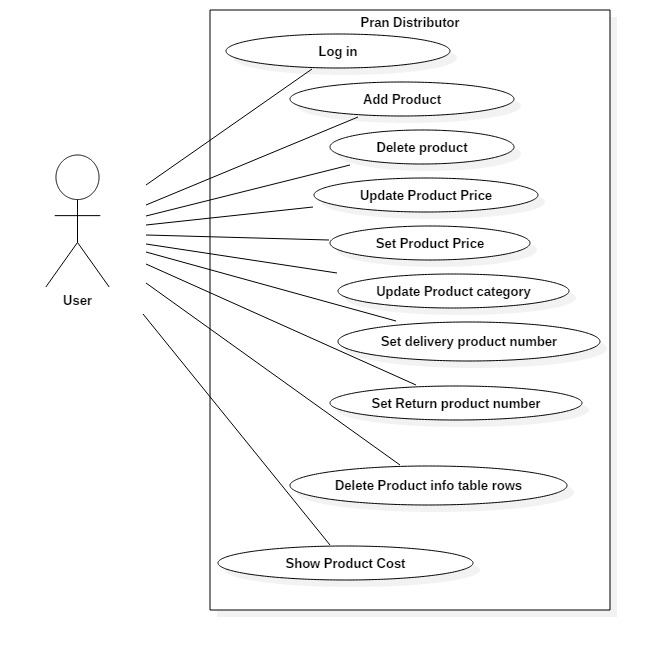


Figure: Use case diagram

# **Architecture:**

In architecture have system architecture and class diagram. In the system, have no interact with online server. Here, used local server.

System architecture:

In this case, user interacts with system and system interacts with database via local connection between them.

# **Design**

Here, I have included details class diagram, Entity Relationship diagram, Normalization of data, data dictionary and activity diagram to describe the design of the system.

Detailed class diagram:

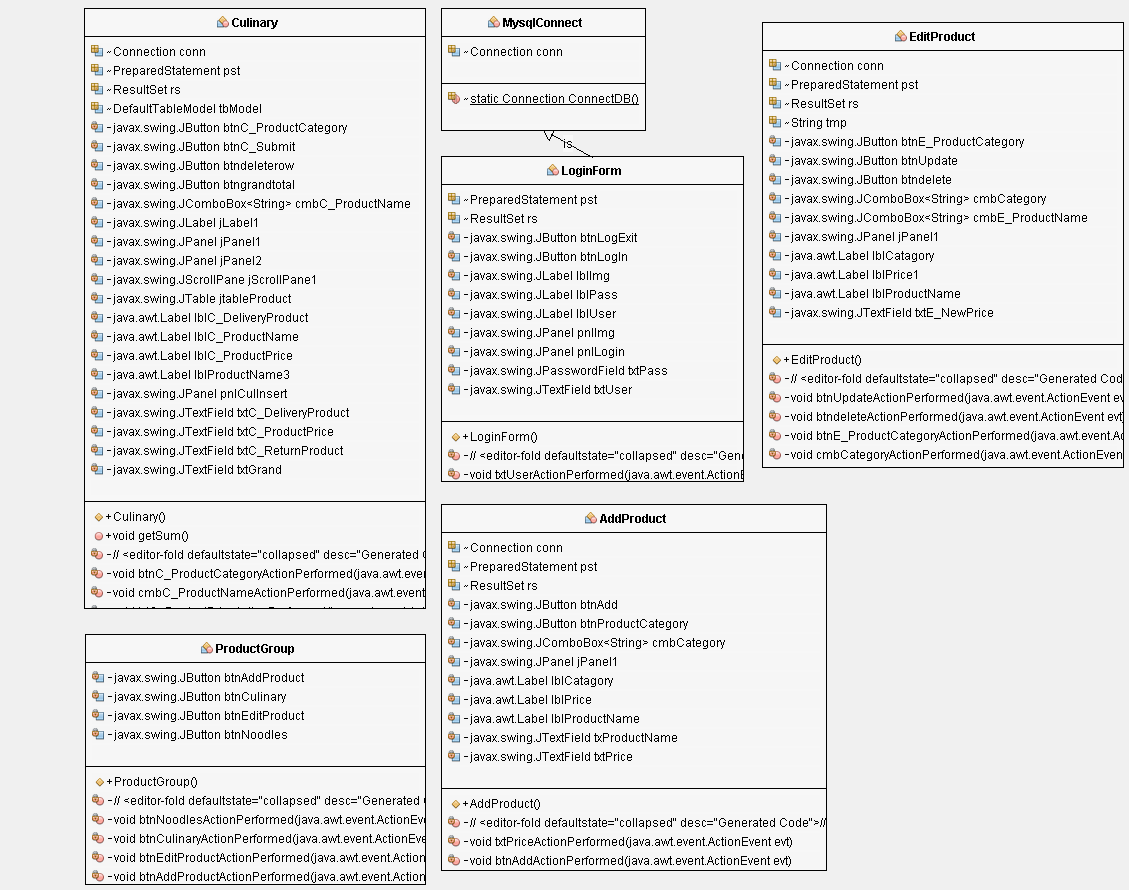
The description of class diagram is added in the appendix.

Figure: Class diagram

## **Activity diagram:**

[Behavior diagrams](http://www.uml-diagrams.org/uml-24-diagrams.html#behavior-diagram) show the **dynamic behavior** of the objects in a system, which can be described as a series of changes to the system over **time.** (uml-diagrams organization, 2009)

Activity diagram is a behavior diagram.

