

**Assignment Cover Sheet**

|  |  |  |  |
| --- | --- | --- | --- |
| **Qualification** | | **Module Number and Title** | |
| HND in Computing/ HND in Software Engineering | | SED52010 Windows Application Development | |  |
| **Student Name & No.** | | **Assessor** | |
| Widana Gamage Avindi Premaratne – ST/HNDCOM/19/15 | | Aruna Indika | |
| **Hand out date** | | | **Submission Date** |
| //2016 | | | //2016 |
| **Assessment type**  Coursework | **Duration/Length of**  **Assessment Type**  Practical project/report | | **Weighting of Assessment**  **100%** |

|  |  |
| --- | --- |
| **Learner declaration** | |
| I certify that the work submitted for this assignment is my own and research sources are fully acknowledged. | |
| |  |  |  |  | | --- | --- | --- | --- | | **Marks Awarded** | | | | | First assessor | |  | | | IV marks | |  | | | Agreed grade | |  | | | Signature of the assessor |  | Date |  | |

**Feedback Form**

**International College of Business & Technology**

**Module:** Windows Application Development

**Student: Widana Gamage Avindi Premaratne**

**Assessor:**

**Assignment:**

**Strong features of your work:**

**Areas for improvement:**

**Marks Awarded:**

**Learning outcomes covered**

* Critically evaluate and select appropriate windows platform and development tools
* Design and develop Windows applications
* Critically test and deploy Windows applications

**Scenario and the Task**

**“Total Fitness”** is the largest and very busy pharmacy in Matara. They are the sole agent for the Cipla product in the region.

They have number of suppliers to supply verity of medicine and some types of milk powders. Some of them are cash and some of them are credit. For the credit suppliers you have to pay your total bill within 3 weeks. As well that must notify when the particular date comes closer. As well they can terminate any supplier at any time.

As **“Total Fitness”** is the sole agent for the Cipla, they have separate transport division to distribute Cipla products to all over the region. When they deliver particular products to distance pharmacy they will keep following records. Pharmacy name, contact person, contact number (mobile/fix) product(s), quantity, price and any other required details. Normally end of the year they will awarding pharmacies that are having bestselling performance for Cipla products and that should automatically calculate and display by the system.

Also **“Total Fitness”** having two types of customers they are register and non-register. Register customers always get 2% discount for their every bill when they buy more than LKR 750.00. To register; every customer has to pay LKR 2500.00 per year. If someone unable to update their registration automatically they will remove form the registered customers group. As well register customers are allowed to buy products for a 1 week credits, but it only for below LKR 1500.00 bills. Customers cannot buy anything without paying previous credit bills. As well registered customers can check their purchase history.

All purchasing order invoices, purchasing bills, customer bills have to maintain and in any case **“Total Fitness”** wants to find that particular record by all possible ways…

Finally, **Total Fitness** wants to keep all medicines, Cipla product’s, Milk powder product’s inventory up to date…

**Tasks**

1. Critically compare different windows or other platforms and select the suitable platform for different user groups? (Word count 500 [+- 50] ) (10 marks)
2. Critically compare and select development tools and technologies for the above mention application? (Word count 500[+- 50]) (10 marks)
3. Design a solution for above mention requirements. (30 marks) Provide
   1. System architecture diagram
   2. UML diagrams (Class, Use case, sequence etc...)
   3. Data model design (ER diagram)
   4. Sample UI designs (Wireframe will be fine)
4. Develop suitable application based on the design. Should be able to demonstrate and provide all source codes. Need use proper coding standards and must focus on reusability and maintainability of the application. (30 marks)
5. Select appropriate test technique(s). Test and provide proper test cases and critically evaluate the test result. (10 marks)
6. Explain deployment techniques that are selected and critically evaluate suitability of the selection. ( 10 marks)

**Submission Guidelines**

* Submission format Report
* Paper Size: A4
* Words: 3000 words
* Printing Margins: LHS; RHS: 1 Inch
* Binding Margin: ½ Inch
* Header and Footer: 1 Inch
* Basic Font Size: 12
* Line Spacing: 1.5
* Font Style: Times New Roman
* **Referencing should be done strictly using Harvard system**

**Source code, database backup and installation packages should be submitted in a single zip file.**

**Assessment Criteria**

**Task 1 contains 10 marks**

|  |  |  |
| --- | --- | --- |
| **Criteria** | **Marks** | **Marks obtained by the student for the answer provided** |
| **Out of 10** |
| **Excellent**  Excellent level of understanding of platforms, critical comparison technically and also focusing other environment factors, Higher level of depth and breadth of study with extensive reading and integration of information from a wide range of sources. | **7-10** |  |
| **Good**  Reasonable level of understanding of platforms and critical comparison technically, Good level of depth and breadth of study, | **6-7** |  |
| **Satisfactory**  Reasonable level of understanding of platforms and comparison, satisfactory level of depth and breadth of study | **4-6** |  |
| **Poor**  Limited understanding of platforms and poor comparison. Limited reading, lack of depth and breadth of study | **0-4** |  |

**Task 2 contains 10 marks**

|  |  |  |
| --- | --- | --- |
| **Criteria** | **Marks** | **Marks obtained by the student for the answer provided** |
| **Out of 10** |
| **Excellent**  Excellent level of understanding of platforms, critical comparison and proper justification for the selections, Higher level of depth and breadth of study with extensive reading and integration of information from a wide range of sources. | **7-10** |  |
| **Good**  Reasonable level of understanding of platforms and critical comparison technically, Good level of depth and breadth of study, | **6-7** |  |
| **Satisfactory**  Reasonable level of understanding of related tools and comparison with correct selection, satisfactory level of depth and breadth of study | **4-6** |  |
| **Poor**  Limited understanding of related tools and poor comparison. Limited reading, lack of depth and breadth of study | **0-4** |  |

**Task 3 contains 30 marks**

|  |  |  |
| --- | --- | --- |
| **Criteria** | **Marks** | **Marks obtained by the student for the answer provided** |
| **Out of 30** |
| **Excellent**  Excellent level work shown on design, Complete set of UI designs focus on easy navigation and proper structure , Exceptional solution focus on reuse, maintainability, use of proper architecture, and error free,correctness and completeness of the UML diagrams. | **21-30** |  |
| **Good**  Good set of diagram with proper separation of layers and communication, Correct notations and evidence of effort to design user friendly design. Detail diagrams. | **18-21** |  |
| **Satisfactory**  Reasonable level of diagrams, has identified basic data requirements and evidence of basic UML diagrams. | **12-18** |  |
| **Poor**  Limited or no of evidence given for design, incorrect use of notations and diagrams are incomplete. No proper architecture | **0-12** |  |

**Task 4 contains 30 marks**

|  |  |  |
| --- | --- | --- |
| **Criteria** | **Marks** | **Marks obtained by the student for the answer provided** |
| **Out of 30** |
| **Excellent**  Exceptional solution focus on reuse, maintainability, use of proper architecture, error free and innovative features, demonstration with proper planning, proper flow and good presentation skills with clear explanations. | **21-30** |  |
| **Good**  Good solution complete application satisfying all user requirements, Proper error handling, Proper demonstration planed and well structured | **18-21** |  |
| **Satisfactory**  Basic application that can run without any build errors, fulfil the basic requirements. Presentation and demonstration is reasonable but flawed in structure or in some other way | **12-18** |  |
| **Poor**  Develop solution run with build errors, lack of error handling and validation, the presentation and demonstration is incoherent, incomplete or seriously weak in other ways | **0-12** |  |

**Task 5 contains 10 marks**

|  |  |  |
| --- | --- | --- |
| **Criteria** | **Marks** | **Marks obtained by the student for the answer provided** |
| **Out of 10** |
| **Excellent**  Excellent justification for the selected test techniques. Proper set of test cases to conduct a comprehensive test for the develop solution with proper test data. Selection of appropriate test data. Conduct test and critically analysis test results. | **7-10** |  |
| **Good**  Good justification for the selected test technique. Test cases to cover testing of entire application with meaningful data. Conduct test and a critically analysis test results. | **6-7** |  |
| **Satisfactory**  Reasonable justification of test techniques and test cases to test the basic functionalities successfully. Analysis of test results. | **4-6** |  |
| **Poor**  Limited justification for test technique selection and incomplete set of test cases, improper set of test data. Poor analysis of test result. | **0-4** |  |

**Task 6 contains 10 marks**

|  |  |  |
| --- | --- | --- |
| **Criteria** | **Marks** | **Marks obtained by the student for the answer provided** |
| **Out of 10** |
| **Excellent**  Excellent justification for the selected deployment techniques, by critically comparing with other available deployment options. Excellent explanation on selected technique. | **7-10** |  |
| **Good**  Good justification for the selected deployment techniques, by critically comparing with other available deployment options. Proper explanation on selected technique. | **6-7** |  |
| **Satisfactory**  Reasonable justification for the selected deployment techniques, by comparing with other available deployment options. Satisfactory explanation on selected technique. | **4-6** |  |
| **Poor**  Limited justification, limited comparison with poor explanation | **0-4** |  |

|  |  |  |
| --- | --- | --- |
| **Total Marks** | **Out of 100** |  |

# Acknowledgement

First of all, I would like to express my deep gratitude to my parents who advised me to enroll to the Dual HND in Software Engineering + Computing and Systems Development, offered by the ICBT Southern Campus Matara. They were always guiding me to the correct path to achieve my goals.

Also, I would thank to our Windows Application Development lecturer, Mr.Aruna Indika,for giving me guidance to prepare this assignment. His lecturers have been educative and very interesting. The production of this assignment has helped me to expand the knowledge on Object Oriented Programming. As well as improve my skills in research, designing and communication. It would not have been possible for me to produce this assignment in the professional way in which I have done it without his guidance and advice.

In addition, I thank to the ICBT Campus for offering this course in Matara in the first place, which has enabled me as well as many others living in the Southern Province to join it.

I acknowledge with thanks all the authors, experts and other professionals from whose publications I have used for referencing information for this assignment.

# Executive Summary

Total Fitness pharmacy, are the sole agent for Cipla products. So, in this assignment we have to build a system for them to keep the records of their inventory, customers, purchases, sales, and billing.

In the task 1 the learner has to critically compare windows and other platforms and select a suitable platform that suits for the application that the learner is going to build up for this pharmacy.

In the task 2 the learner has to compare and select development tools and technologies for the application. In here the learner has to select a suitable programming language for the application and a suitable IDE for it giving reasons in why the learner selected it.

Next, in task 3 the learner has to design System architecture diagram, UML diagram, Data model design (ER diagram) and Sample UI designs of the application.

Then, in task 04 the learner should develop suitable application based on the designs that he has designed. Also, he should demonstrate and provide all source codes and use proper coding standards. The learner should focus on reusability and maintainability of the application.

Next in task 5, the learner has to select appropriate test techniques. Also he should, Test and provide proper test cases and critically evaluate the test result.

Finally, in task 6, the learner has to explain deployment techniques that are selected and critically evaluate suitability of the selection.

# Table of Contents

[Acknowledgement i](#_Toc474611832)

[Executive Summary ii](#_Toc474611833)

[Table of Contents iii](#_Toc474611834)

[List of Figures v](#_Toc474611835)

[List of Tables viii](#_Toc474611836)

[1. Introduction 1](#_Toc474611837)

[2. Task 01 2](#_Toc474611838)

[2.1. Comparing different windows or other platforms and select the suitable platform for different user groups. 2](#_Toc474611841)

[3. Task 2 4](#_Toc474611842)

[3.1. Compare and select development tools and technologies for the above mention application 4](#_Toc474611846)

[4. Task 3 6](#_Toc474611847)

[4.1. Entity Relationship Diagram 6](#_Toc474611849)

[4.2. Wireframe Design 7](#_Toc474611850)

[5. Task 04 13](#_Toc474611851)

[5.1. Developing a suitable application for Total Fitness Pharmacy system based on the design. 13](#_Toc474611853)

[1. Connection Class File 13](#_Toc474611854)

[2. Login 14](#_Toc474611855)

[3. Administrator Main 17](#_Toc474611856)

[4. Pharmacy Main 27](#_Toc474611857)

[5. User Manual 65](#_Toc474611858)

[6. Task 05 74](#_Toc474611859)

[6.1. System Testing 74](#_Toc474611861)

[6.2. Test Case 76](#_Toc474611862)

[7. Task 06 84](#_Toc474611863)

[7.1. Deployment Techniques 84](#_Toc474611865)

[Conclusion 89](#_Toc474611866)

[Recommendation 90](#_Toc474611867)

[Gantt Chart 91](#_Toc474611868)

[References 92](#_Toc474611869)

# List of Figures

[Figure 1 - ERD 6](file:///J:\C%23%20Assignment%20-%20Avindi1.docx#_Toc474611449)

[Figure 2 - User Configuration Wireframe 7](#_Toc474611450)

[Figure 3 - Stock Wireframe 8](file:///J:\C%23%20Assignment%20-%20Avindi1.docx#_Toc474611451)

[Figure 4 - Pharmacy Dealers Wireframe 9](file:///J:\C%23%20Assignment%20-%20Avindi1.docx#_Toc474611452)

[Figure 5 - Suppliers Wireframe 9](file:///J:\C%23%20Assignment%20-%20Avindi1.docx#_Toc474611453)

[Figure 6 - Distribution Details Wireframe 10](file:///J:\C%23%20Assignment%20-%20Avindi1.docx#_Toc474611454)

[Figure 7 - Customer Registration Wireframe 11](#_Toc474611455)

[Figure 8 - View Bills Wireframe 12](file:///J:\C%23%20Assignment%20-%20Avindi1.docx#_Toc474611456)

[Figure 9 - Billing Wireframe 12](file:///J:\C%23%20Assignment%20-%20Avindi1.docx#_Toc474611457)

[Figure 10 - Connection Class File 13](file:///J:\C%23%20Assignment%20-%20Avindi1.docx#_Toc474611458)

[Figure 11 - Login 14](file:///J:\C%23%20Assignment%20-%20Avindi1.docx#_Toc474611459)

[Figure 12 - Login public class 14](file:///J:\C%23%20Assignment%20-%20Avindi1.docx#_Toc474611460)

[Figure 13 - Login Button Code 15](file:///J:\C%23%20Assignment%20-%20Avindi1.docx#_Toc474611461)

[Figure 14 - Administrator Main 17](file:///J:\C%23%20Assignment%20-%20Avindi1.docx#_Toc474611462)

[Figure 15 - Date and Time Code 17](file:///J:\C%23%20Assignment%20-%20Avindi1.docx#_Toc474611463)

[Figure 16 - Administrator Main Icon Codes 18](file:///J:\C%23%20Assignment%20-%20Avindi1.docx#_Toc474611464)

[Figure 17 - User Configuration Tab page 19](file:///J:\C%23%20Assignment%20-%20Avindi1.docx#_Toc474611465)

[Figure 18 - usergrid() method 20](file:///J:\C%23%20Assignment%20-%20Avindi1.docx#_Toc474611466)

[Figure 19 - Add User Button 21](file:///J:\C%23%20Assignment%20-%20Avindi1.docx#_Toc474611467)

[Figure 20 - Update User 22](#_Toc474611468)

[Figure 21 - Delete User Button 23](file:///J:\C%23%20Assignment%20-%20Avindi1.docx#_Toc474611469)

[Figure 22 - Fillcustomers() method 24](#_Toc474611470)

[Figure 23 - clear() method 24](file:///J:\C%23%20Assignment%20-%20Avindi1.docx#_Toc474611471)

[Figure 24 - Search User 25](file:///J:\C%23%20Assignment%20-%20Avindi1.docx#_Toc474611472)

[Figure 25 - Data Grid Cell Click Code 26](file:///J:\C%23%20Assignment%20-%20Avindi1.docx#_Toc474611473)

[Figure 26 - Pharmacy Main 27](file:///J:\C%23%20Assignment%20-%20Avindi1.docx#_Toc474611474)

[Figure 27 - Pharmacy Main Icon Codes 28](file:///J:\C%23%20Assignment%20-%20Avindi1.docx#_Toc474611475)

[Figure 28 - Date and Time Code 29](file:///J:\C%23%20Assignment%20-%20Avindi1.docx#_Toc474611476)

[Figure 29 - Stock Main 29](file:///J:\C%23%20Assignment%20-%20Avindi1.docx#_Toc474611477)

[Figure 30 - Stock Grid 30](file:///J:\C%23%20Assignment%20-%20Avindi1.docx#_Toc474611478)

[Figure 31 - Search Stock 31](file:///J:\C%23%20Assignment%20-%20Avindi1.docx#_Toc474611479)

[Figure 32 = Pharmacy Main Load Event 32](file:///J:\C%23%20Assignment%20-%20Avindi1.docx#_Toc474611480)

[Figure 33 - Stock Add Button 33](file:///J:\C%23%20Assignment%20-%20Avindi1.docx#_Toc474611481)

[Figure 34 - Stock Update Button 34](file:///J:\C%23%20Assignment%20-%20Avindi1.docx#_Toc474611482)

[Figure 35 - Stock Delete Button 35](file:///J:\C%23%20Assignment%20-%20Avindi1.docx#_Toc474611483)

[Figure 36 - Customer Registration Main 36](file:///J:\C%23%20Assignment%20-%20Avindi1.docx#_Toc474611484)

[Figure 37 - customergrid() 36](file:///J:\C%23%20Assignment%20-%20Avindi1.docx#_Toc474611485)

[Figure 38 - Customer Add 37](file:///J:\C%23%20Assignment%20-%20Avindi1.docx#_Toc474611486)

[Figure 39 - Customer Update 39](file:///J:\C%23%20Assignment%20-%20Avindi1.docx#_Toc474611487)

[Figure 40 - Customer Delete 40](file:///J:\C%23%20Assignment%20-%20Avindi1.docx#_Toc474611488)

[Figure 41 - Search Customer 42](file:///J:\C%23%20Assignment%20-%20Avindi1.docx#_Toc474611489)

[Figure 42 - dealergrid() 43](file:///J:\C%23%20Assignment%20-%20Avindi1.docx#_Toc474611490)

[Figure 43 - Pharmacy Dealers Main 43](file:///J:\C%23%20Assignment%20-%20Avindi1.docx#_Toc474611491)

[Figure 44 - Add Dealer 44](file:///J:\C%23%20Assignment%20-%20Avindi1.docx#_Toc474611492)

[Figure 45 - Search Dealer 44](file:///J:\C%23%20Assignment%20-%20Avindi1.docx#_Toc474611493)

[Figure 46 - Delete Dealer 45](file:///J:\C%23%20Assignment%20-%20Avindi1.docx#_Toc474611494)

[Figure 47 - Update Dealer 45](file:///J:\C%23%20Assignment%20-%20Avindi1.docx#_Toc474611495)

[Figure 48 - Dealer Data Grid Cell Click 46](file:///J:\C%23%20Assignment%20-%20Avindi1.docx#_Toc474611496)

[Figure 49 - Supplier Main 47](file:///J:\C%23%20Assignment%20-%20Avindi1.docx#_Toc474611497)

[Figure 50 - Supplier datagrid cell click 47](file:///J:\C%23%20Assignment%20-%20Avindi1.docx#_Toc474611498)

[Figure 51 - suppliergrid 47](file:///J:\C%23%20Assignment%20-%20Avindi1.docx#_Toc474611499)

[Figure 52 - Supplier Add 48](file:///J:\C%23%20Assignment%20-%20Avindi1.docx#_Toc474611500)

[Figure 53 - Supplier Delete 49](file:///J:\C%23%20Assignment%20-%20Avindi1.docx#_Toc474611501)

[Figure 54 - Supplier Update 49](file:///J:\C%23%20Assignment%20-%20Avindi1.docx#_Toc474611502)

[Figure 55 - Search Supplier 50](file:///J:\C%23%20Assignment%20-%20Avindi1.docx#_Toc474611503)

[Figure 56 - Distribution Data grid cell click 51](file:///J:\C%23%20Assignment%20-%20Avindi1.docx#_Toc474611504)

[Figure 57 - Distribution Main 51](file:///J:\C%23%20Assignment%20-%20Avindi1.docx#_Toc474611505)

[Figure 58 - Distribution Detail Add 52](file:///J:\C%23%20Assignment%20-%20Avindi1.docx#_Toc474611506)

[Figure 59 - Delete 53](file:///J:\C%23%20Assignment%20-%20Avindi1.docx#_Toc474611507)

[Figure 60 - Update 53](file:///J:\C%23%20Assignment%20-%20Avindi1.docx#_Toc474611508)

[Figure 61 - Search Detail 54](#_Toc474611509)

[Figure 62 - Billing Main 55](file:///J:\C%23%20Assignment%20-%20Avindi1.docx#_Toc474611510)

[Figure 63 - Billing Public Class 55](file:///J:\C%23%20Assignment%20-%20Avindi1.docx#_Toc474611511)

[Figure 64 - billgrid() 56](file:///J:\C%23%20Assignment%20-%20Avindi1.docx#_Toc474611512)

[Figure 65 - searchcustomer() and Fillcustomer() 57](#_Toc474611513)

[Figure 66 - amountcalc() 59](file:///J:\C%23%20Assignment%20-%20Avindi1.docx#_Toc474611514)

[Figure 67 - Save Bil 63](file:///J:\C%23%20Assignment%20-%20Avindi1.docx#_Toc474611515)

[Figure 68 - Update Bill 63](file:///J:\C%23%20Assignment%20-%20Avindi1.docx#_Toc474611516)

[Figure 69 - View Bill 64](file:///J:\C%23%20Assignment%20-%20Avindi1.docx#_Toc474611517)

[Figure 70 - Delete 64](file:///J:\C%23%20Assignment%20-%20Avindi1.docx#_Toc474611518)

# List of Tables

[Table 1 - Test Case 76](#_Toc474611745)

[Table 2 - Test Case 77](#_Toc474611746)

[Table 3 - Direct Advantages and Disadvanatges 85](file:///J:\C%23%20Assignment%20-%20Avindi1.docx#_Toc474611747)

[Table 4 - Parallel Advantages and Disadvantages 86](file:///J:\C%23%20Assignment%20-%20Avindi1.docx#_Toc474611748)

[Table 5 - Pilot Advantages and Disadvanatges 86](file:///J:\C%23%20Assignment%20-%20Avindi1.docx#_Toc474611749)

[Table 6- Phased 87](file:///J:\C%23%20Assignment%20-%20Avindi1.docx#_Toc474611750)

# Introduction

C#.net is an Object Oriented Programming Language developed by the Microsoft Corporation within its .Net framework. It’s a hybrid of C and C++ languages. This language was developed to beat Java language. .Net is a software development framework developed by Microsoft Corporation to run applications which are having .Net installed. It provides a controlled programming environment where software can be developed. Programming languages in Visual Studio.net always based on this framework like C#.net. So, there are many advantages in C#.net. It’s a modern language with many tools and techniques. Also it supports for Object Oriented Programming. We can build not only desktop applications, but also mobile applications like Android and iOS.