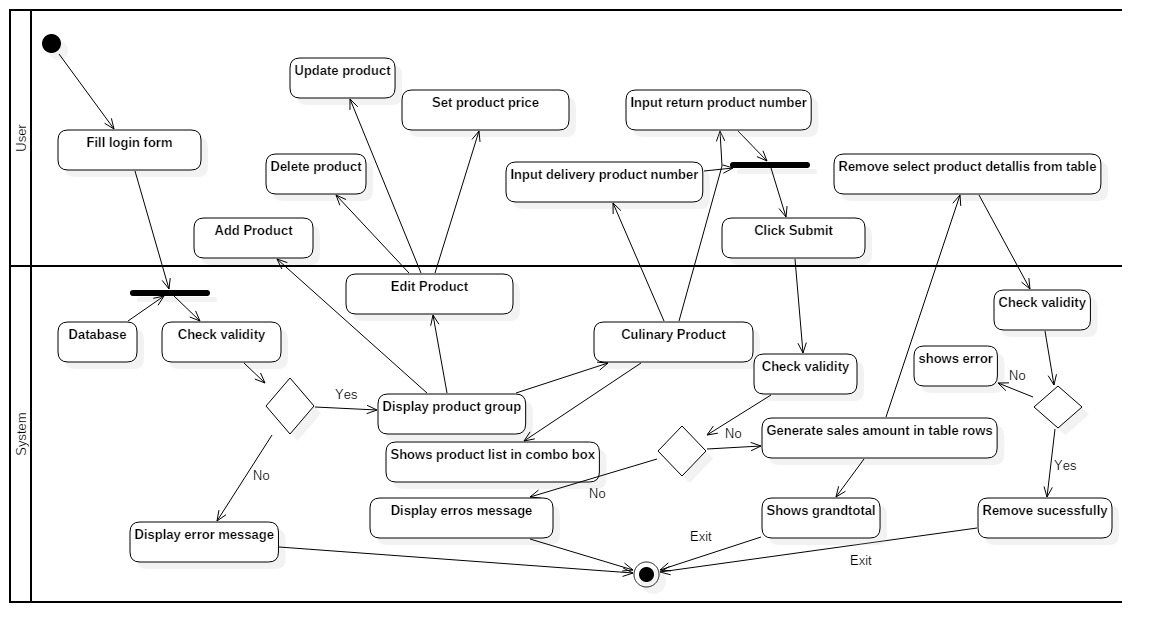


Figure: Activity Diagram

## **Normalization:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | | | | | |
| Table | UNF | 1NF | Level | 2NF | 3NF |
| Productdetails  Category  ProductCategory  Userlogin | **ProductId** (Candidate key)  ProductName  ProductPrice  CategoryId  CategoryName  Userid  UserName  Password | ProductId  ProductName  ProductPrice  Userid  UserName  Password  CategoryId  CategoryName  ProductId | 1  1  1  1  1  1  2  2  2 | ProductId  ProductName  ProductPrice  Userid  UserName  Password   |  | | --- | | Category | | CategoryId (PK) | | CategoryName | | ProductId (FK) | | |  | | --- | | Productdetails | | ProductId (PK) | | ProductName | | ProductPrice |  |  | | --- | | UserlogIn | | UserId (PK) | | UserName | | Password |  |  | | --- | | Category | | CategoryId (PK) | | CategoryName | | ProductId (FK) | |

Normalization is used to reduce data redundancy. In general, there steps are following to normalize from unnormalized data.

## **Data Dictionary:**

Data dictionary defines the structure of database. A data dictionary is a file or a set of files that contains a database's metadata. Here, I explained the structure of all attributes and table of the system.

**Productdetails**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attributes | Data Type | Length | Primary key | Foreign key | Reference Table |
| ProductId | int | 11 | yes |  |  |
| ProductName | varchar | 30 |  |  |  |
| ProductPrice | double |  |  |  |  |

**Userlogin**

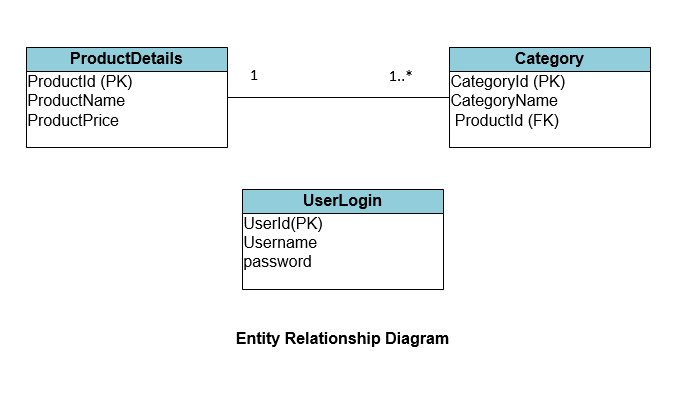
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attributes | Data Type | Length | Primary key | Foreign key | Reference Table |
| UserId | int | 11 | yes |  |  |
| UserName | varchar | 50 |  |  |  |
| Password | varchar | 30 |  |  |  |

**Category**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attributes | Data Type | Length | Primary key | Foreign key | Reference Table |
| categoryId | int | 11 | yes |  |  |
| CategoryName | varchar | 100 |  |  |  |
| ProductId | int | 11 |  | yes | Productdetails |

## **Entity Relationship Diagram (ERD):**

It is used to show the relationship between the tables. The entity relationship diagram of the proposed system is following:



By analyzed using different diagram the proposed system is explained. I used activity diagram for design the behavior of the system and also use class diagram and ERD diagram to make system more understandable.

# **Implementation**

## **The choice of programming language:**

For the implement of the system, I have used JAVA as programming language. Because it is platform independent. So, Here I develop the system for windows. In future, I can be used it for IOS or UNIX operating system nodes. JAVA is object oriented language and have OOPs features. So, portability and consider the system architecture, I have chosen JAVA for development the system.

## **Development architecture to system architecture:**

I have used Netbeans IDE for development the project. Here, I have created database connection to my application using mysql database connector. I used mysql server to store data. Using these, I am trying my best to implement the architecture of the system.

To do the implementation, I have used agile methodology. As it is a solo project, I have no team member without me. As it is an iterative approach, I work using the plan day by day. In the middle of the task, I have changed my plan without discussion anyone because here I am the stakeholder and project manager in the project. Here I have used MoSCoW prioritization technique. The system meets only “must have” of the system. If I get enough time, I will develop the “should have” and “could have: function.

## **Training:**

For using the system, I don’t propose any training session. Because it is easy to use. User can use it, only using user manual. Which I have included in the appendix.

# **Testing**

Testing is important during system development. Test driven development is needed to fixed system bugs, check user friendliness, reliability and meet the system standard. Here, I have covered unit testing, integration testing and usability testing to check the system.

### Functionality those are going to be tested:

1. User login
2. Adding Product
3. Deleting Product
4. Updating Product
5. Count selling product
6. Removing selling info
7. Calculate grand total

## **Unit Testing:**

A unit test is a way of testing a unit - the smallest piece of code that can be logically isolated in a system. It helps us to find out algorithmic problem of the system and the function and features are running is provide expected results or not. (SMARTBEAR, 2011)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Unit Test 1 | | Tests Class: AddProduct | Designed By: Rakibul Alam | |
| Data Source: user entry | | Objective: Basic functionality Testing | Tester: Rakibul Alam | |
| Test  Case | Description | Tasks | Expected  Result | Actual  Result |
| 1.1 | Basic functionality testing: ‘product adding’ by user | Enter product Name:  Product name: Chilli masala | Value is added to the Productdetails table in database | Same |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Unit Test 2 | | Tests Class: EditProduct | Designed By: Rakibul Alam | |
| Data Source: user entry | | Objective: Basic functionality Testing | Tester: Rakibul Alam | |
| Test  Case | Description | Tasks | Expected  Result | Actual  Result |
| 1.2 | Basic functionality testing: ‘product updating’ by user | Enter product New price:  Product name: Chilli masala  Price: 20.68 | Value is updated in the Productdetails table in database | Same |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Unit Test 3 | | Tests Class: EditProduct | Designed By: Rakibul Alam | |
| Data Source: user entry | | Objective: Basic functionality Testing | Tester: Rakibul Alam | |
| Test  Case | Description | Tasks | Expected  Result | Actual  Result |
| 1.3 | Basic functionality testing: ‘product deleting’ by user | Product Name:  Chilli masala | Value is deleted form the Productdetails table in database | Same |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Unit Test 4 | | Tests Class: EditProduct | Designed By: Rakibul Alam | |
| Data Source: user entry | | Objective: Basic functionality Testing | Tester: Rakibul Alam | |
| Test  Case | Description | Tasks | Expected  Result | Actual  Result |
| 1.4 | Basic functionality testing: ‘delivery product added’ by user | Enter delivery product number  Product name: Chilli masala  Delivery: 20 | Value is added in the form table for calculate actual sells | Same |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Unit Test 5 | | Tests Class: Culinary | Designed By: Rakibul Alam | |
| Data Source: user entry | | Objective: Basic functionality Testing | Tester: Rakibul Alam | |
| Test  Case | Description | Tasks | Expected  Result | Actual  Result |
| 1.5 | Basic functionality testing: ‘return product added’ by user | Enter return product number  Product name: Chilli masala  Return: 4 | Value is added in the form table for calculate actual sells | Same |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Unit Test 6 | | Tests Class: Culinary | Designed By: Rakibul Alam | |
| Data Source: user entry | | Objective: Basic functionality Testing | Tester: Rakibul Alam | |
| Test  Case | Description | Tasks | Expected  Result | Actual  Result |
| 1.6 | Basic functionality testing: ‘calculate cost for sell product’ by user | Enter submit button  Product name: Chilli masala  Product Price: 20.8  Delivery: 20  Return: 4 | Value is added in the form table for calculate actual sell cost. | Same |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Unit Test 7 | | Tests Class: Culinary | Designed By: Rakibul Alam | |
| Data Source: user entry | | Objective: Basic functionality Testing | Tester: Rakibul Alam | |
| Test  Case | Description | Tasks | Expected  Result | Actual  Result |
| 1.7 | Basic functionality testing: ‘calculate grand total for selling products’ by user | Enter submit button  Generate three rows of products details and click submit button to see grand total | Value is shows in the form table in grand total textbox. | Same |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Unit Test 8 | | Tests Class: Culinary | Designed By: Rakibul Alam | |
| Data Source: user entry | | Objective: Basic functionality Testing | Tester: Rakibul Alam | |
| Test  Case | Description | Tasks | Expected  Result | Actual  Result |
| 1.8 | Basic functionality testing: ‘remove one product details from selling products information table’ by user | Enter Remove Product Button  Product name: Chilli masala  Click on chilli masala rows and then click on Remove Product Buttton | Value is shows in the product details form table. | Same |

## **Integration testing:**

Integration testing is how multiple module interacts with the system. To Identify limitation during module interacts this testing is needed.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Integration Test 1 | | Tests Class: ProductGroup | Designed By: Rakibul Alam | |
| Data Source: user entry | | Objective: Basic functionality Testing | Tester: Rakibul Alam | |
| Test  Case | Description | Tasks | Expected  Result | Actual  Result |
| 2.1 | Basic functionality testing: ‘Edit Product form’ by users | * Click Edit Product button. | Edit product form is shown. | Same |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Integration Test 2 | | Tests Class: ProductGroup | Designed By: Rakibul Alam | |
| Data Source: user entry | | Objective: Basic functionality Testing | Tester: Rakibul Alam | |
| Test  Case | Description | Tasks | Expected  Result | Actual  Result |
| 2.2 | Basic functionality testing: ‘Culinary product group shown’ by users | * Click Culinary button. | Culinary form is shown. | Same |

## **Usability testing:**

Usability testing means the system interface how much user friendly. The interface shows proper guideline to user by showing message in real time, then the system will be more user-friendly.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Usability Test 1 | | Tests Class: LoginForm | Designed By: Rakibul Alam | |
| Data Source: user Entry | | Objective: Basic functionality Testing | Tester: Rakibul Alam | |
| Test  Case | Description | Tasks | Expected  Result | Actual  Result |
| 3.1 | Basic functionality testing: ‘Error message testing’ by user | * Click login button without entering right username and password | System will show an error message that username and password don’t match. | Same |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Usability Test 2 | | Tests Class: Culinary | Designed By: Rakibul Alam | |
| Data Source: user Entry | | Objective: Basic functionality Testing | Tester: Rakibul Alam | |
| Test  Case | Description | Tasks | Expected  Result | Actual  Result |
| 3.2 | Basic functionality testing: ‘Selling Error message testing’ by user | User enter negative number like (-200) in delivery or return product and click submit button | System will show an error message that Please enter positive number. | Same |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Usability Test 3 | | Tests Class: Culinary | Designed By: Rakibul Alam | |
| Data Source: user Entry | | Objective: Basic functionality Testing | Tester: Rakibul Alam | |
| Test  Case | Description | Tasks | Expected  Result | Actual  Result |
| 3.3 | Basic functionality testing: ‘remove selling product rows form table Error message testing’ by user | User clicking Remove Product button without selecting any row. | System will show an error message that unable to remove. | Same |