So, in here we have to build up a system for the “**Total Fitness**” the largest and very busy pharmacy in Matara, to store records of their daily business activities with the customers and suppliers. They are the sole agent for the Cipla product in the region. So, the learner has to build up a system to keep records of their customers, suppliers, stocks, dealers, all purchasing order invoices, purchasing bills, customer bills. Before doing that the learner should compare windows and other platforms and select a suitable platform for the application. Also, he should select a suitable programming language and an IDE to develop the system.

# Task 01



## Comparing different windows or other platforms and select the suitable platform for different user groups.

In this task, we are going to compare different windows or other platforms and select the suitable platform for different user groups for the **Total Fitness** system. A computing platform is a hardware or software needed to start an application. It consists of hardware or operating system. In this case, here we say platforms as operating systems. We have to compare windows and other platforms and select the suitable platform for different user groups. So we have selected some Operating Systems and then we would compare each of them based on the application we are going to build on for the Total Fitness Pharmacy System.

* Microsoft Windows
* MacOs
* Linux
* **Microsoft Windows**

First let’s see, about the advantages of Windows OS. Microsoft Windows developed by Microsoft Corporation is the most famous operating systems in the world. This operating system can be installed on any device PC, laptop, and tablets or even on phones. Also, is very easy to use for any user if the users have used it before. It is quite cheap than the Mac OS. Because there is wide range of software’s many people use this operating system than the Mac OS. Windows Operating System has a good eye catching interface than the Mac OS.

* **Mac OS (Macintosh Operating System)**

Mac OS was developed by the Apple Inc. It works only on Mac OS devices and its quite expensive than the Windows Operating System. Also there are only very few viruses and malware attacks in this operating system rather than windows. Also there are very few software in Mac OS. But this operating system isn’t much friendlier to many users because only few people use this

* **Linux Operating System**

Linux operating system is known as the most secure operating system. As it is an open source one, there are only very few virus attacks. Also, this is free of charge. This OS was first released by Linus Torvalds. Linux is not as easy to use as [Windows](http://www.online-sciences.com/technology/microsoft-windows-advantages-and-disadvantages/)or[Mac](http://www.online-sciences.com/technology/macintosh-operating-system-advantages-and-disadvantages/) , It requires a knowledge about computing than other operating systems, It can be very challenging for a beginning user. Also its quite hard to find software for Linux because it doesn’t dominate the market like Windows or Mac OS. This OS is often by the users with more technical knowledge and background. If you are handling a big software this is good.

So, we discussed about three operating systems and its advantages and disadvantages. Now we have to choose a suitable operating system for our application. When we consider about Windows OS, it’s the most famous operating system among users. Also, it’s very easy to use rather than Mac OS and Linux. It can be installed on any pc and large number of software can be found. But windows get virus and other malware attacks very often and sometimes it gets stuck.

When we look at Mac OS it has less virus and other malware attacks. But it’s too expensive and it can be installed only on Mac devices. It’s not much popular among users.

Then, if we look at Linux OS, it’s the secure operating system because it’s an open source OS. But it’s quite technical and this OS can be used for users who have quite a computer knowledge. Also large number of software cannot be found on Linux. After considering these points, the learner selects Microsoft Windows as the suitable platform for this application. It is because its user friendly, easy to use, and it offers a large number of software.

# Task 2



## Compare and select development tools and technologies for the above mention application

In this section, we have to compare some development tools and technologies for the application that we are going to build for the Total Fitness Pharmacy. As this system is for a pharmacy management system, the users would be pharmacist. So, there will be no users with computer knowledge to use this system. Therefore, this system should be more user-friendly which make pharmacists easy to use the system. To build up a system we should first select a programming language. There are many languages like Java, C++, C#.net, Python etc... But, among these programming languages the learner chooses c#.net to build up this system. The reasons are

* It’s facilitates with many methods for variables
* Microsoft Corporation developed C#.net. Therefore, it is very famous for building desktop applications, mobile applications, REST APIS, websites, games and even native Android or iOS apps. As we have selected Microsoft Windows as the suitable platform, the application will run more well if we use C#.net.
* In addition, C#.net uses the .net framework too provides hundreds of libraries for building websites, implementing security, working with the file system, etc.

Next, we have to choose a suitable IDE to develop the system. An Integrated Development Environment is a software application that helps programmers to design and develop software. Therefore, in here the learner chooses Visual Studio as the best IDE for the C#.net.

* Visual Studio is very user friendly for developers. It has a good interface and it helps for the developer while coding providing suggestions, declared variables, classes, methods, etc. in a pop up menu.
* It consists of a variety of tools, which can be more helpful in designing the interface of the system. Tools like error provider, date time picker, timer are some significant tools that you cannot find even in java.
* There is a solution explorer and class view to view and open the forms and classes.
* Also, this IDE has debugging tools and compiling which the developer can easily test each part of the system easily.

C# language uses .net framework rather than any other languages. The latest .net framework is .net 4.5. This framework provides an environment to run any kind of .net based applications.

# Task 3



## Entity Relationship Diagram

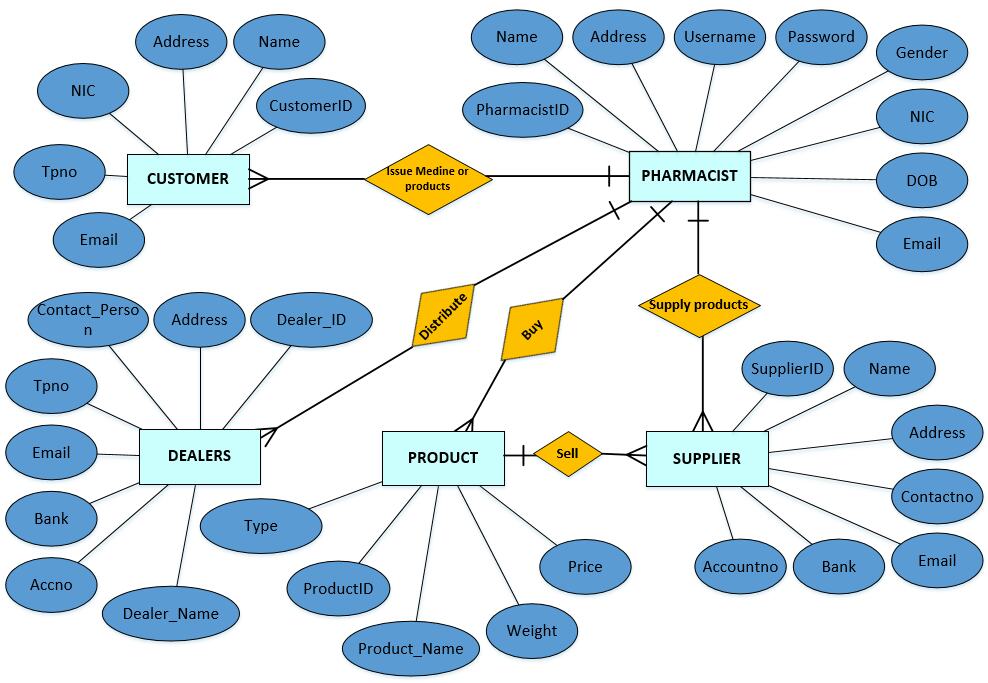


Figure 1 - ERD

## Wireframe Design

****

Figure 2 - User Configuration Wireframe

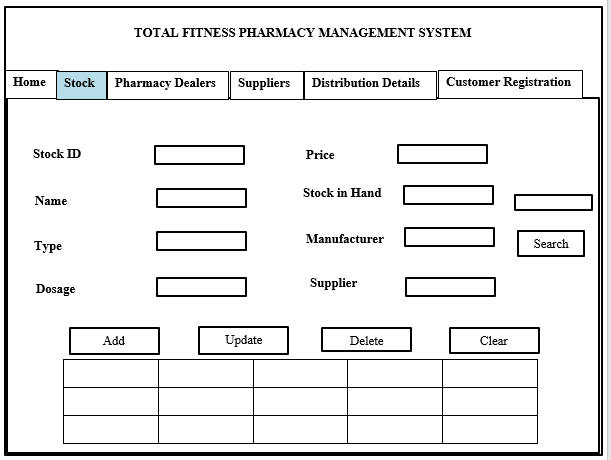


Figure 3 - Stock Wireframe

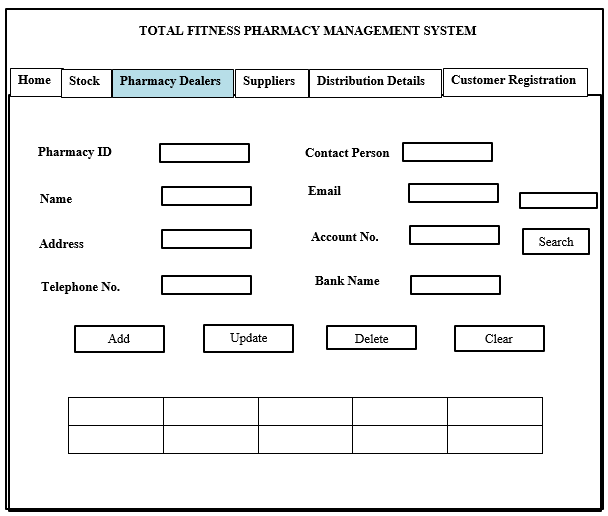
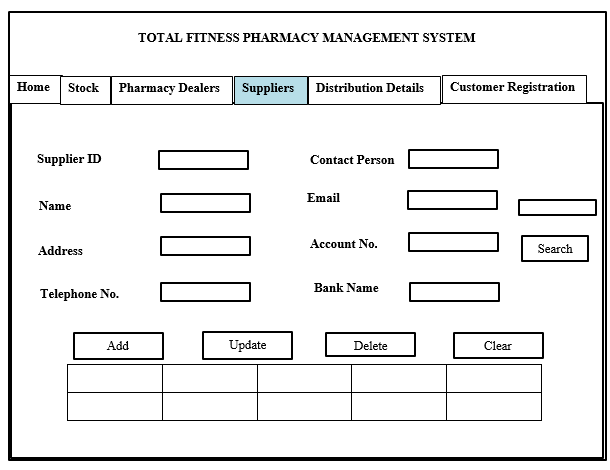


Figure 4 - Pharmacy Dealers Wireframe

Figure 5 - Suppliers Wireframe



Figure - Distribution Details Wireframe

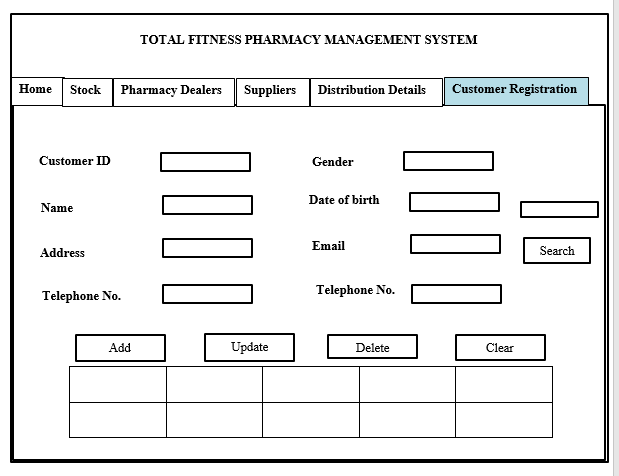


Figure 7 - Customer Registration Wireframe

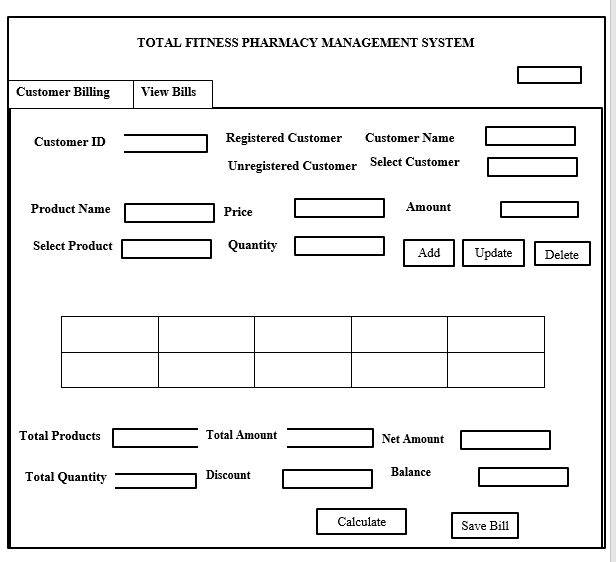
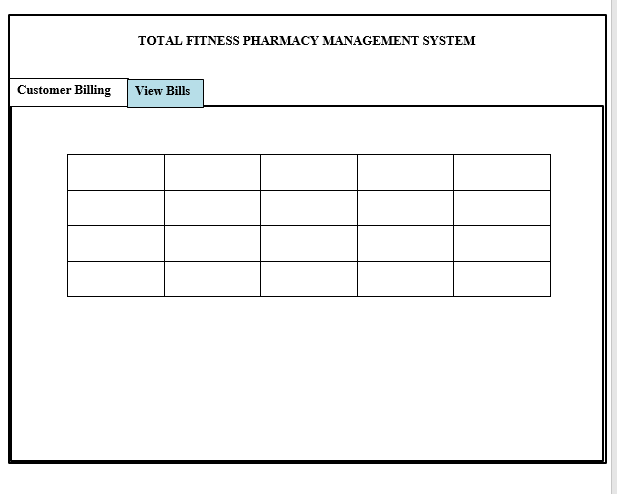


Figure 8 - View Bills Wireframe

Figure 9 - Billing Wireframe

# Task 04



## Developing a suitable application for Total Fitness Pharmacy system based on the design.

In this task the learner is going to develop a suitable application for the Total Fitness Pharmacy. This application has been created using C#.net programming language which we decided to select as the suitable programming language for this. Visual Studio is the IDE which was used to develop the application.

### Connection Class File

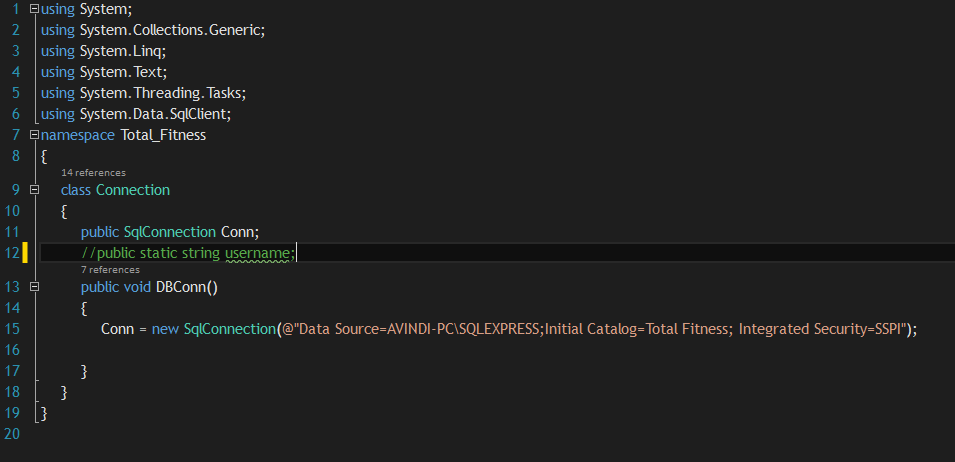
To create a proper system, we should connect our system to a proper database to store records of the customers, suppliers, dealers, etc. Below is a screenshot of a class file named “Connection” which we used here to connect the system to the SQL Database. We have given the sever name and the database name too in the class to connect.

Figure 10 - Connection Class File

### Login

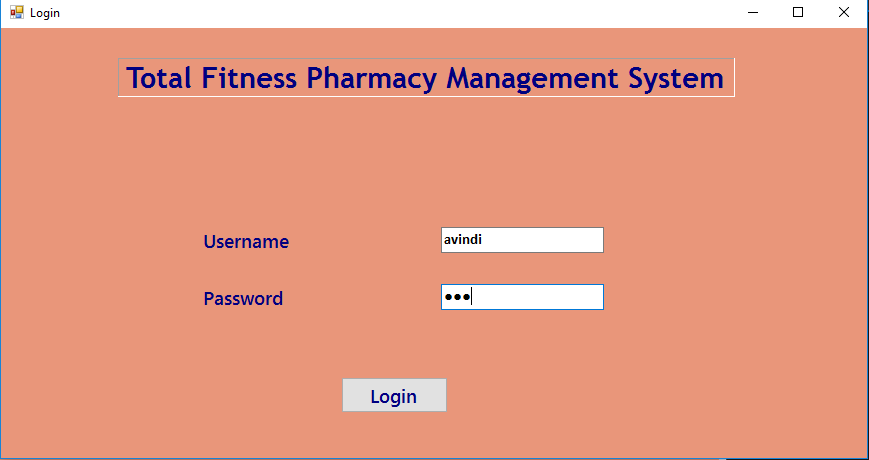


Figure 11 - Login

The above screenshot is the login of the Total Fitness Pharmacy Management System. In here there are two user types namely Administrator and Pharmacist. So, whoever the user administrator or pharmacist logs into the system, the user should give the correct username and password to log into the system. After the user gives the correct login details, the system automatically takes the user to the correct main window. Let’s see below how it works

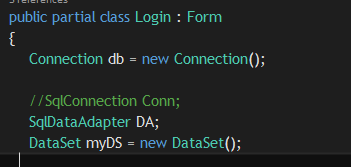


Figure 12 - Login public class

Above screenshot is the public class of the Login Form. In here, we have created an object instance to call the Connection class to establish the connection between the application and database. Also we have created a **SqlDataAdapter** and a **DataSet** named **DS** and **myDS** for the login purposes.

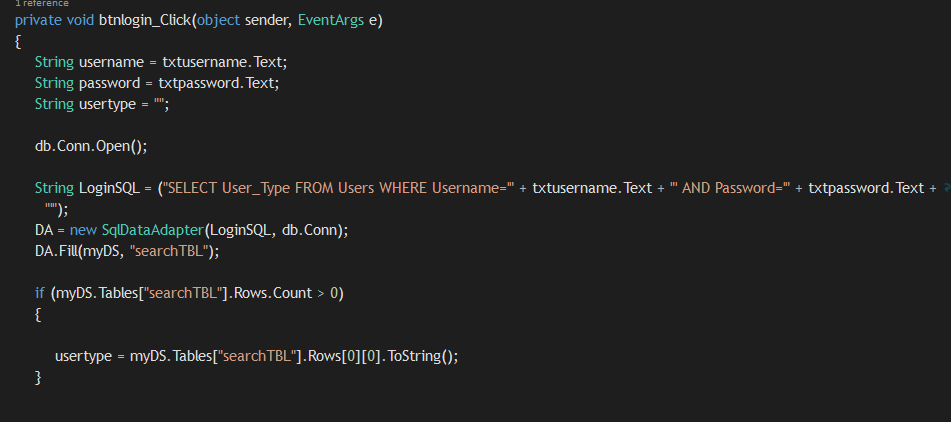
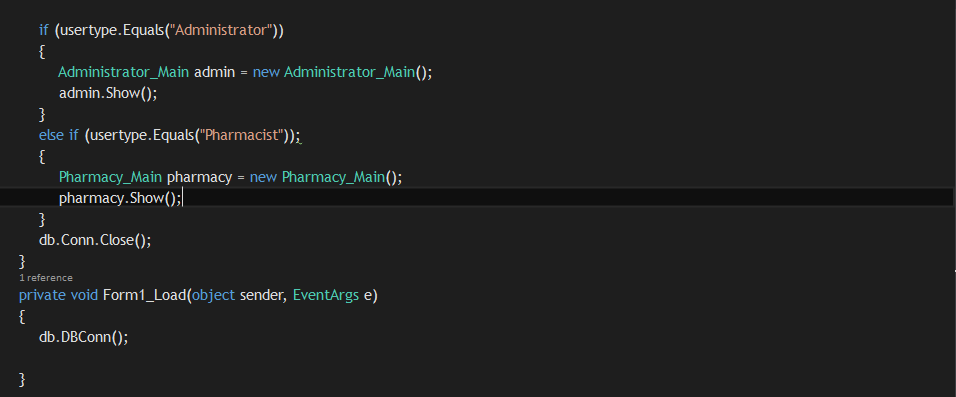


Figure 13 - Login Button Code

This the screenshot of the code written for the login form.**”db.Conn.Open()”** is private method we used in the Connection class. It includes the connection to establish with the application and the server. So we call it to establish the connection here. Then we write a SQL statement in the **LoginSQL** variable to check in the Users table whether the texts given by the user in the username and password textboxes are equal to the values given in the table. Then these values will be filled to a data set named **“DA”** After that using an IF function the system checks in the table’s User Type column whether the user’s type is Administrator or Pharmacist. If the user is **“Administrator”** the system will load up the Administrator Main window or if the user is pharmacist the system will load the pharmacy main window.

### Administrator Main

Figure 14 - Administrator Main

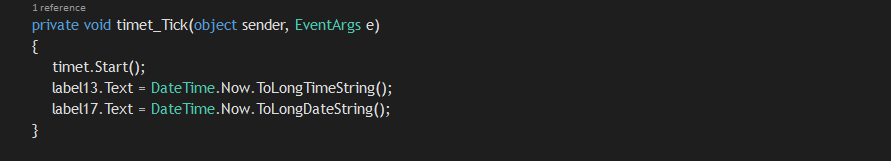
Above image shows the Administrator Main of the Total Fitness system. First the user will see the home of the main window. There are two icons namely **“User Configuration”** and “**Go to the Pharmacy Main”.** The administrator has to select any option of it to proceed. **User Configuration** is where the administrator can manage users in the system. **Go to the Pharmacy Main takes** the administrator to the pharmacy main.

Figure 15 - Date and Time Code

To display the time and date, we have used timer tool provided by the Visual Studio IDE. **timet\_tick** includes its code. First, the timer starts and then it sets the time to long format in the **label13**, and set the date to long format to **lable17**. That’s how you can see the date and time at Administrator main home window. Next will see the buttons **User Configuration** and **Go to Pharmacy Main.**

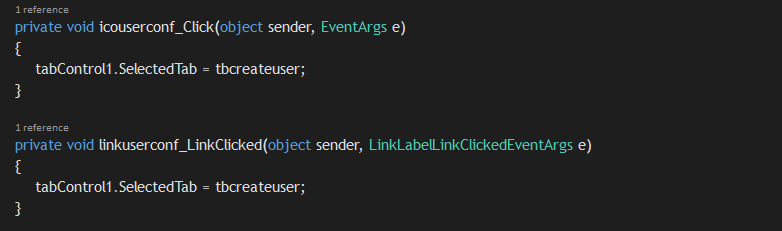
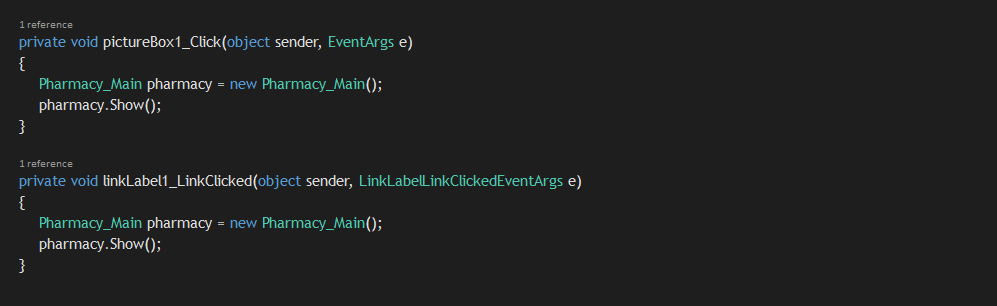


Figure 16 - Administrator Main Icon Codes

For this we have used a picture box and a Link label, so that it looks like an icon and we gave links to both picture box and the link label to open the relevant forms**. icouserconf** is the picture box and **linkuserconf** is the link label which opens the User Configuration tab page when clicked. **pictureBox1\_Click** and **linkLable1** are also picture box and a link label which opens the pharmacy main window. To open it we have created an object instance of pharmacy main and make it open.

We have used tab controls in the Administrator main window. It includes the User Configuration tab page. User can click on its tab page or either click on the User Configuration icon to open it. Below you can see the image of the User Configuration tab page. User can add, update and delete users in this page.

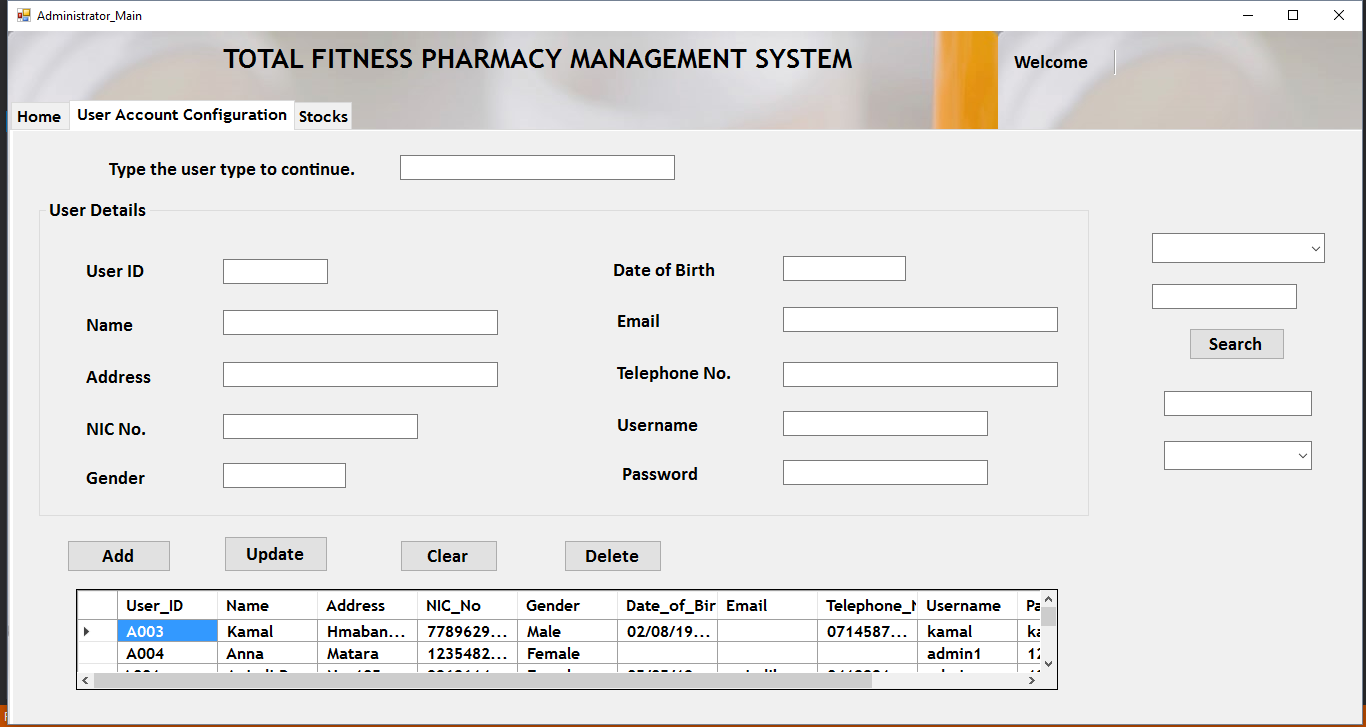


Figure 17 - User Configuration Tab page

This is the **usegrid** private method which was used to show the users in the Users table that you can see in the **User configuration main,** using a data grid tool. A SQL statement has been written here to select all the users in the Users table in the database. Then the details will be added into the dataset. Then the dataset will display the details.

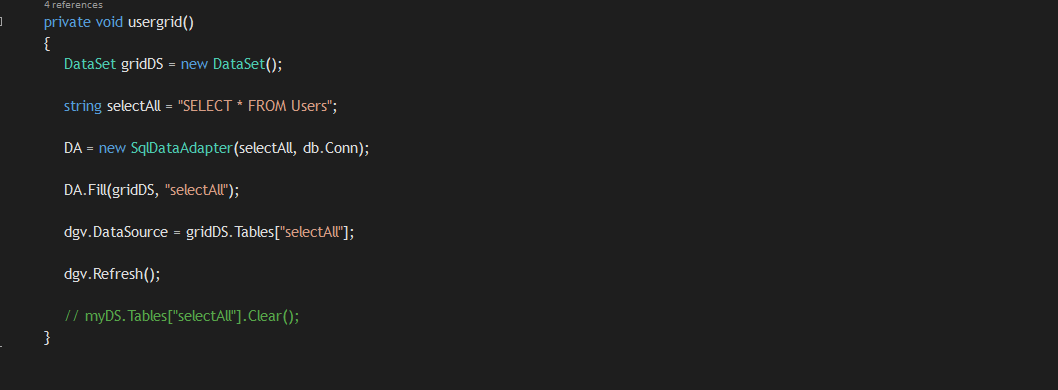


Figure 18 - usergrid() method

Below image is the code given for the Add button to add new users into the system. New users are inserted to the Users table, in the database. First using an IF function the system checks whether there are any connections current state is open or not. If it’s open it will close. Then using another IF function it checks whether the User ID text box and NIC NO text box are empty. If it is true, the system will display an error message as **“You are not allowed to add a new record without User ID and Name”. Else it the** system will establish the connection by calling the **Conn** method in the Connection class. After that a SQL statement is written to insert into the Users table, the values given by the user to the text boxes given in the interface.

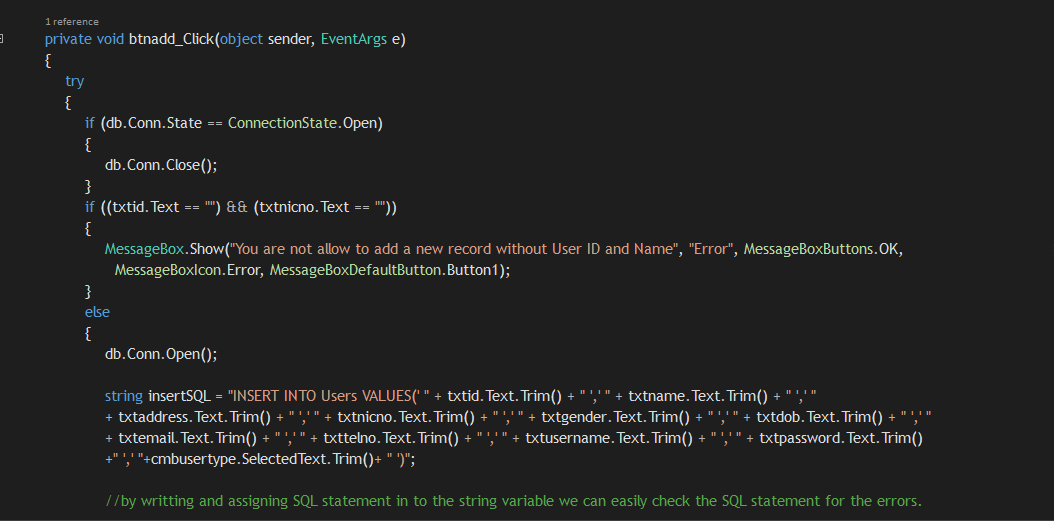
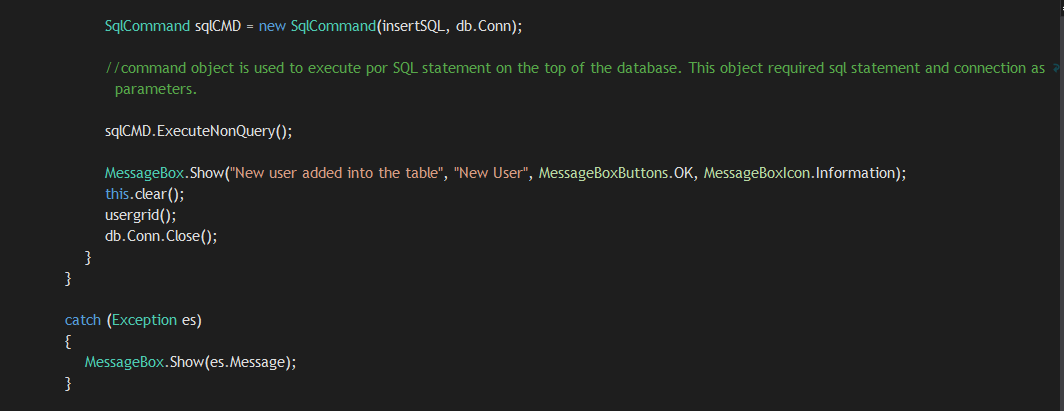


Figure 19 - Add User Button

Then, using **SqlCommand** command object it executes the SQL statement given. If the data insertion was successful it will display a message as **“New User Added into the table”**. Then the there’s another method named **clear()** to clear the text boxes if the insertion of data is successful and refreshers the table to show the new user has been added to the table. After that it closes the connection. All these codes are written inside a try catch block to handle exceptions.

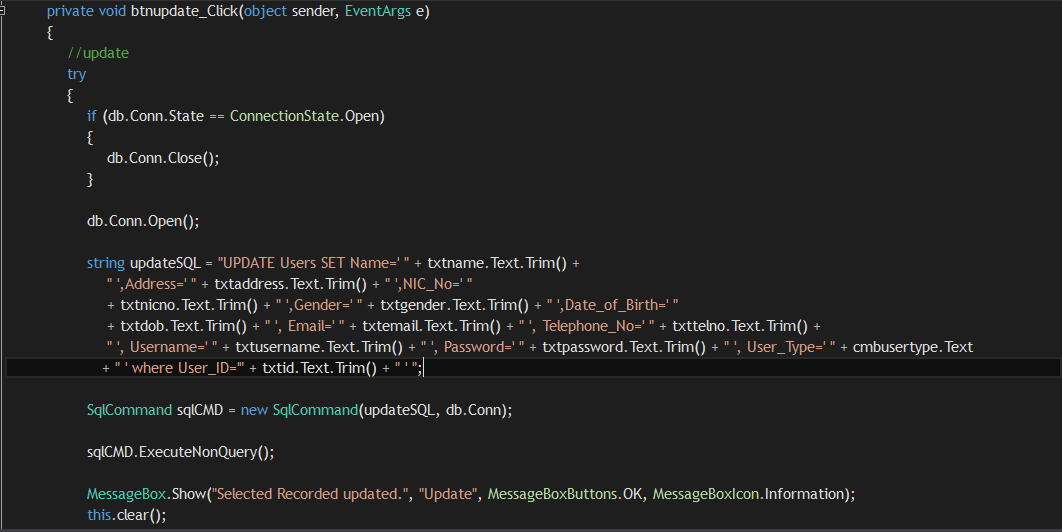


Figure 20 - Update User

Above picture is the code written when administrator updates user details. To update a user, administrator can select a user from the table or by searching a user. Then the text will be set to the text boxes and then administrator has to update it and click update button. Like in Add button first it checks whether are there any connections whose state is open and if so to close them. Then, an SQL statement is written to update the Users table by setting the text given in the text boxes by the user. **updateSQL** variable includes the SQL statement.

Then the **SqlCommand** executes the statement and shows a message **“Selected Record Updated”** if the update is successful. Also, you will see the updated records in the table

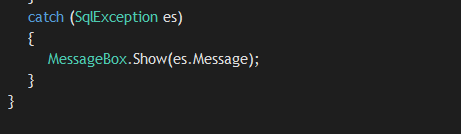
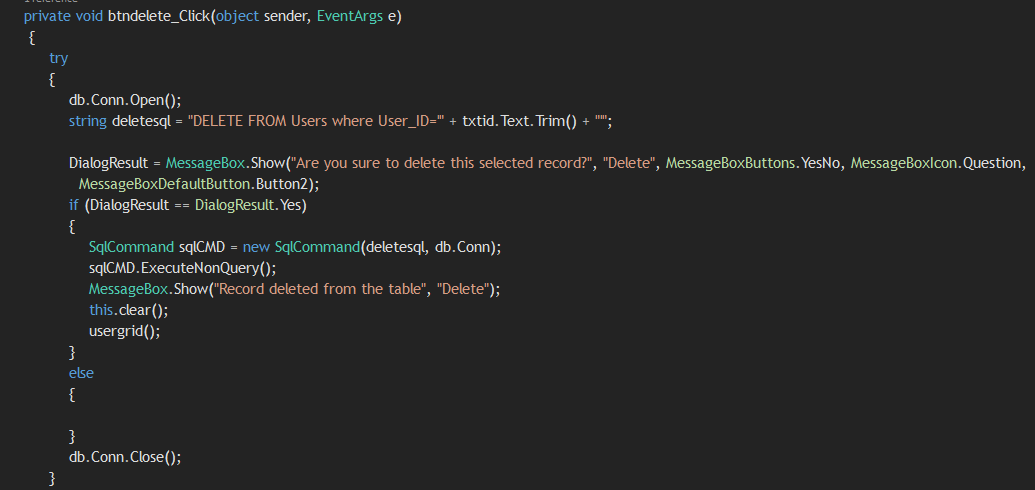


Figure 21 - Delete User Button

Above image is the code given for the delete button, which is used to delete users in the table. To delete a user, administrator can select a user from the table or search a user and delete it. In here too an SQL statement has been written to delete the user where the User ID is equal to the ID displayed in the User ID text box. Then using a **DialogResult** again a message is displayed requesting the user whether to delete this record or not. If the yes button is clicked, the **SqlCommand** starts to execute the statement. If it is successful, it will show a message as **“Record deleted from the table”** and you will see the details gets clear from the text boxes. Also you will see the record has been removed from the table. A try catch block has been used here to handle the exceptions.