## **Test environment:**

Testing environment is important for getting proper output from test. For proper testing, I have set up pcs with database server.

|  |  |  |
| --- | --- | --- |
| Test Phase | Time | Who is done |
| Unit testing | 2 days | Developer |
| Integration Testing | 1 days | Developer |
| Usability testing | 1 days | Developer/User |

## **Test execution:**

Unit test 1:

Test steps:

* User must be logged in
* Click Add Product button
* Filled the product name

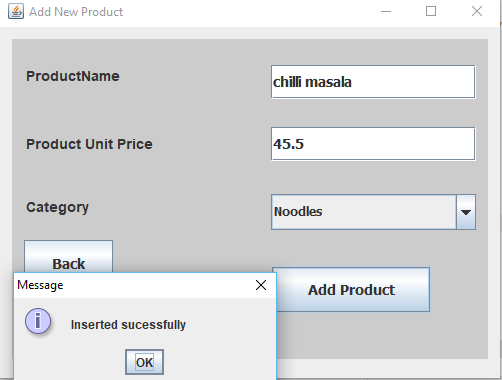


Figure: Add new Product insertion

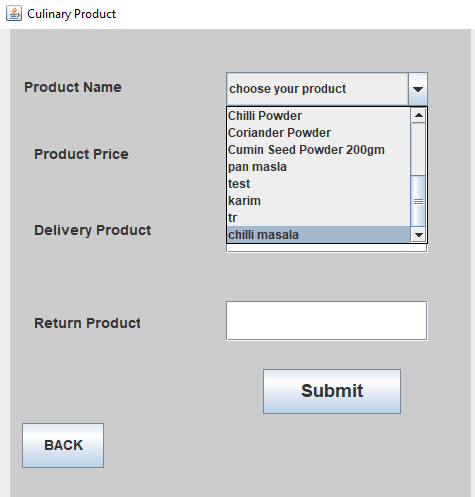


Figure: Add new Product successfully

Unit test 2:

Test steps:

* User must be logged in
* Click Edit Product button
* Set new product price

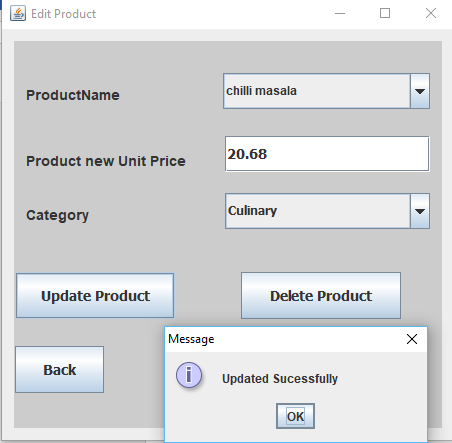


Figure: Update product

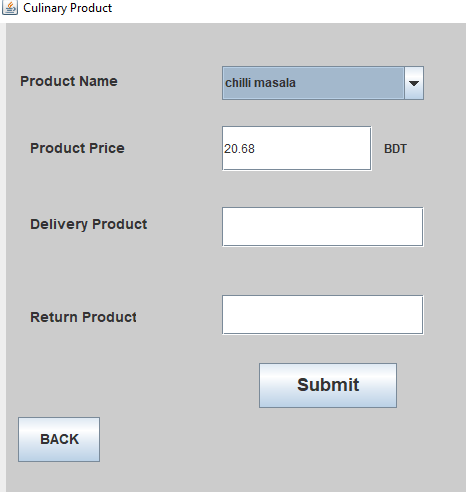


Figure: Product Updated successfully

Unit test 3:

Test steps:

* User must be logged in
* Click Edit Product button
* Click delete button

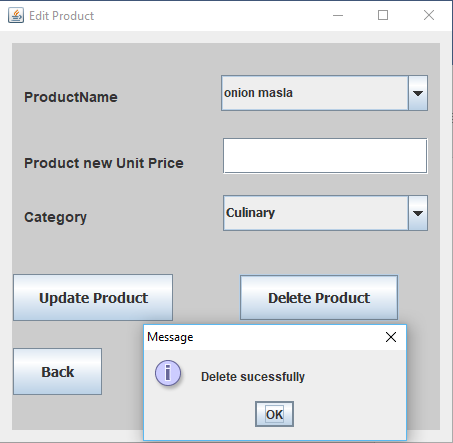


Figure: Delete Product

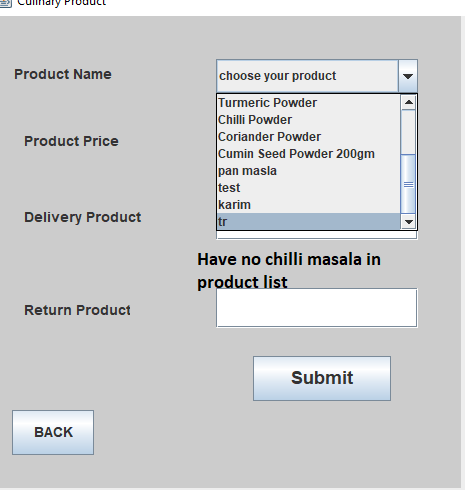


Figure: Delete Product Successfully

Unit test 4,5,6:

Test steps:

* User must be logged in
* Click Culinary button
* Select product
* Enter delivery number
* Enter return number
* Click submit button