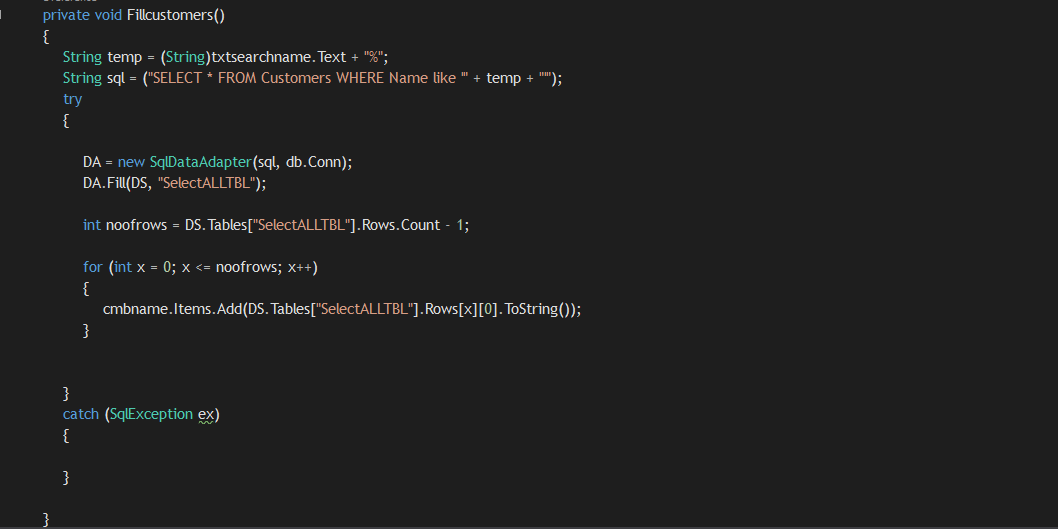


Figure 22 - Fillcustomers() method

This the **Fillcustomers()** method which was used to fill customer ID numbers to the combo box to search. In here a variable has been declared to store the text the user gives to search by name in the search box. Then a SQL statement has been written to select all from the customers table



Figure 23 - clear() method

This is the method used to clear the text the text in the text boxes.

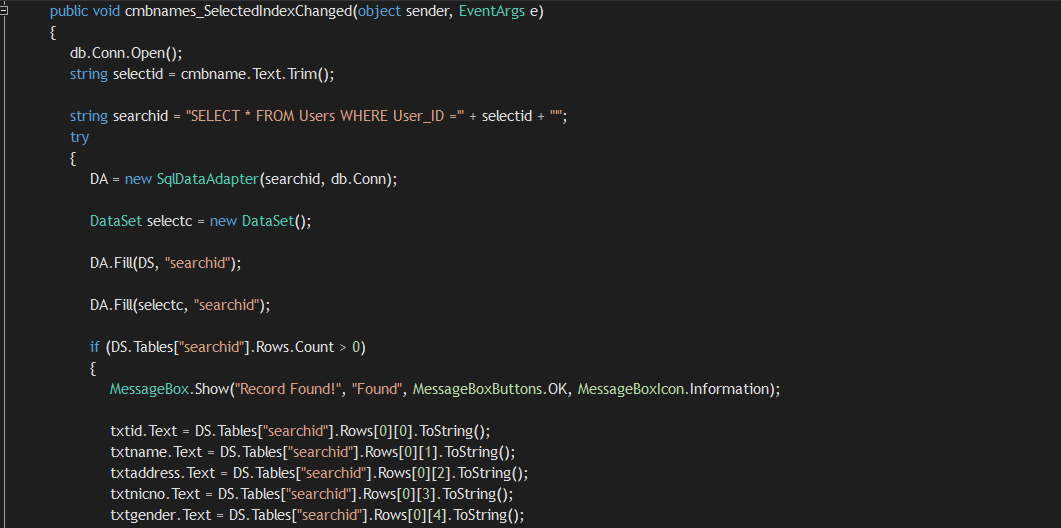
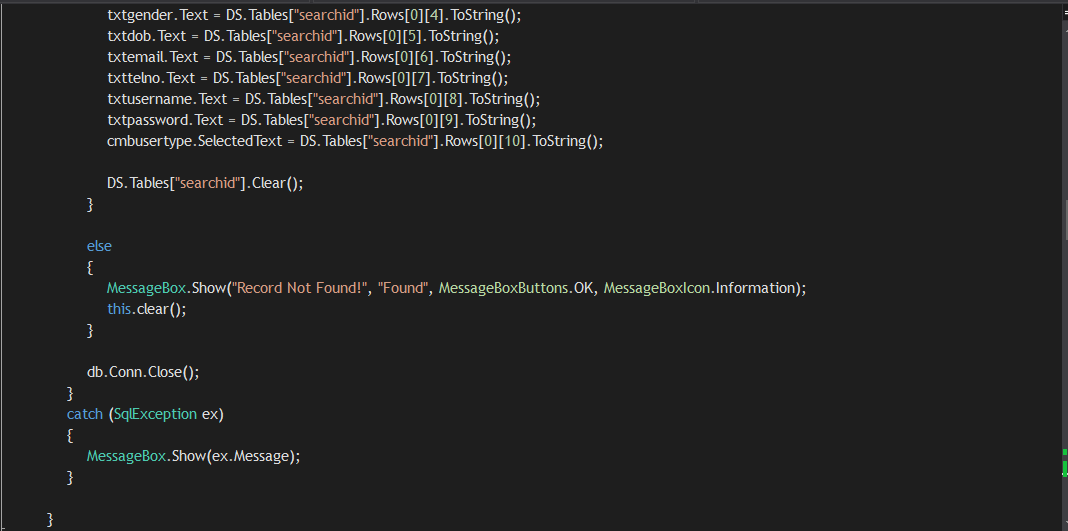


Figure 24 - Search User

This is the code which for searching users using ID’s in the user configuration. First the user types the ID in a text box to search. Then, the string value will be stored inside a variable named **selectid**. Then a SQL statement is written to select the user where the User ID is equal to the ID given to the variable **selectid.** Then the statement executes and the results are first stored inside a dataset. It will show a message if the record is found or else it displays a message that record cannot be found. After that the records in the data set will be displayed in the text box, by setting the text in the relevant cells to the text box.

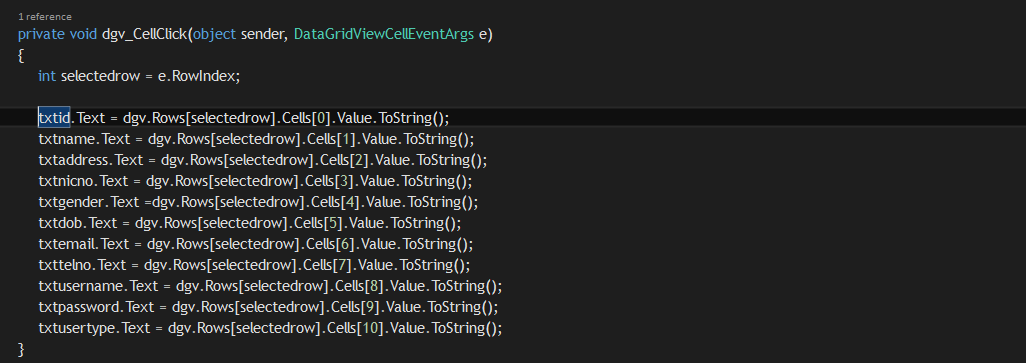


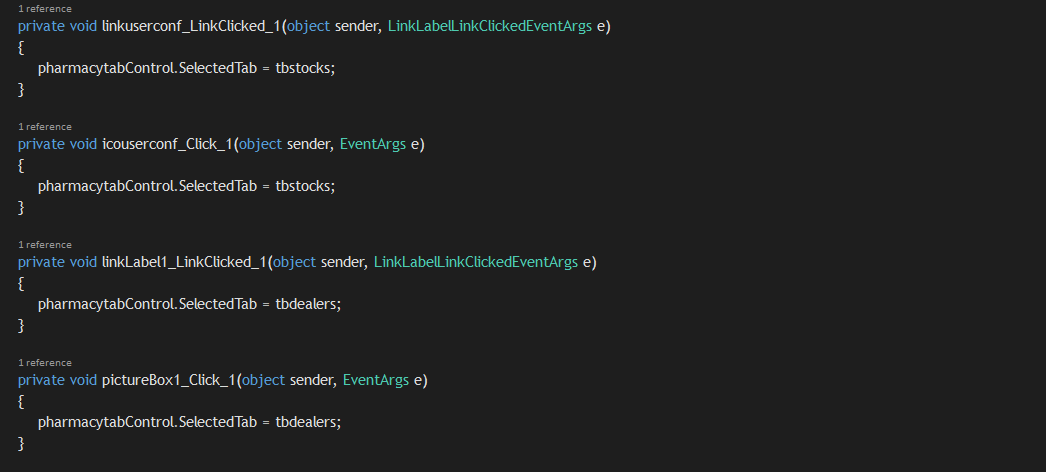
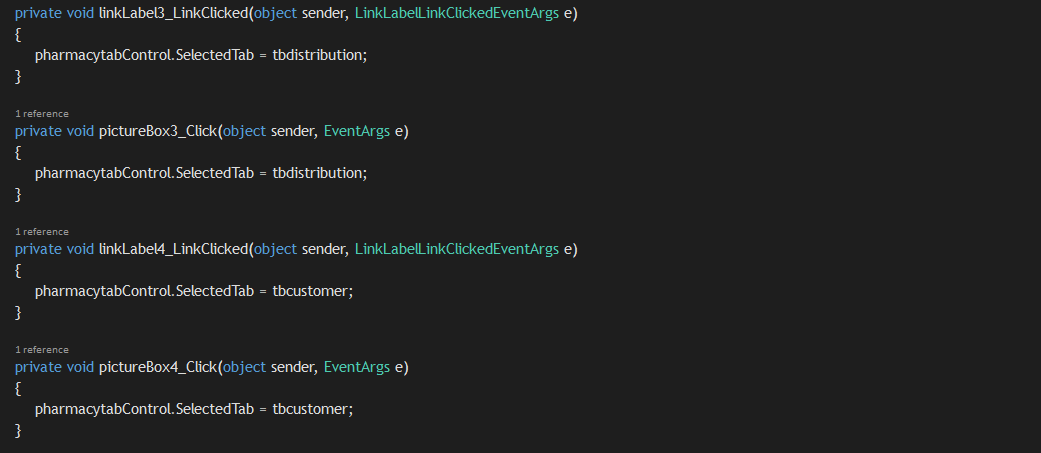
Figure 25 - Data Grid Cell Click Code

The above code includes the code which was written to set the values of the table, to the text boxes when the user selects a row from the table.

### Pharmacy Main

Figure 26 - Pharmacy Main

Above picture shows the pharmacy main window. In here pharmacists can keep records of the stocks, pharmacy dealers, suppliers, distribution details, customers and billing process. Like the Administrator Main there is a date and time. Using tab controls pharmacists can go to each section in one main window. So let’s see how each components work.



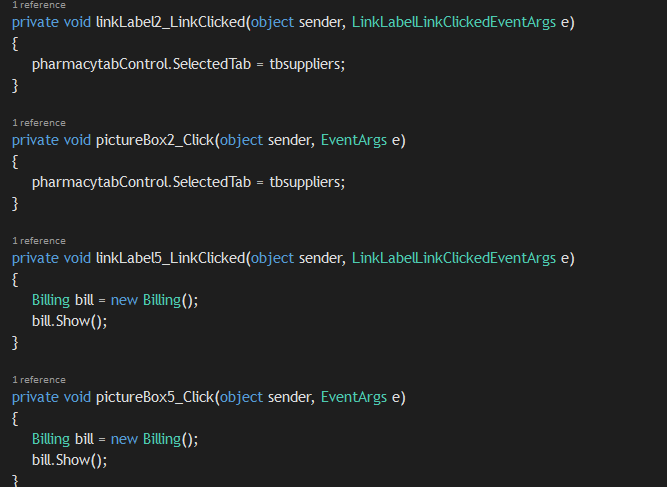


Figure 27 - Pharmacy Main Icon Codes

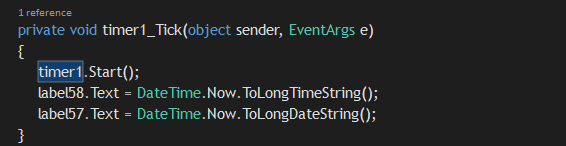
Above picture shows the codes written for the icons that you see in the pharmacy main window. In here we have used a picture box and link label to link each tab page to the icon and also the billing form. When clicked the label or the picture the application directs the user to the relevant tab page or form by the help of these codes.

Figure 28 - Date and Time Code

This is the code written to display the date and time using the Timer tool. First the timer starts to tick then the date and time are set to the relevant tables.

Above picture shows the tab page of stocks. In here users can add new stocks, update, delete and view them too. So les see how this window has been made.

Figure 29 - Stock Main

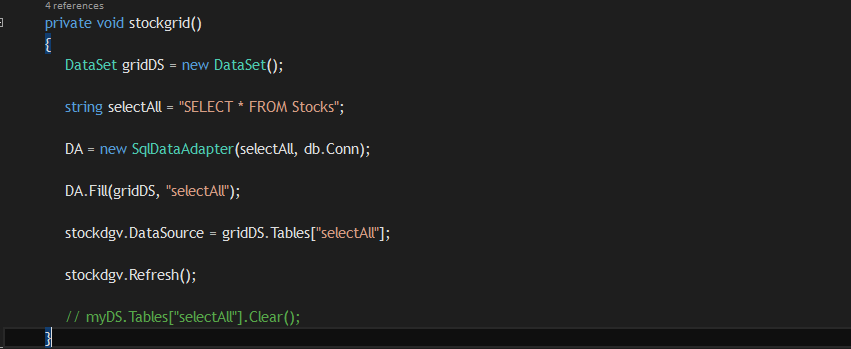


Figure 30 - Stock Grid

**stockgrid()** is a private method which was used to fill the data of the Stock table in the database to the data grid table in the application. Using a SQL statement, we have displayed all the records of the stock table to the stock tab page window. So the users can see the records and view them.

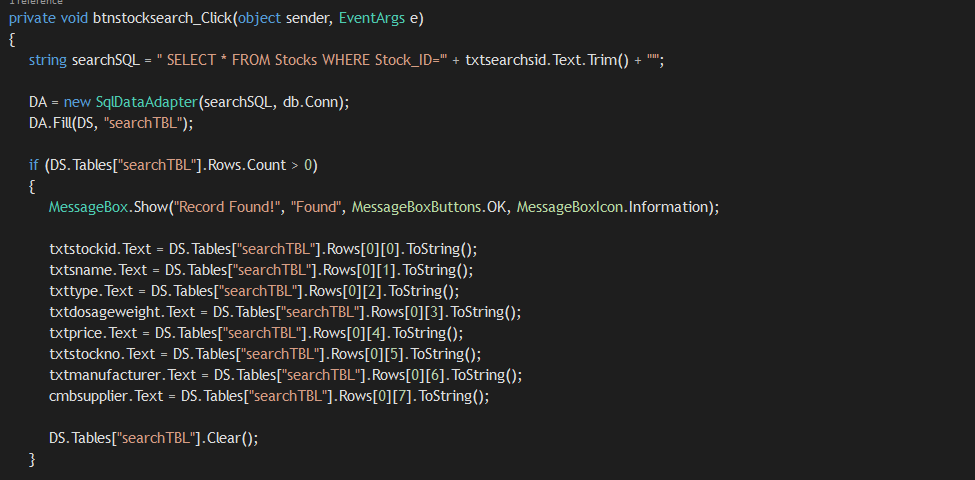
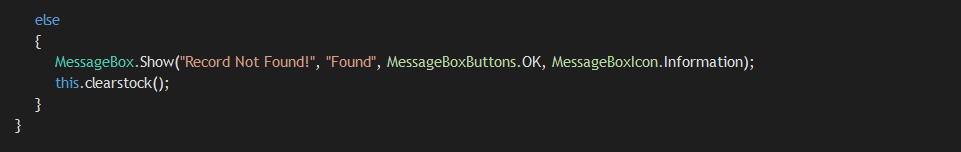


Figure 31 - Search Stock

This the code written for the search button to search stocks in the table. In here above the search button there’s a text box to enter the Stock ID. After the user enters the stock ID and click **Search** button, the Stock ID will be stored in a variable first. Then using a SQL statement, we ask to select the record where the Stock ID is equal to the ID given in the variable. When the record is found it will display a message as **“Record Found”** and the details are first loaded into a dataset. It’s like a virtual table. Then one by one each value in the dataset are set to display in text boxes for the user to see it.

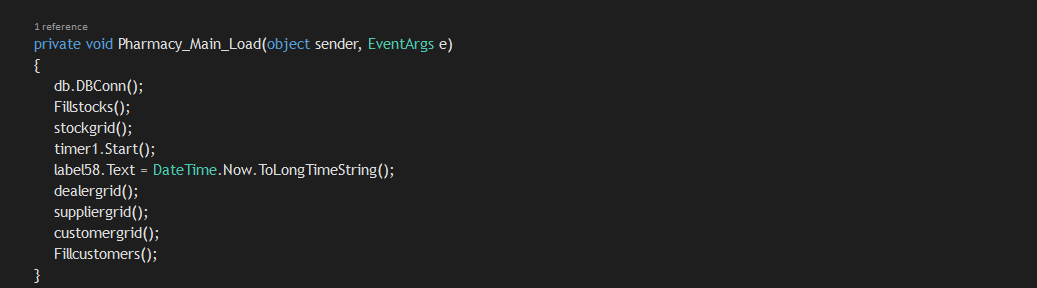
This the Pharmacy Main load event. In here the methods are called to start functioning when the Pharmacy Main loads up. You can see there are some methods which have been called to start its operation.

Figure 32 = Pharmacy Main Load Event

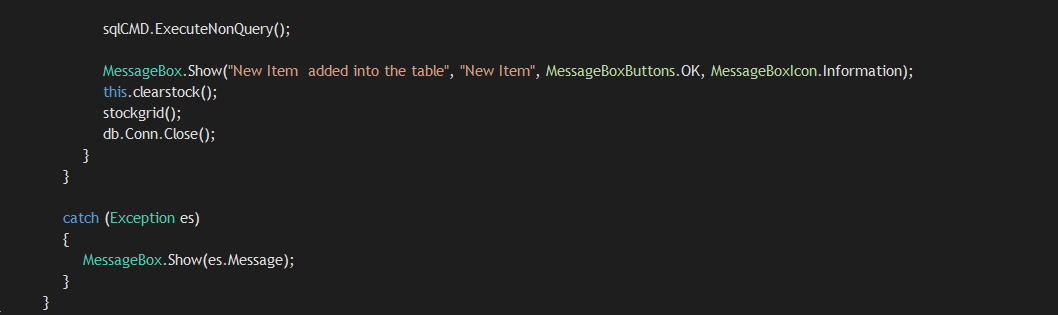
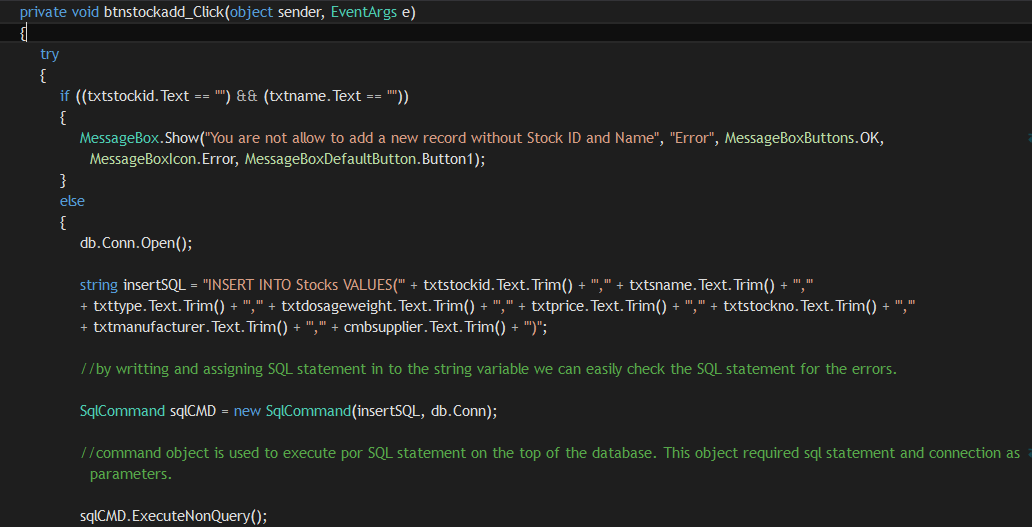


Figure 33 - Stock Add Button

This the code written for the Add button to add new stock details in the database. First it has been coded to check whether the User has entered the User ID and Name to proceed. If it is so it will start to do the next steps or else, it will display a message asking the user to enter them. After that the connection starts to open and a SQL statement has been written to insert the values given in the text boxes to the Stock table. After that the **SqlCommand** executes the query and a message will be displayed as **“New Record Added into the table”**. You will see the new record in the table then. The code is written inside a try catch block to handle exceptions.