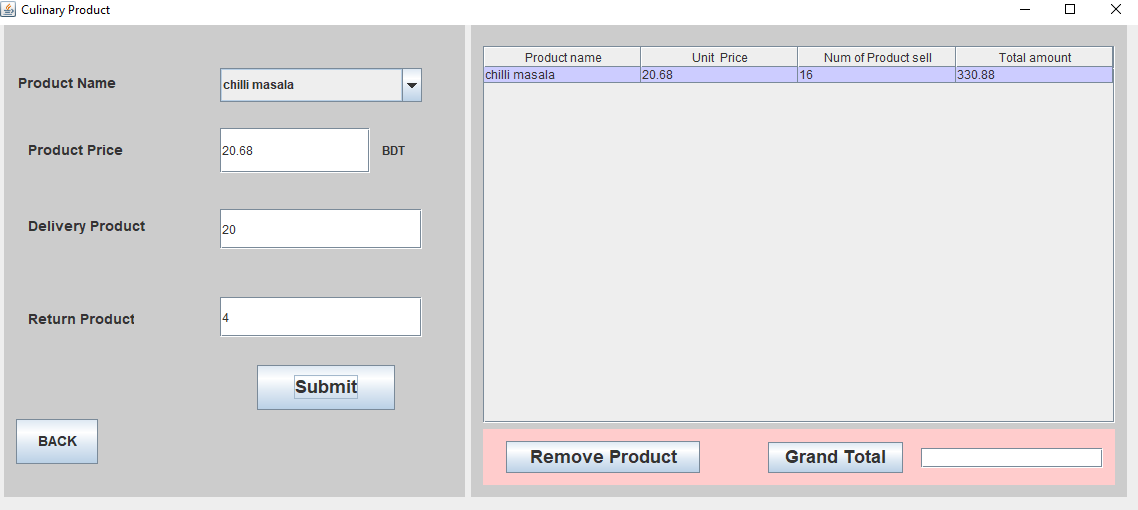


Figure: Generate product sell costs

Unit test 7:

Test steps:

* User must be logged in
* Click Culinary button
* Select products (3 times)
* Enter delivery number (3 times)
* Enter return number (3 times)
* Click submit button (3 times)

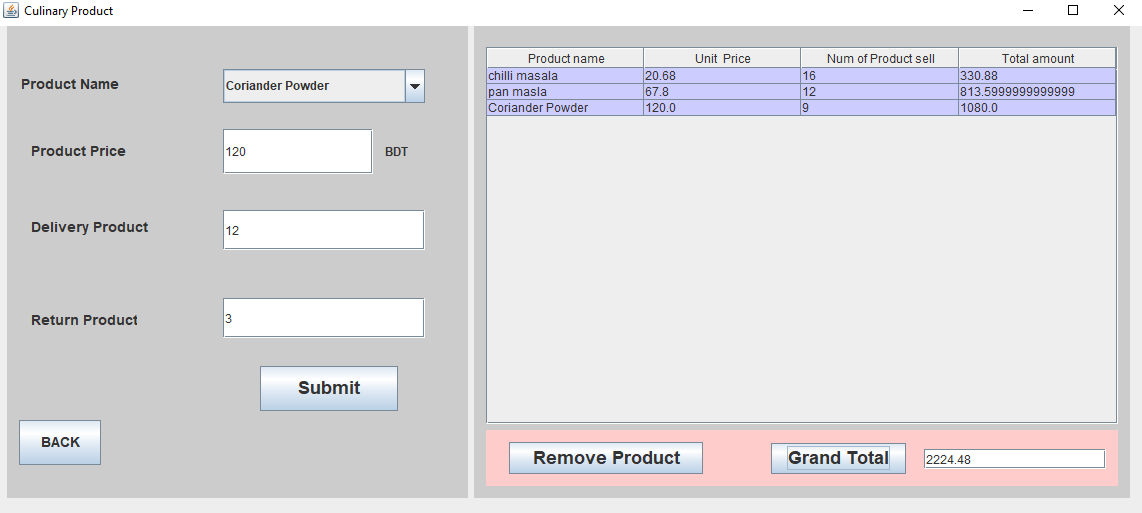


Figure: Shows Grand total of selling product successfully

Unit test 8:

Test steps:

* User must be logged in
* Click Culinary button
* Select products (3 times)
* Enter delivery number (3 times)
* Enter return number (3 times)
* Click submit button (3 times)
* Click one product details for table
* Click Remove Product