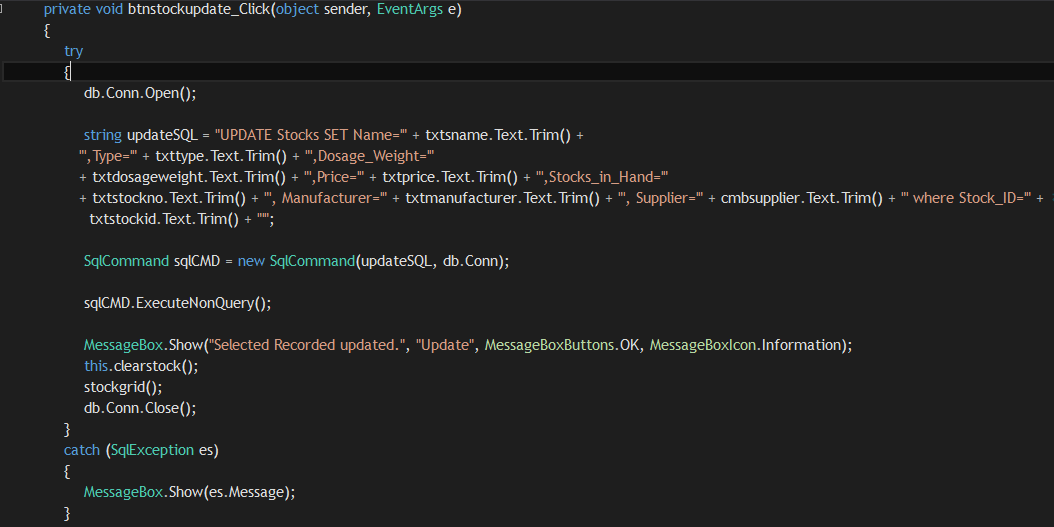


Figure 34 - Stock Update Button

This the code written for the Update button to update stock details in the table. To update a record first, the user, have to select a record from the table or by searching a record. Then the user has to modify the relevant details and click Update button to update. A SQL statement has been written to Update the records by setting the values given in the text boxes. Then the **SqlCommand** executes the query and a message will be displayed as, **“Selected Record Updated”**. The updated details will be shown in the table.

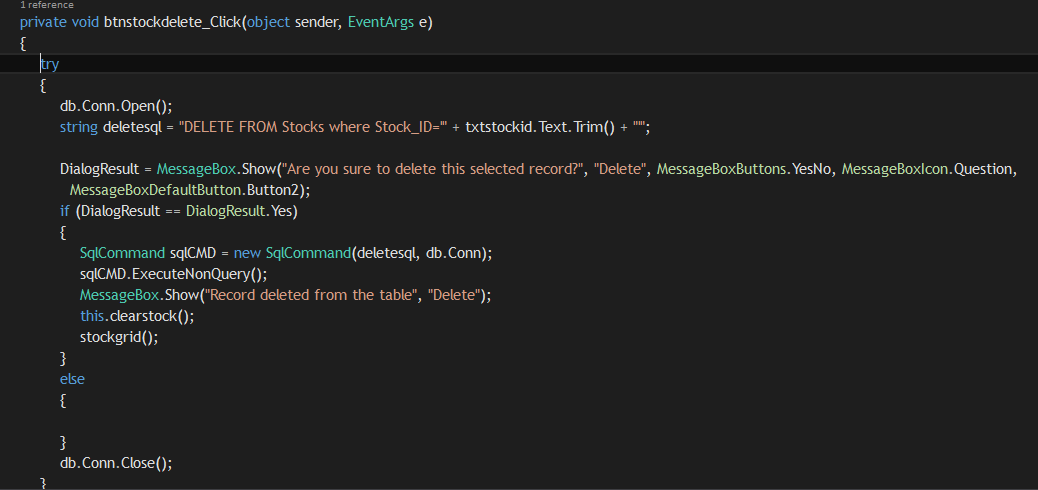
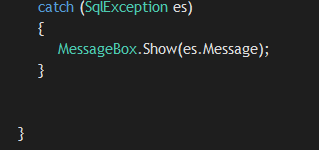


Figure 35 - Stock Delete Button

This is the code for the Delete button. In here a SQL statement has been written to delete the record where the Stock ID is equal to the ID given in text box. After that Using a **DialogResult** we display a message **“Are you sure to delete the selected record**” If the user clicks yes, the record will be deleted and a message will be displayed again “**Record Deleted from the table**”. Also the item will be removed from the data grid table also to make sure the it has been deleted.

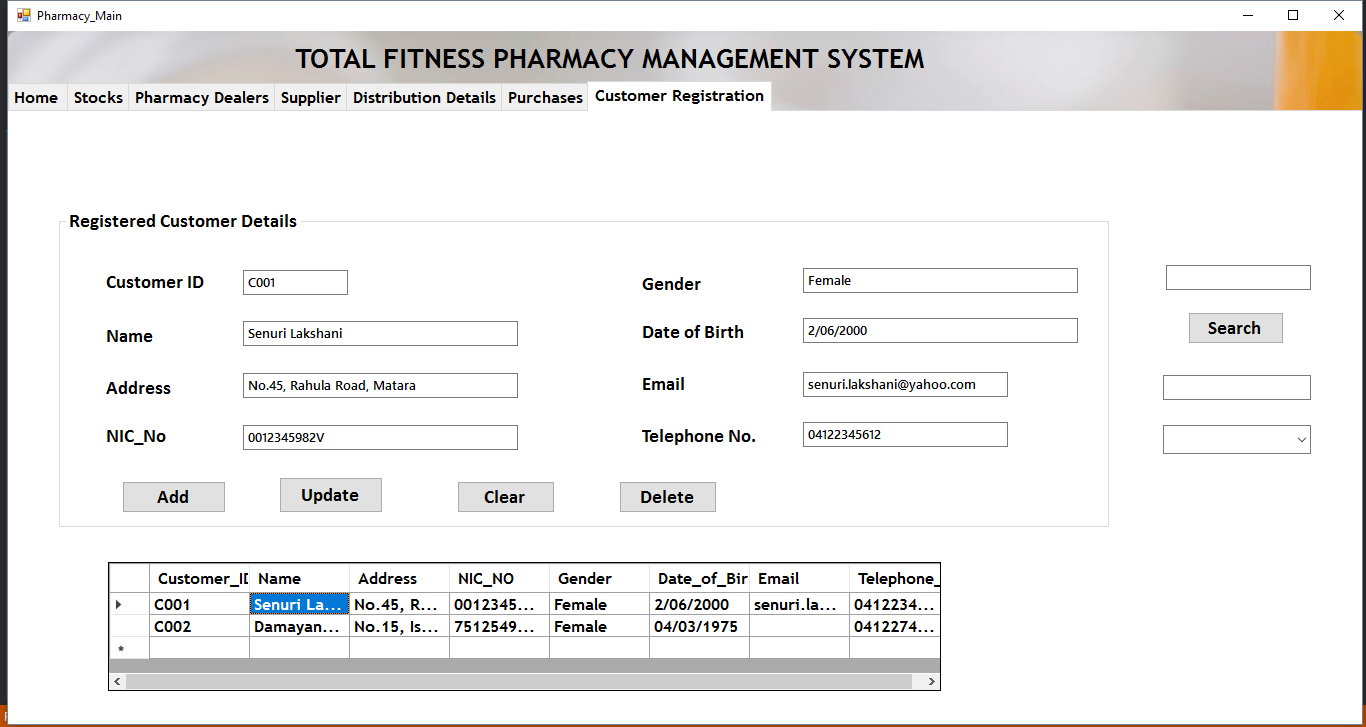


Figure 36 - Customer Registration Main

Above picture shows the tab page of customer registration. In here users can add new customers, update, delete and view them too. So let’s see how this window has been made.

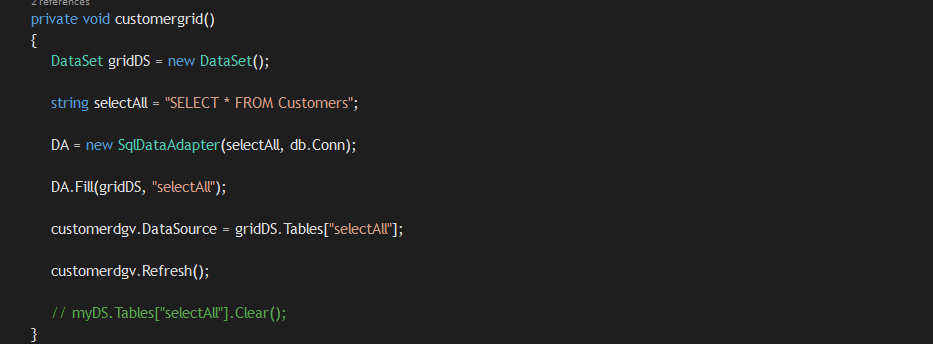


Figure 37 - customergrid()

**customergrid()** is a private method which was used to fill the data of the Customer table in the database to the data grid table in the application. Using a SQL statement, we have displayed all the records of the customer table to the customer registration tab page window. So the users can see the records and view them.

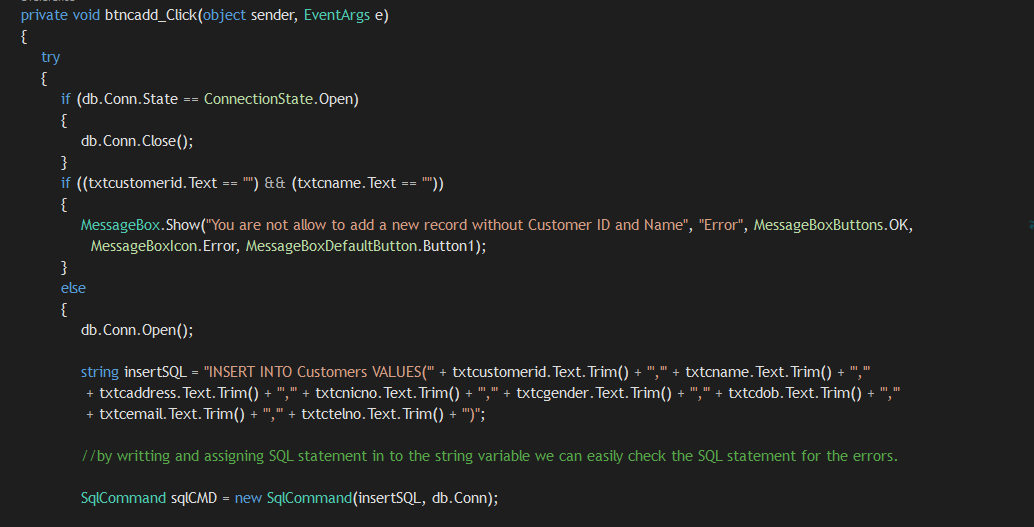
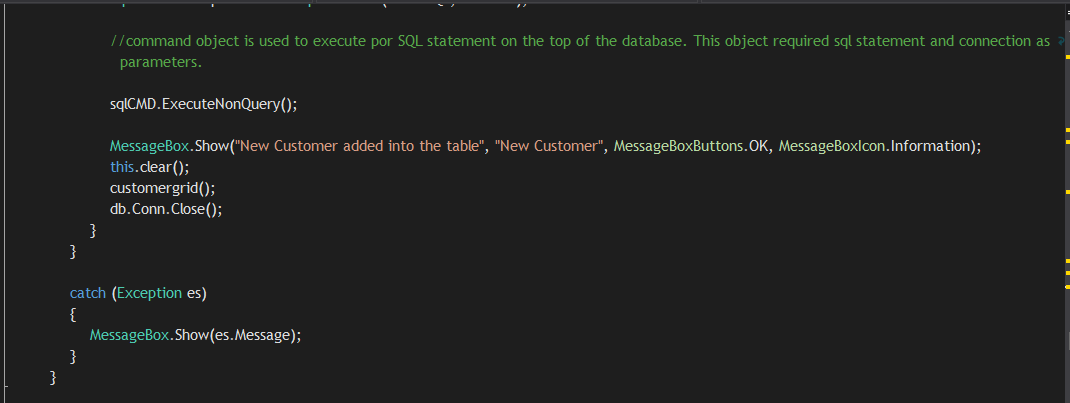


Figure 38 - Customer Add

This the code written for the Add button to add new customer details in the database. First it has been coded to check whether the user has entered the customer ID and Name to proceed. If it is so it will start to do the next steps or else, it will display a message asking the user to enter them. After that the connection starts to open and a SQL statement has been written to insert the values given in the text boxes to the Customer table. After that the **SqlCommand** executes the query and a message will be displayed as **“New Record Added into the table”**. You will see the new record in the table then. The code is written inside a try catch block to handle exceptions.

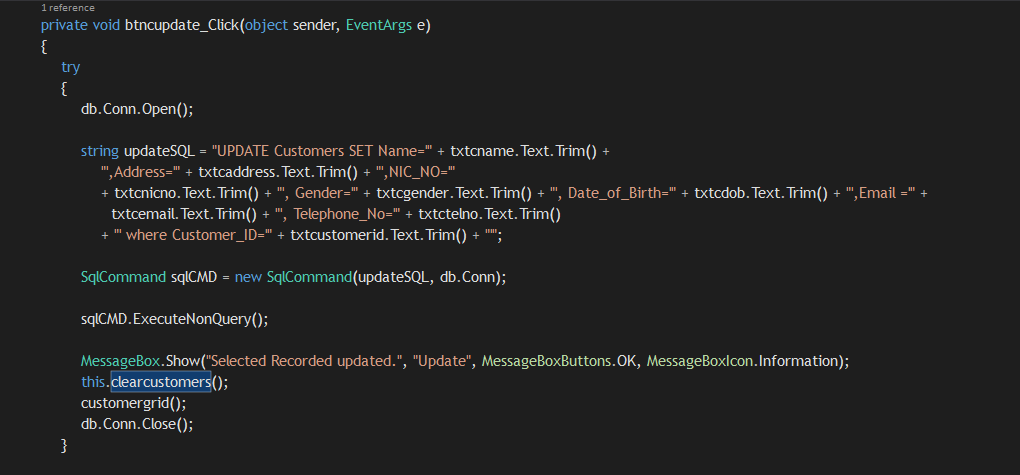


Figure 39 - Customer Update

This the code written for the Update button to update Customer details in the table. To update a record first, the user, have to select a record from the table or by searching a record. Then the user has to modify the relevant details and click Update button to update. A SQL statement has been written to Update the records by setting the values given in the text boxes. Then the **SqlCommand** executes the query and a message will be displayed as, **“Selected Record Updated”**. The updated details will be shown in the table. To handle exceptions this code has been written inside a try catch block

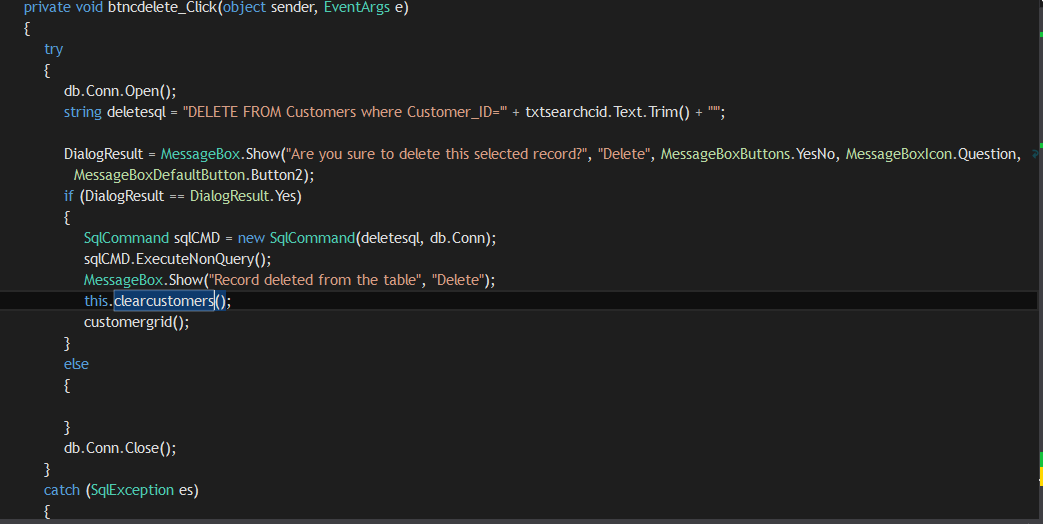
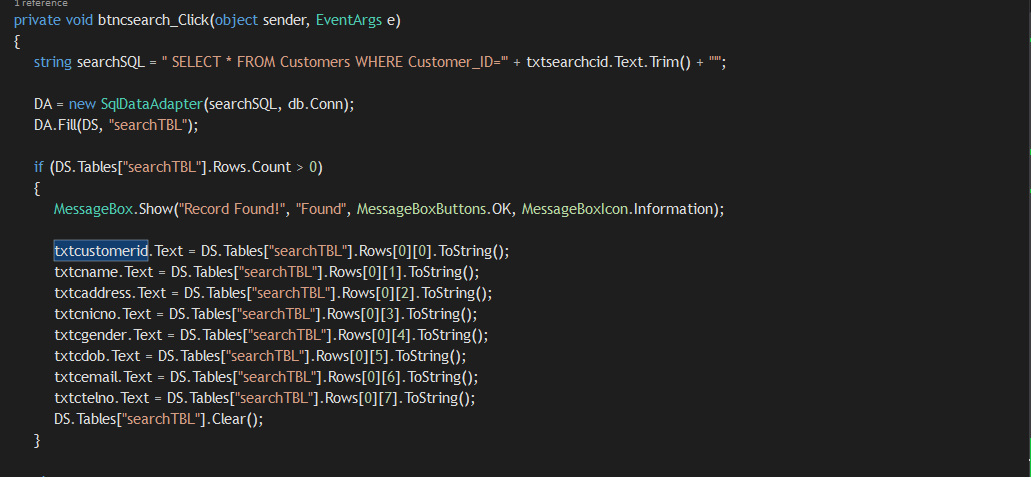


Figure 40 - Customer Delete

This is the code for the Delete button. In here a SQL statement has been written to delete the record where the Customer ID is equal to the ID given in text box. After that Using a **DialogResult** we display a message **“Are you sure to delete the selected record**” If the user clicks yes, the record will be deleted and a message will be displayed again “**Record Deleted from the table**”. Also the customer will be removed from the data grid table also to make sure the it has been deleted.



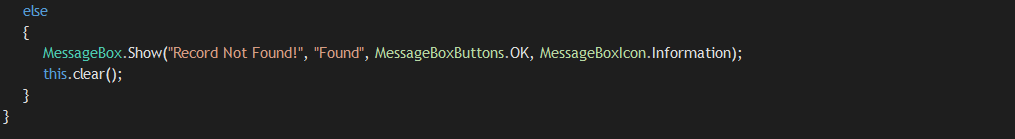


Figure 41 - Search Customer

This the code written for the search button to search customers in the table. In here above the search button there’s a text box to enter the Stock ID. After the user enters the Customer ID and click **Search** button, the Customer ID will be stored in a variable first. Then using a SQL statement, we ask to select the record where the Customer ID is equal to the ID given in the variable. When the record is found it will display a message as **“Record Found”** and the details are first loaded into a dataset. It’s like a virtual table. Then one by one each value in the dataset are set to display in text boxes for the user to see it

Likewise, we have created the other forms **pharmacy dealers, suppliers, distribution details** as same as the previous forms. Insert, update and delete codes are included in these forms too. Below pictures show the forms and their source codes.