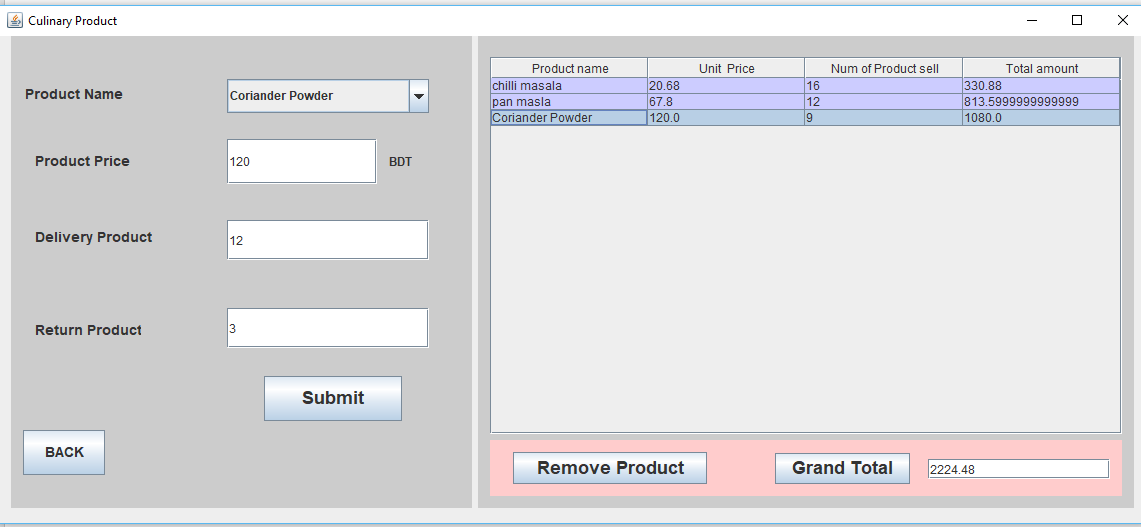


Figure: Selecting one product from table

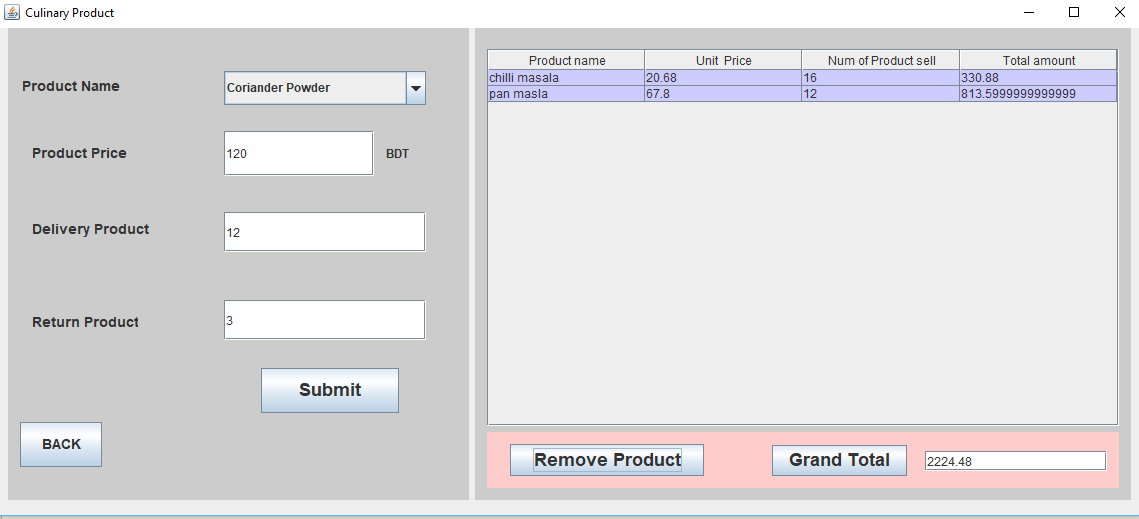


Figure: Remove Product successfully

Integration testing 1:

Test steps:

* User must be logged in
* Click Edit Product button

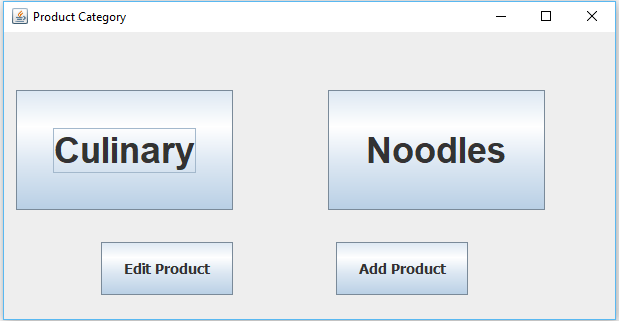


Figure: Showing Edit Product button from product category

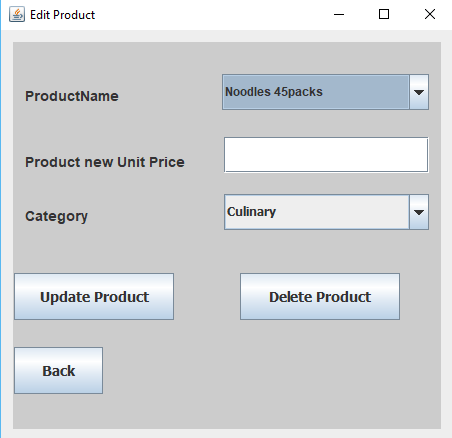


Figure: Showing Edit Product Successfully

Integration testing 2:

Test steps:

* User must be logged in
* Click Culinary button

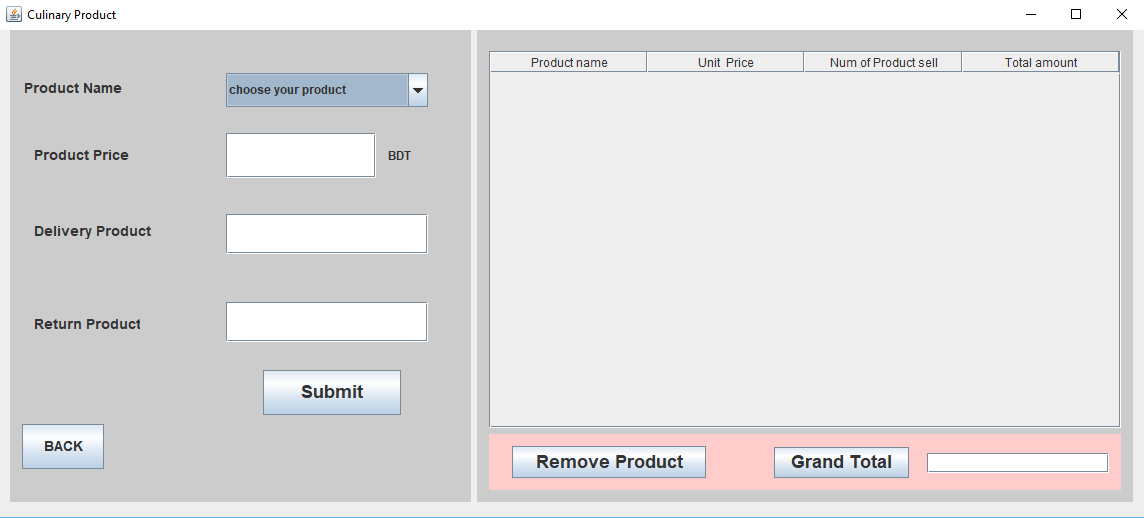


Figure: Showing Culinary Successfully

Usability testing 1:

Test steps:

* User enter wrong password or username
* Click Log in button

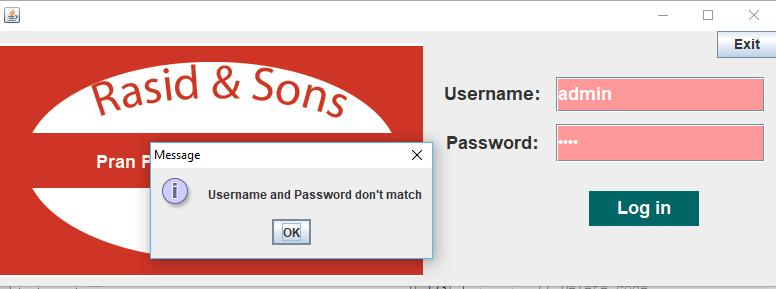


Figure: Showing error massage

Usability testing 2:

Test steps:

* User must be logged in
* Click Culinary button
* Enter negative delivery number

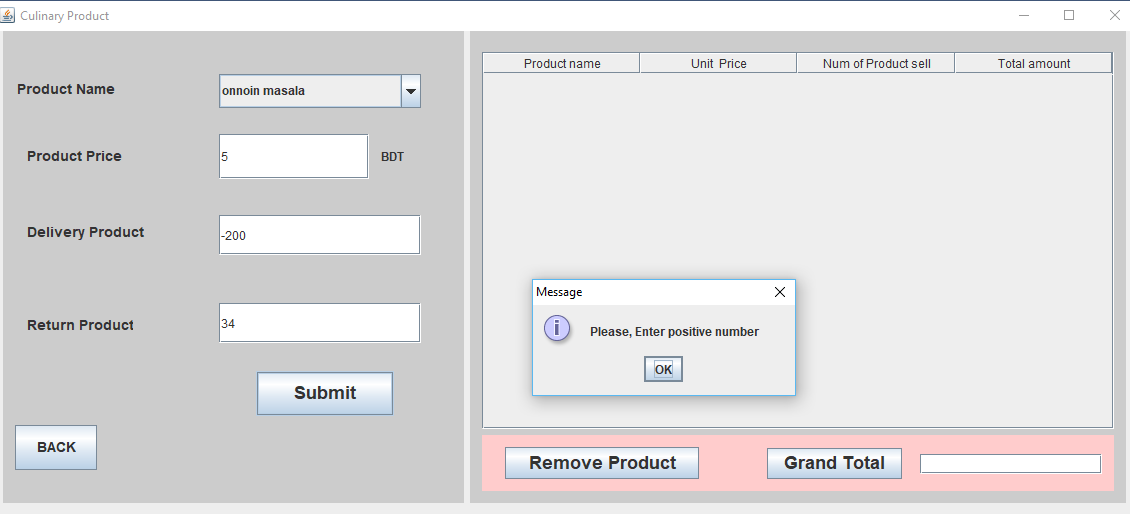


Figure: Showing negative number validation massage successfully

Usability testing 3:

Test steps:

* User must be logged in
* Click Culinary button
* Enter three product details
* Click submit
* Click outside the row in product table
* Click remove button

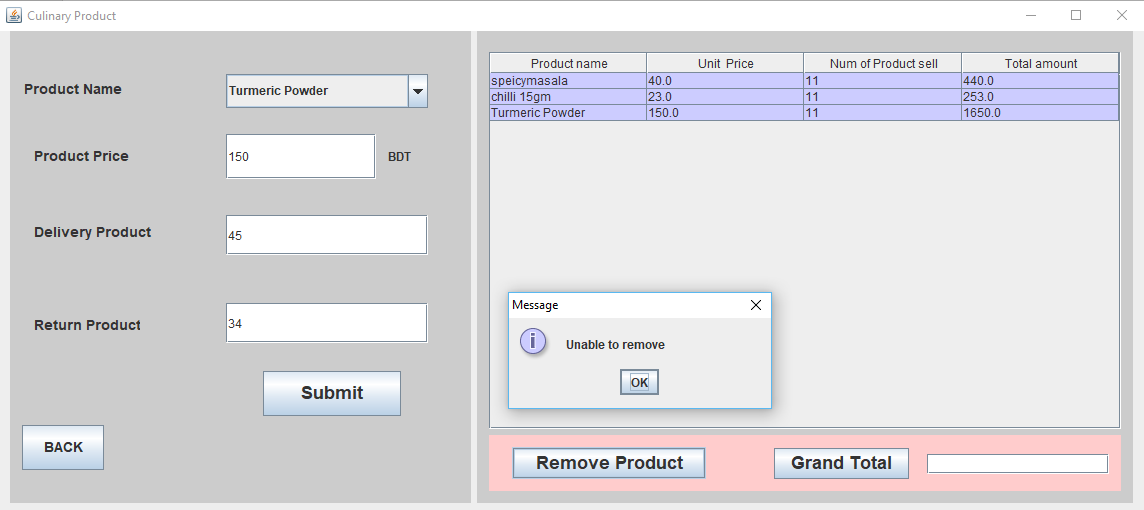


Figure: Show Validation message “Unable to remove”

I have shown the process of testing and the output of the testing here.

# **Critical evaluation**

Here, I will discuss strengths and weakness of the system. Future development also adds here.

## **System Strength:**

* All must have needed functions and features.
* Portable database
* User friendliness
* Normalized database
* Auto calculation of payment

## **System Weakness:**

* Products are not categorized
* No report generated
* Have no printing option
* Not generated graphical chat
* Single level user

## **Problems Identified & Resolved**

I have face problem when I set negative number for delivering product or return product. The system takes it and shows unrealistic output. For example, If I set delivering product number 10 and return product number -10 then the system takes selling product number 20. But return product number -10 is not possible in real world.

To overcome this problem, I have used an “If – else” condition in my system. Which check the number is positive or not. When user enter negative number, the output shows an error message.

## **Further Improvement:**

In future, I will add the functions and features in the system that are following:

1. Categorized products
2. Generate Report with graphical chat
3. The system architecture will be improved
4. Store management
5. Better validation

## **Conclusion:**

I have tried by best to develop the system efficiently. But lack of time I can’t develop some features and function which make the system effective. During the developing and analysis of the project, I have learned a lot of things and it is an interesting journey for me.

# References

investopedia (2017) *Feasibility Study*, [Online], Available: <http://www.investopedia.com/terms/f/feasibility-study.asp> [2 October 2017].

IT Knowledge Portal (2017) *IT Knowledge Portal*, [Online], Available: <http://www.itinfo.am/eng/software-development-methodologies/> [1 October 2017].

SMARTBEAR (2011) *What is unit testing?*, 24 March, [Online], Available: <https://smartbear.com/learn/automated-testing/what-is-unit-testing/> [20 October 2017].