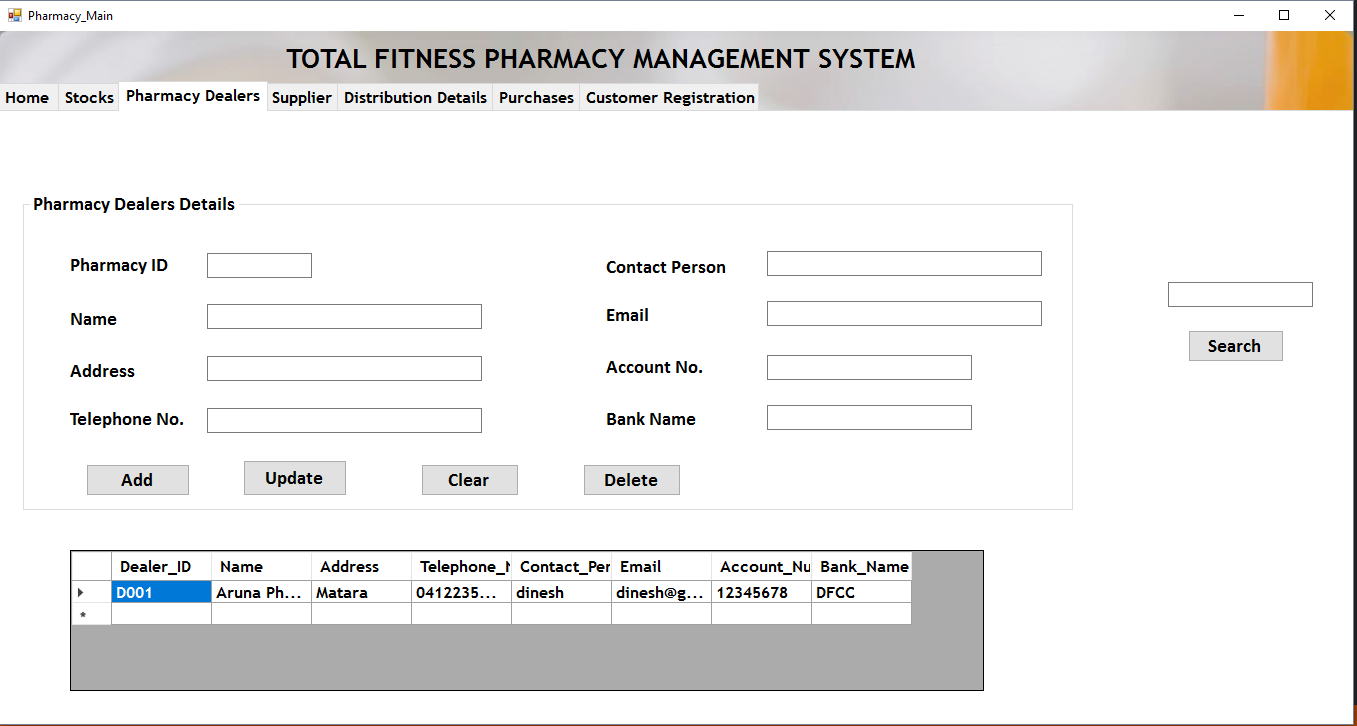
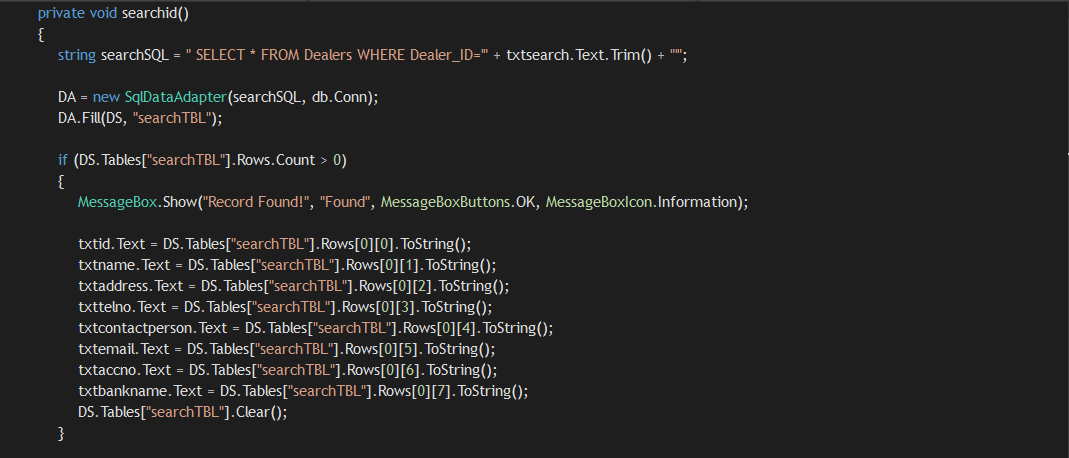
**Pharmacy Dealers**

Figure 42 - dealergrid()

Figure 43 - Pharmacy Dealers Main

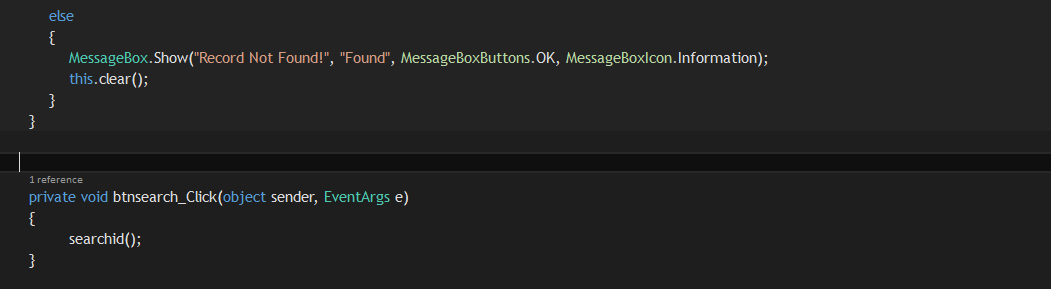
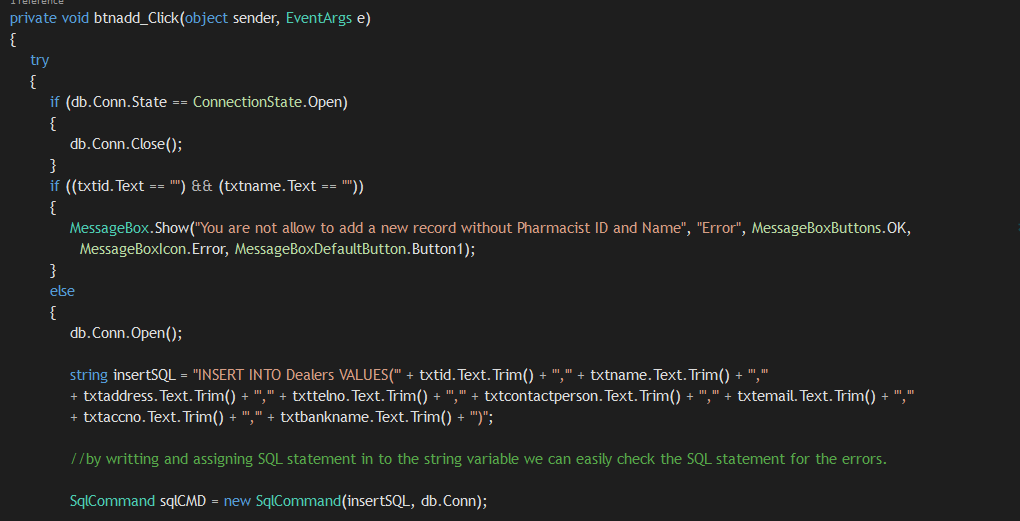
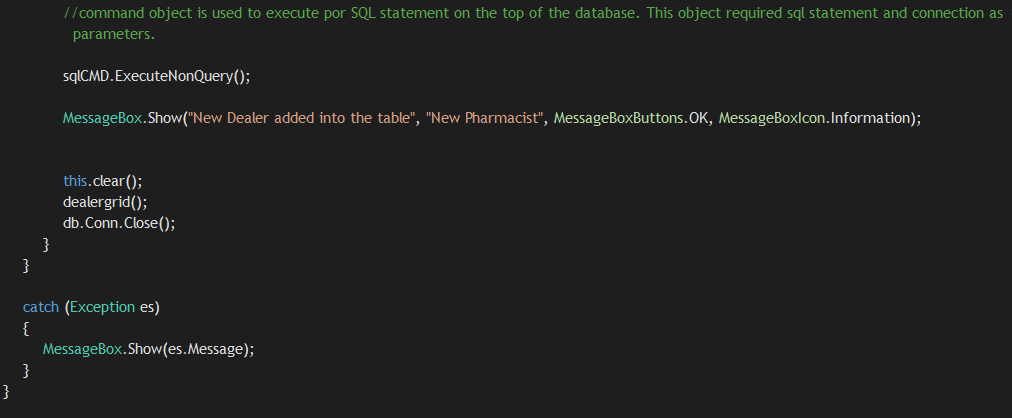


Figure 44 - Add Dealer

Figure 45 - Search Dealer

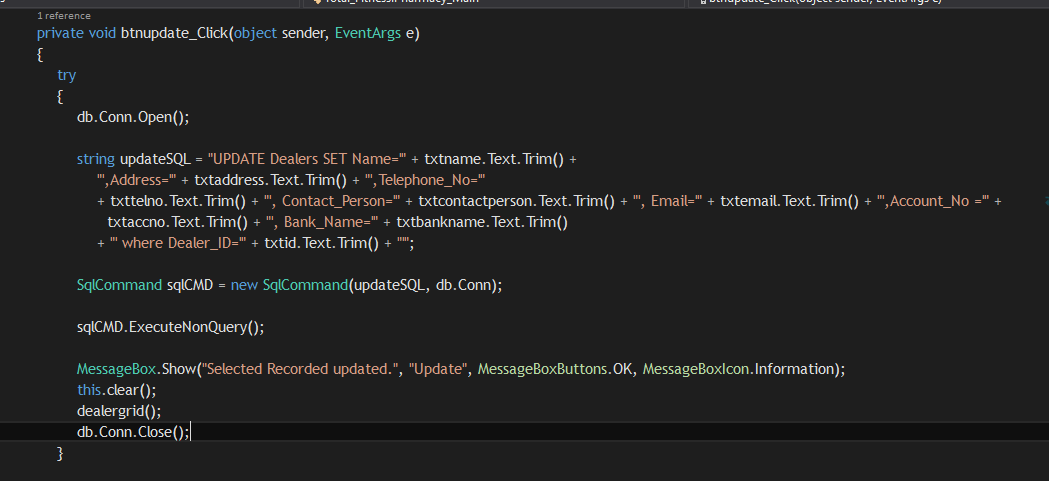
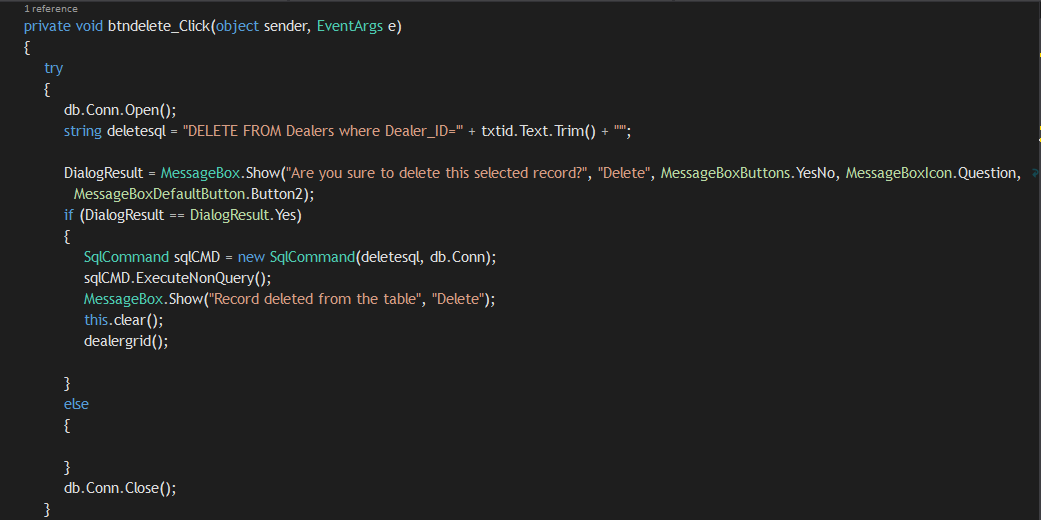


Figure 46 - Delete Dealer

Figure 47 - Update Dealer

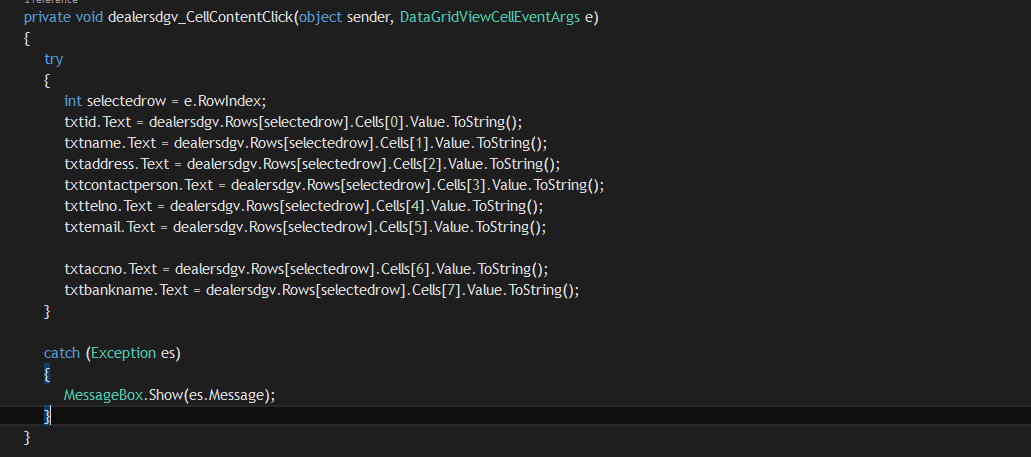


Figure 48 - Dealer Data Grid Cell Click

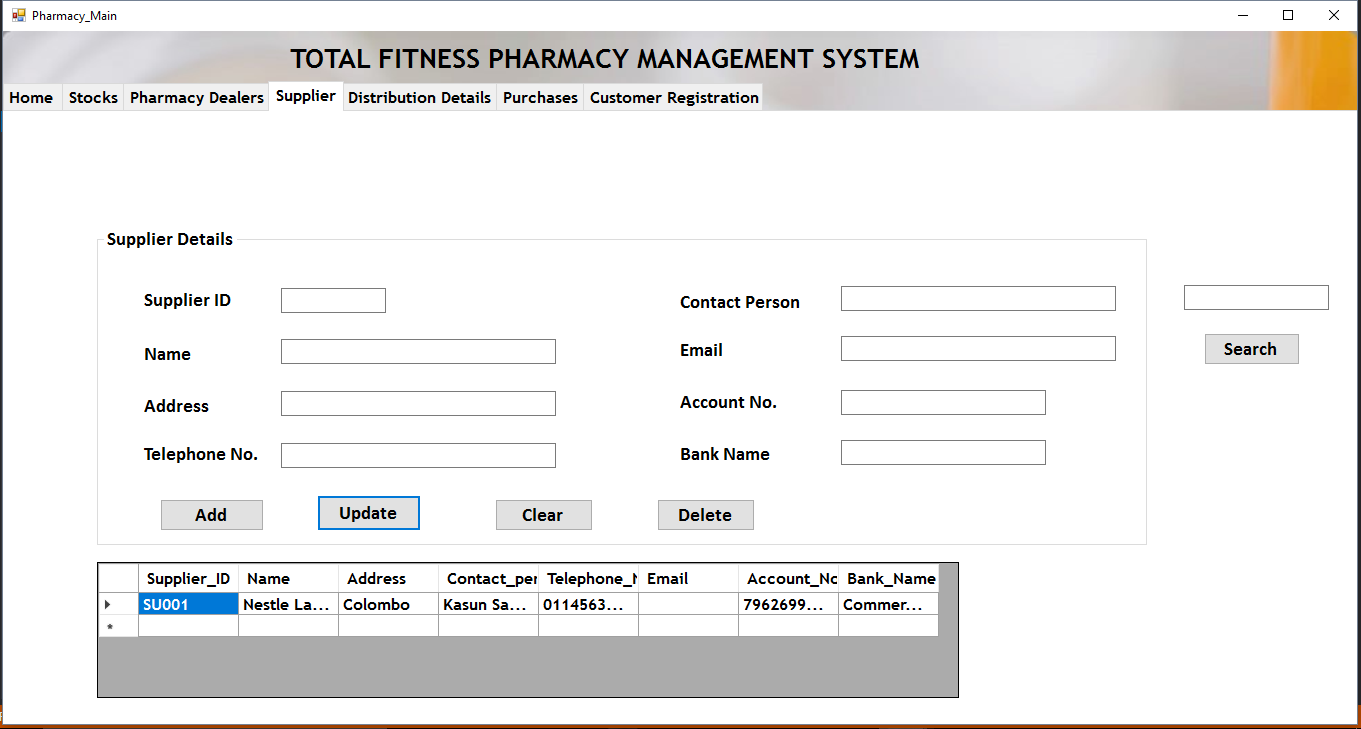
**Suppliers**

Figure 49 - Supplier Main

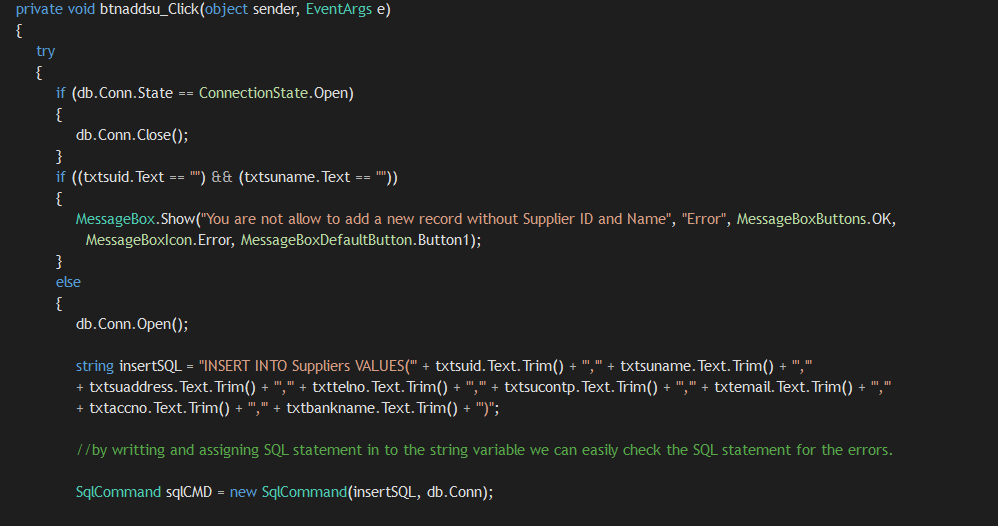
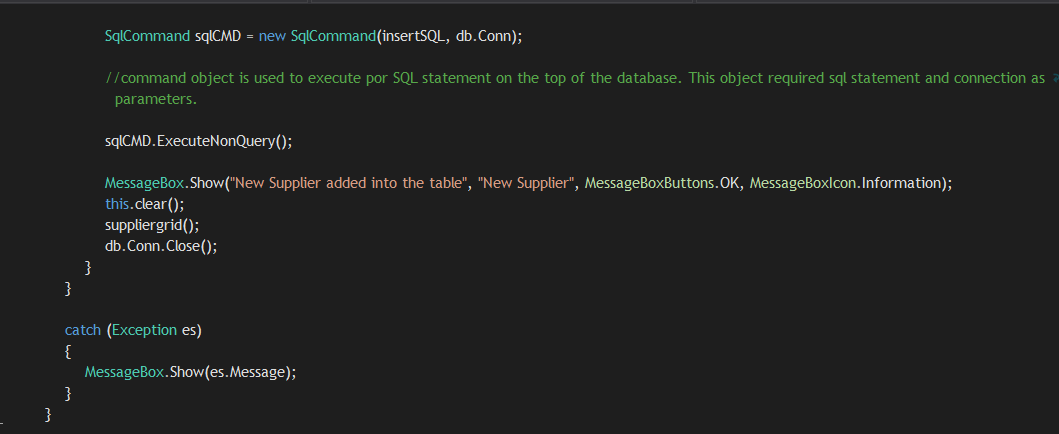
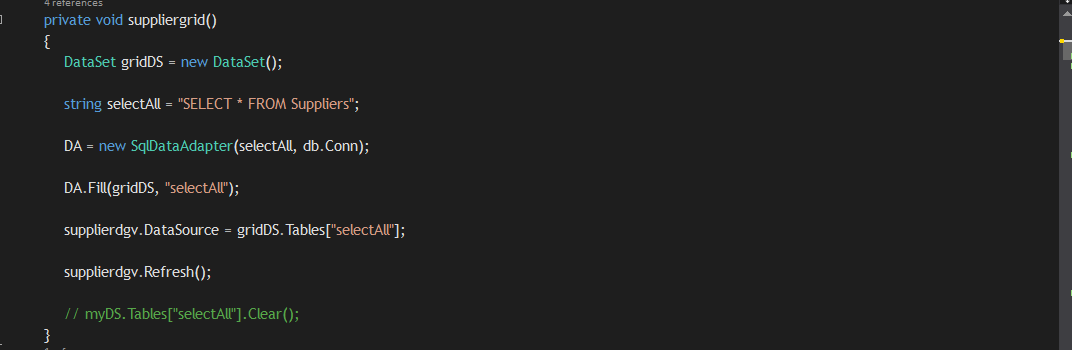
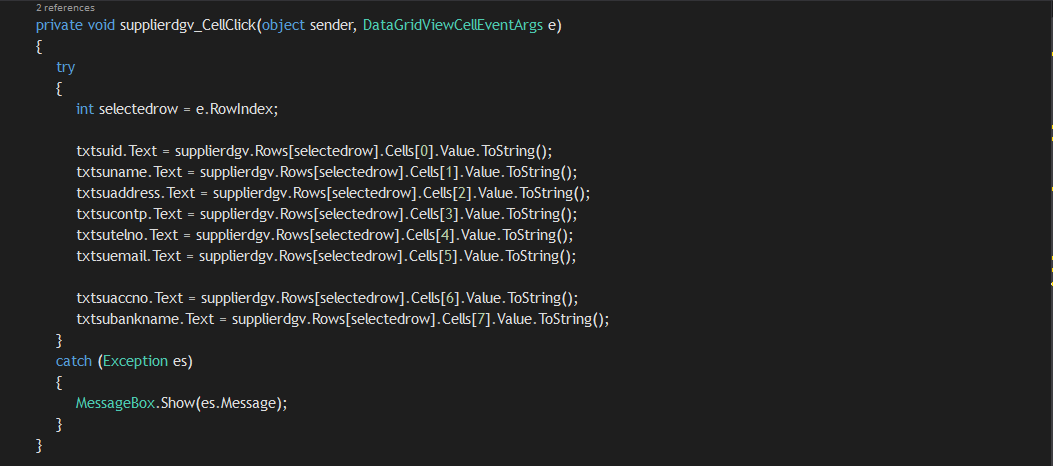
****

Figure 50 - Supplier datagrid cell click

Figure 51 - suppliergrid

Figure 52 - Supplier Add

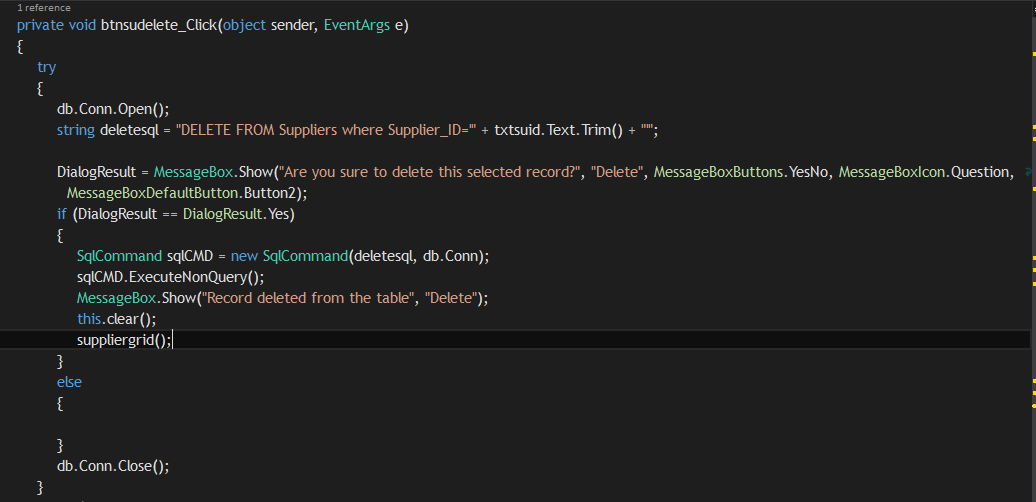
****

Figure 53 - Supplier Delete

Figure 54 - Supplier Update

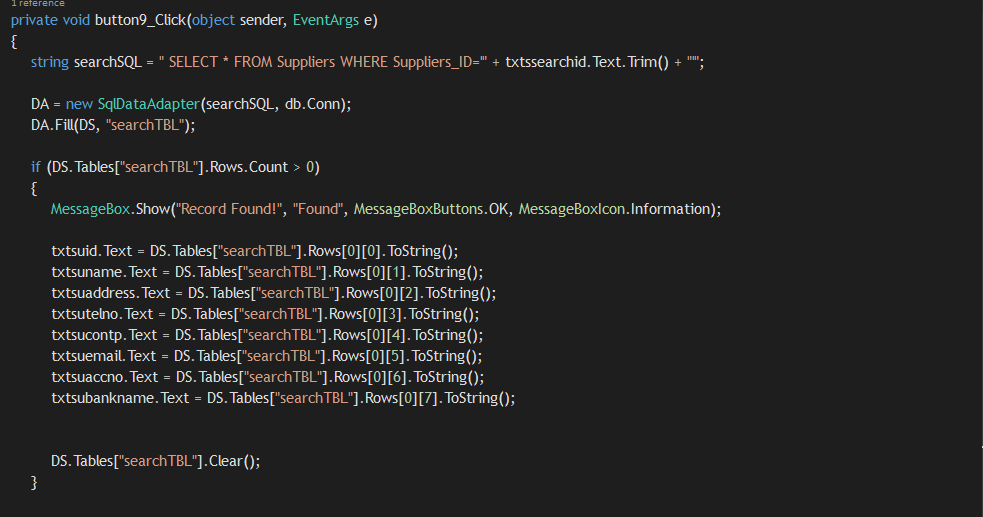
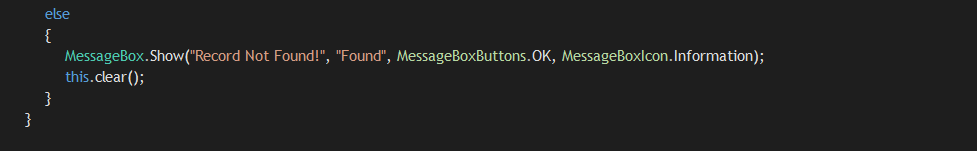
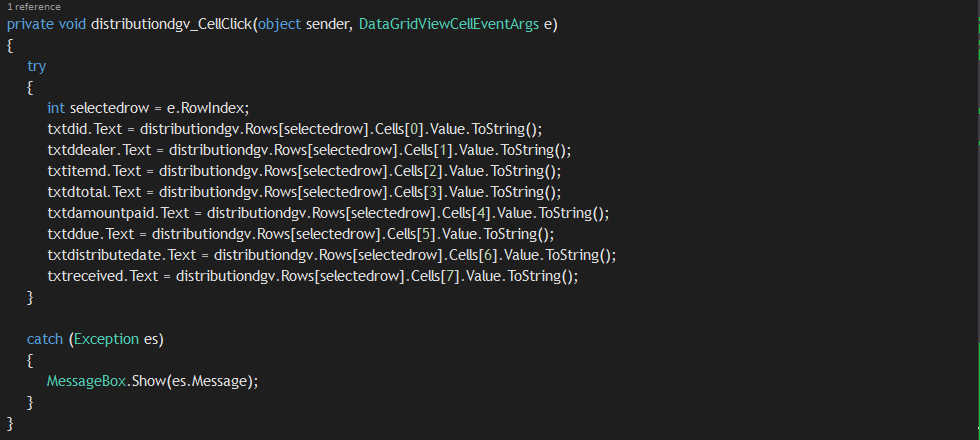
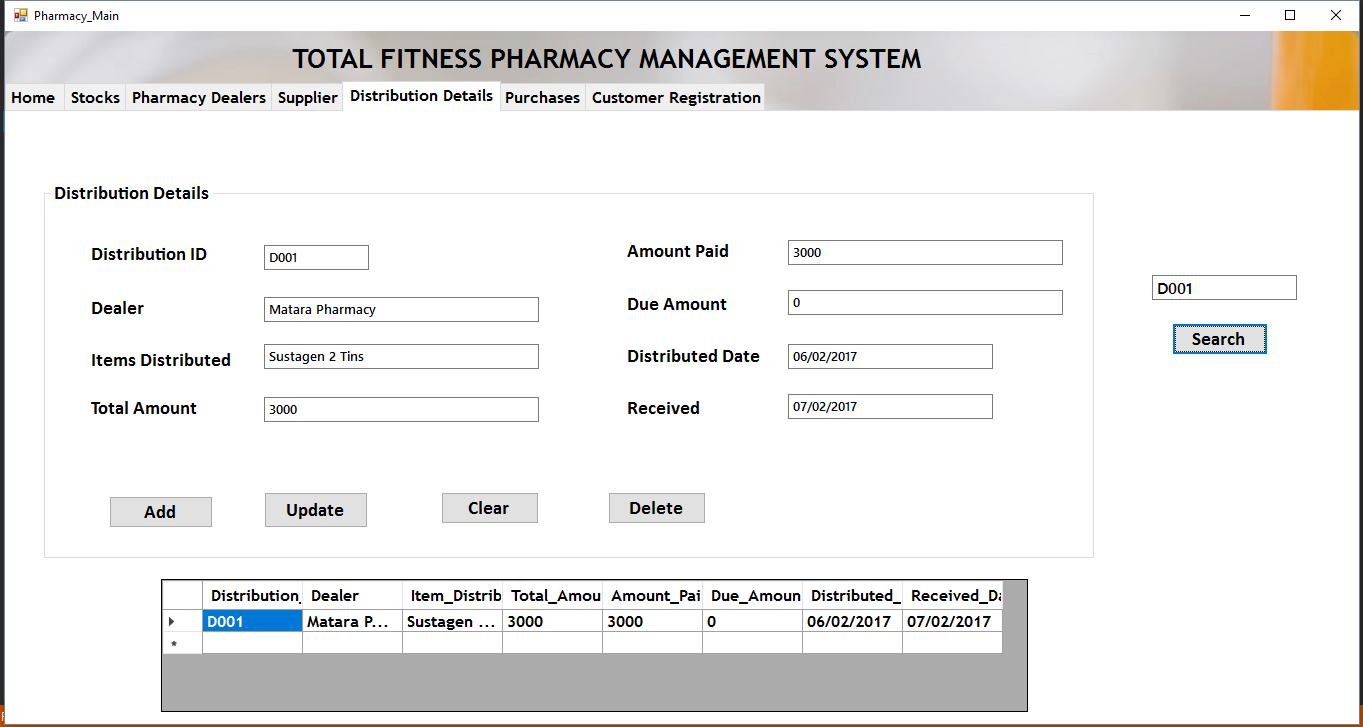
******Distribution Details**

Figure 55 - Search Supplier

Figure 56 - Distribution Data grid cell click

Figure 57 - Distribution Main

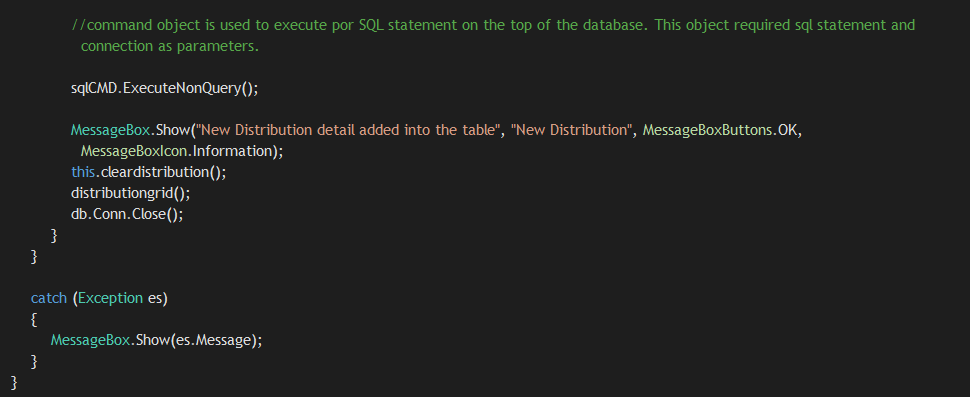
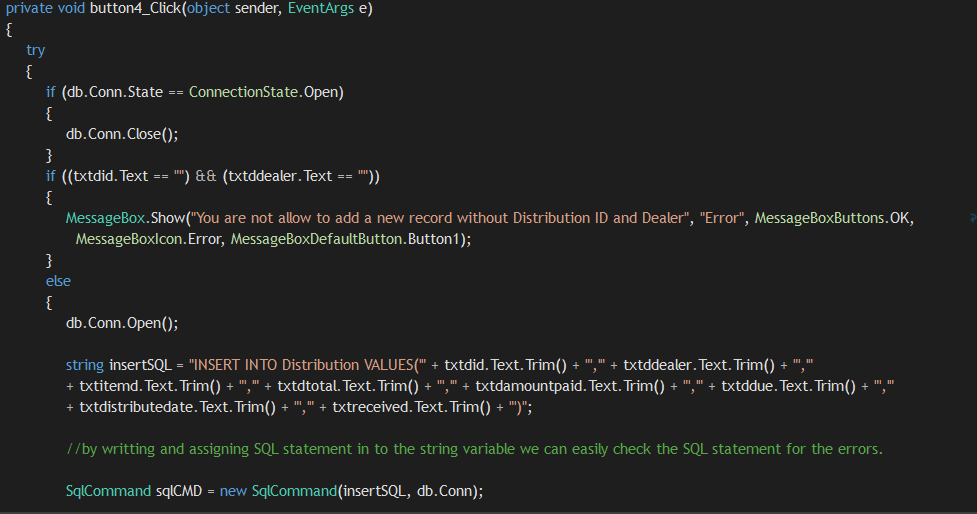
****

Figure 58 - Distribution Detail Add

****

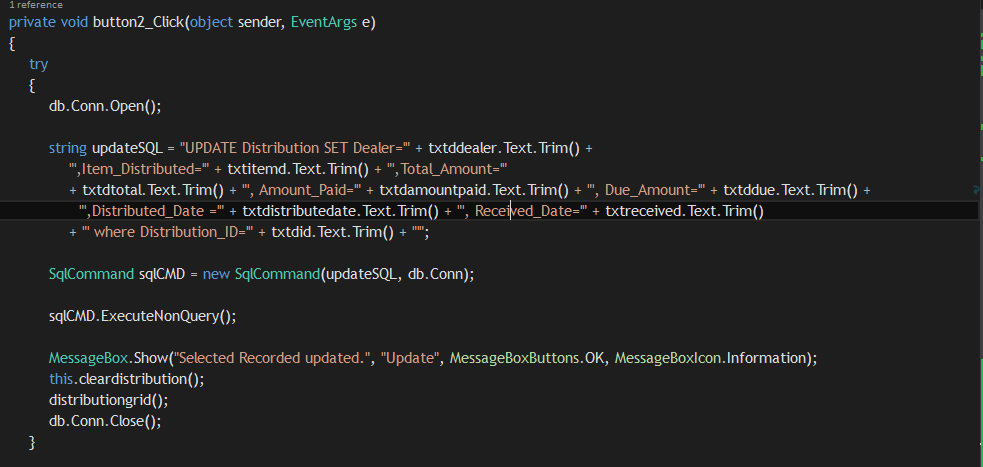
****

Figure 59 - Delete

Figure 60 - Update

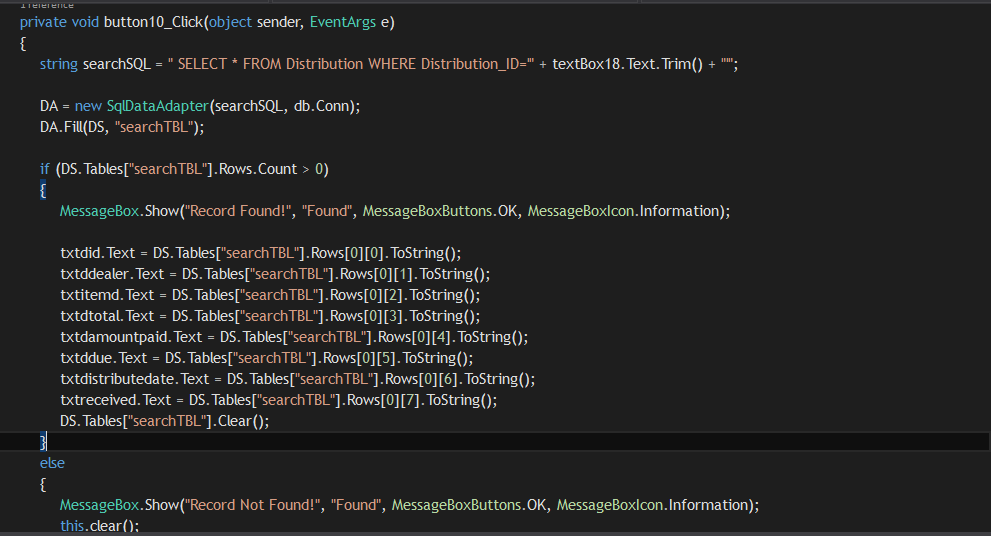
****

Figure 61 - Search Detail

**Billing**

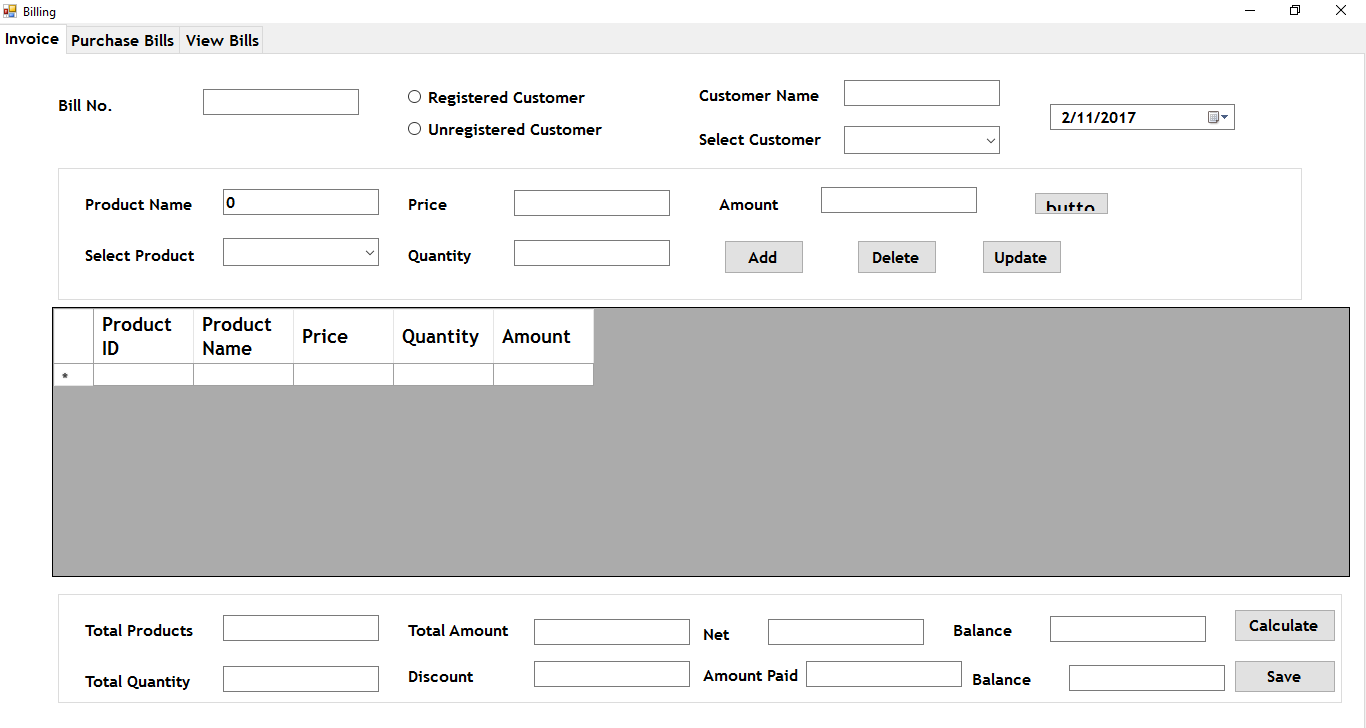
****

Figure 62 - Billing Main

Above picture shows the billing form of the applications. Users can keep records of the daily transactions of their business using this section. Users have to type the Bill no., select the customer, select the product and add the price and quantity. When the user clicks add the items will be added into the table. After that, if you want calculate the total amount, user have to just click Calculate button. Then the total amount will be calculated. Also you can add discount for this. After you have finish, just click save to save the bill.