Sommerville, I. (n.d) 'Software Requirements', in Sommerville, I. *Software engineering*.

uml-diagrams organization (2009) *uml-diagrams.org*, [Online], Available: <http://www.uml-diagrams.org/uml-22-diagrams.html> [09 October 2010].

# **Appendices**

## **Requirements Catalogue:**

1. User need a node where have windows operating system.
2. Need mysql server to keep data
3. Need JRE to run to application

## **Use case Description:**

|  |  |
| --- | --- |
| Use case ID: | UC1 |
| Use case Name: | login |
| Description | User can log in the system |
| Pre-Condition |  |
| Standard Flow | 1. User can login this system |
| Post Condition |  |

|  |  |
| --- | --- |
| Use case ID: | UC2 |
| Use case Name: | Add Product |
| Description | User can add Product |
| Pre-Condition |  |
| Standard Flow | 1. User can add Product |
| Post Condition |  |

|  |  |
| --- | --- |
| Use case ID: | UC3 |
| Use case Name: | Delete Product |
| Description | User can delete Product |
| Pre-Condition |  |
| Standard Flow | 1. User can Delete Product |
| Post Condition |  |

|  |  |
| --- | --- |
| Use case ID: | UC4 |
| Use case Name: | Update Product |
| Description | User can Update Product |
| Pre-Condition |  |
| Standard Flow | 1. User can Update Product |
| Post Condition |  |

|  |  |
| --- | --- |
| Use case ID: | UC5 |
| Use case Name: | Set delivery Product Number |
| Description | User can set delivery Product Number |
| Pre-Condition |  |
| Standard Flow | 1. User can set delivery Product Number |
| Post Condition |  |

|  |  |
| --- | --- |
| Use case ID: | UC6 |
| Use case Name: | Set return Product Number |
| Description | User can set Update Product Number |
| Pre-Condition |  |
| Standard Flow | 1. User can set Update Product Number |
| Post Condition |  |

|  |  |
| --- | --- |
| Use case ID: | UC7 |
| Use case Name: | Show Product cost |
| Description | User can show Product cost |
| Pre-Condition |  |
| Standard Flow | 1. User can show Product cost |
| Post Condition |  |

## **Test Script:**

Shown in the test plan.

## **User Guide:**

First need to login with valid username and password. Then Click Culinary/Nodes order to calculate and sells product. To update/delete product user need to go Edit Product form by click Edit Product button and to add product user need to go Add Product form by clicking Add Product button.

## **System Code:**

Login Form:



Figure: Log in form code

Add Product:



Figure: Add Product Code

Delete Product:

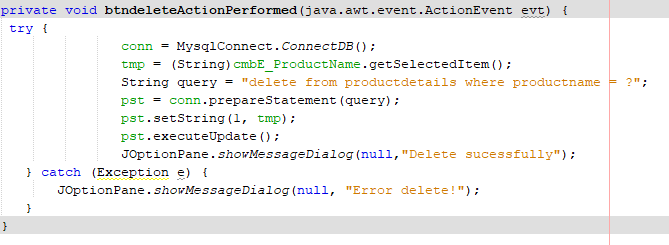


Figure: Delete Product Code

Update Product:

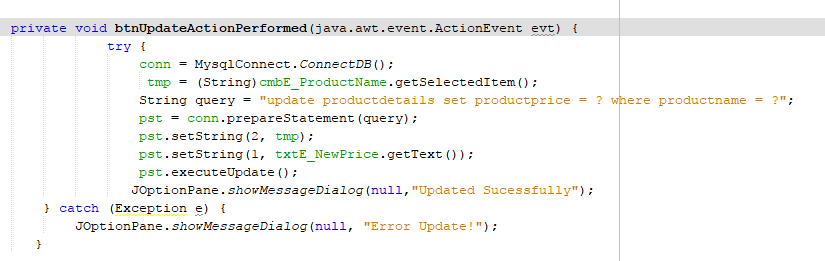


Figure: Update Product Code

Generate sells information code:

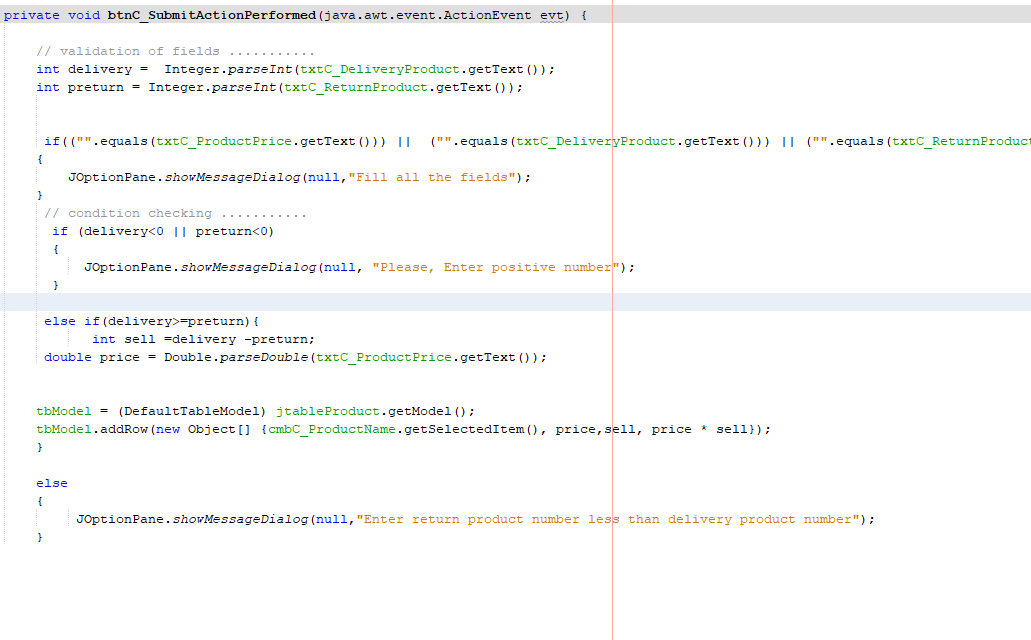


Figure: Generate sells information