Figure 63 - Billing Public Class

This is the public class file of the billing form. In here I have given the connection to the form by making the object instance of **connection** class file. Then I have added **SqlDataAdapter** and **Dataset** also. Three variables have been declared in here to access them in everywhere inside the form to store calculation values.

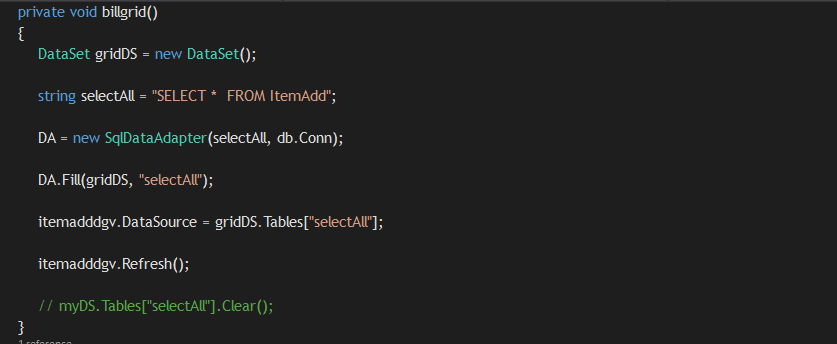


Figure 64 - billgrid()

**Billgrid()** method displays the table named **“ItemAdd”** in the database which is used to store the items that the user put each time after he clicks the Add button. Ordered items are stored here. Using a SQL statement, we have displayed the table to the data grid table.

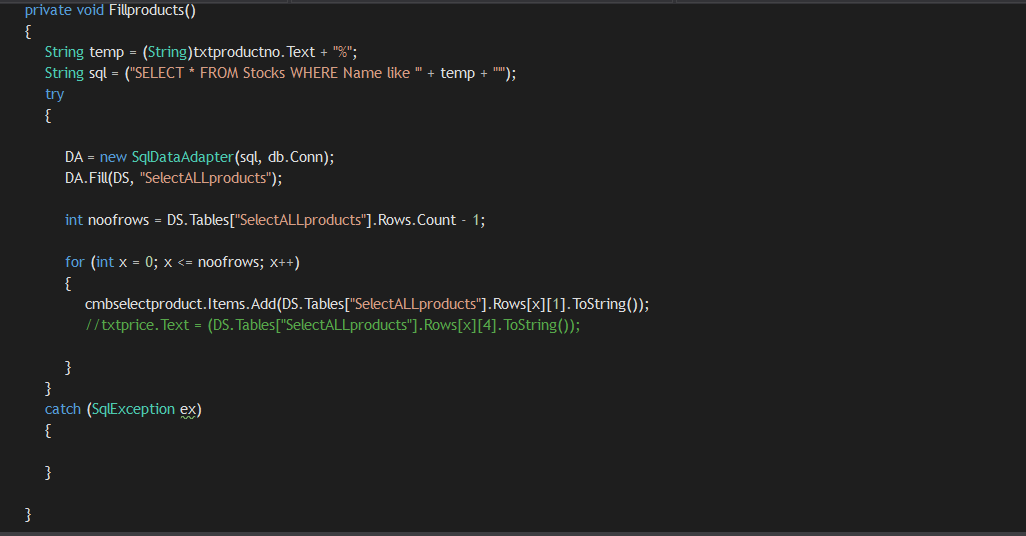
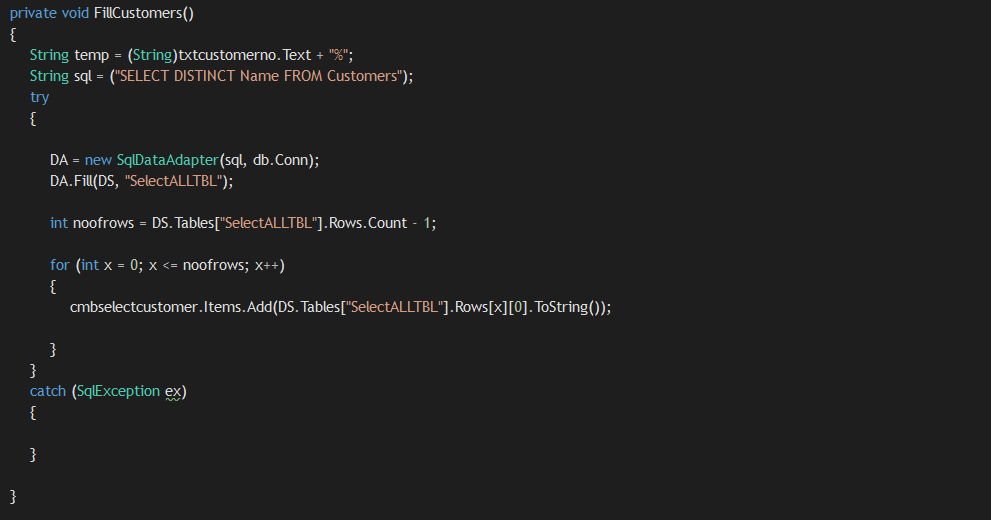
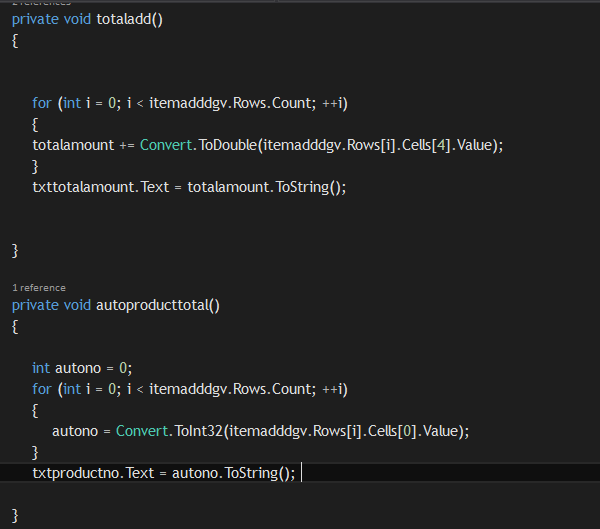


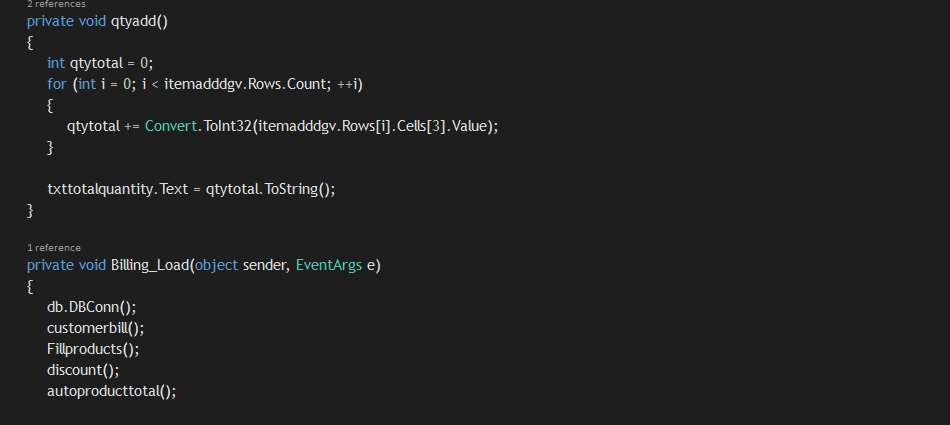
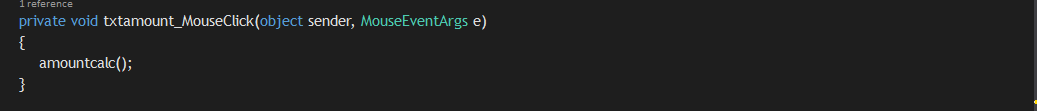
Figure 65 - searchcustomer() and Fillcustomer()

**Fillcustomers()** and **Fillproducts()** methods are used here to fill the customer combo box and product combo box with customers and products. In here we have written a SQL statement to select customers and products to add into the combo box. When the user clicks the customer combo box he will see the registered customers. Unregistered customers, the users can type the name. When you click the product combo box you will see the products. The pharamcists have to select them and proceed. Using a for loop we add the items to the combo boxes.

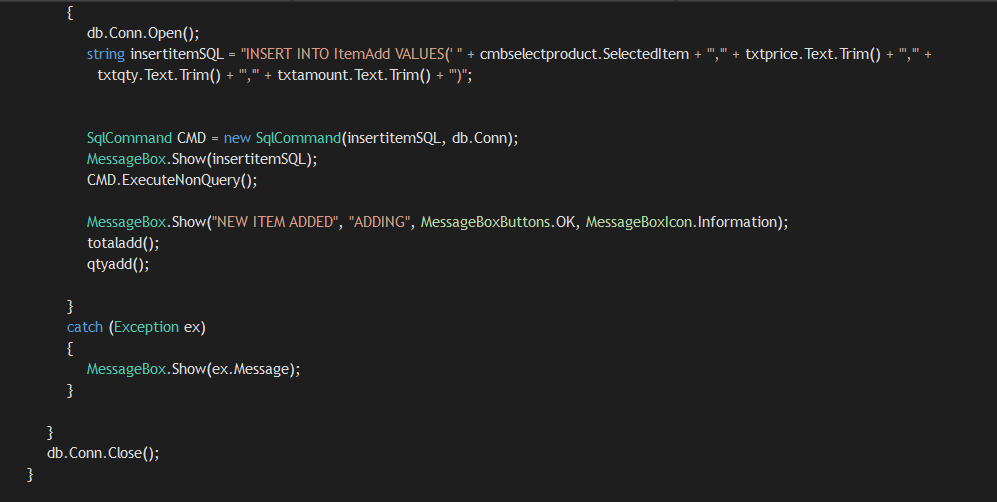


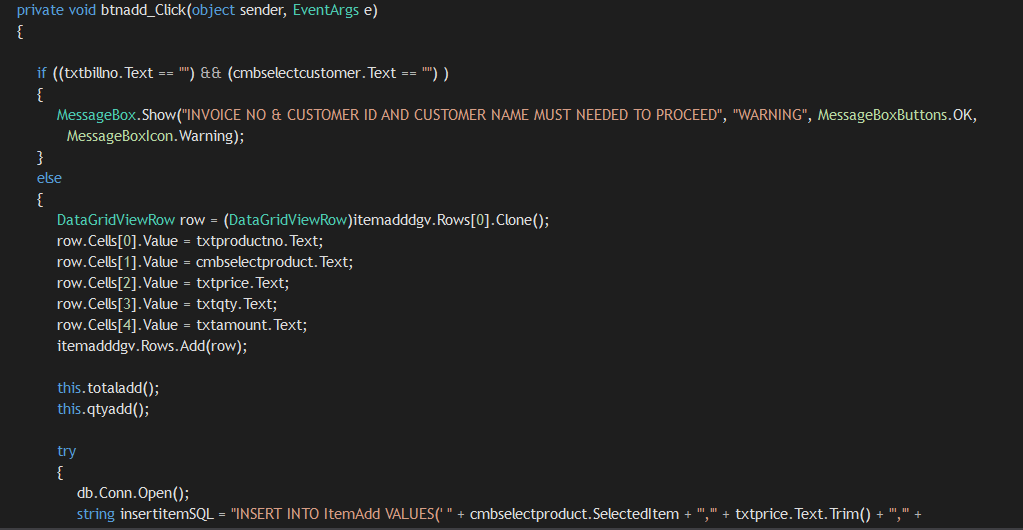
Figure 66 - amountcalc()

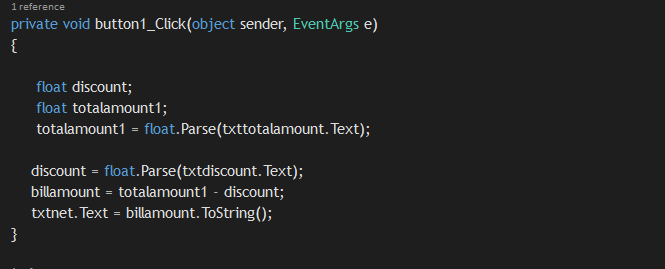
This method does the basic calculation part of the billing. Two variables have been declared to store price, and quantity. And these variables have been assigned values given from the text boxes. After that we multiply the values in **itemprice** and **noofitems** variables to get the total amount. The total amount is stored in **invoiceamount** variable and displayed to the total amount text box

**totaladd()** method does is this. Each time when the user add a new item the **totaladd()** method auto generates the current total amount. When you add another item, this method again calculates the total amount after you have add the second item. This is dine by using a for loop. **autoproduct()** does the same thing as well, but it calculates the total number of items ordered.

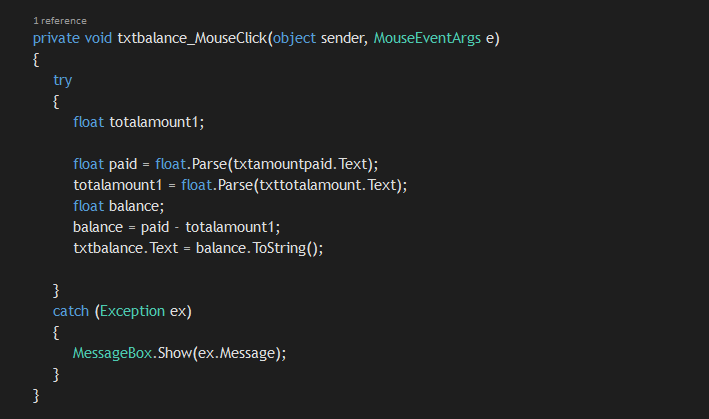
**qtyadd()** also does the same function as the **totaladd()** and **autoproduct().** In here it calculates the total number of quantities ordered. **Billing\_Load()** includes the functions which have been called to start functioning when the form loads.



This is the code written for the Add button like in other forms in pharmacy main. But in here first when the user input the details first they are stored inside a dataset to calculate the total amounts, total quantity, etc.. You can see the **totaladd()**and **qtyadd()** has been called here. After that the inserting of data into the table starts. It’s the same code which has been applied in previous forms.



This the code which has been used to calculate the discount of the bill.

****

This is the code written to check the balance of the bill.

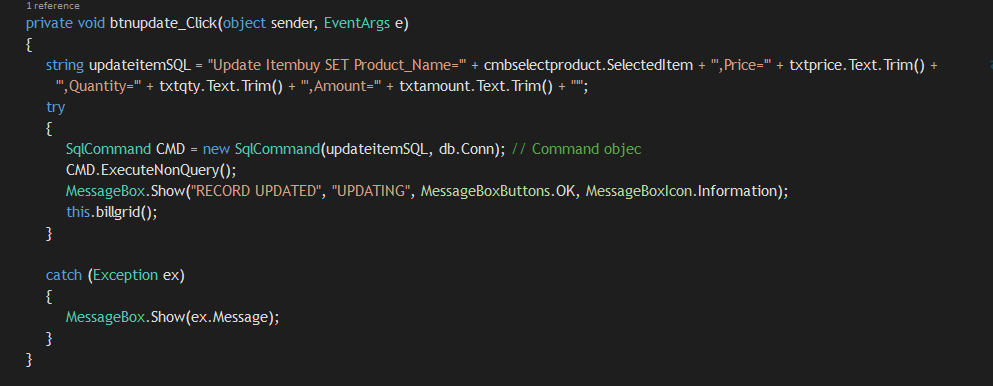
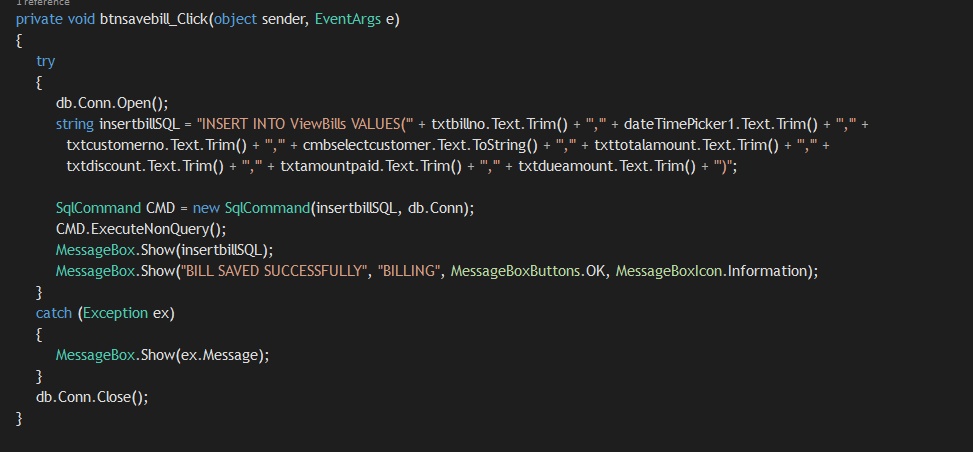
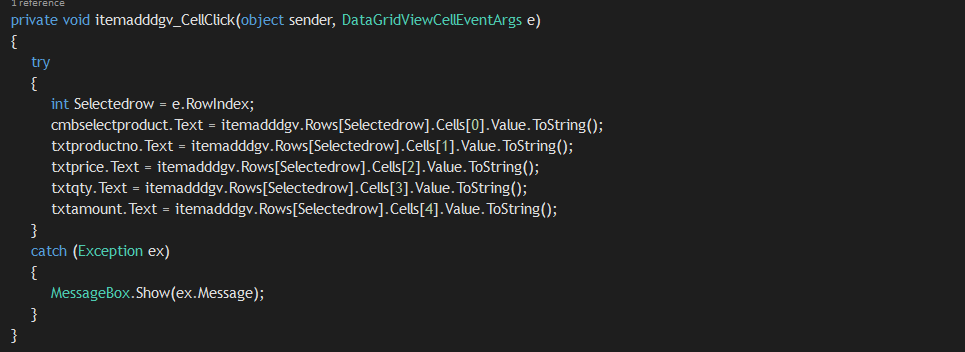


Figure 67 - Save Bil

Figure 68 - Update Bill

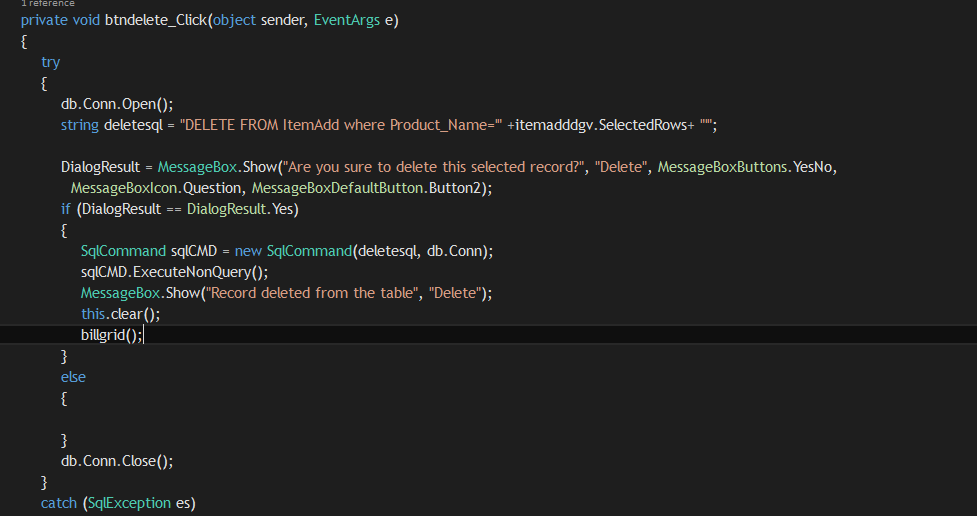
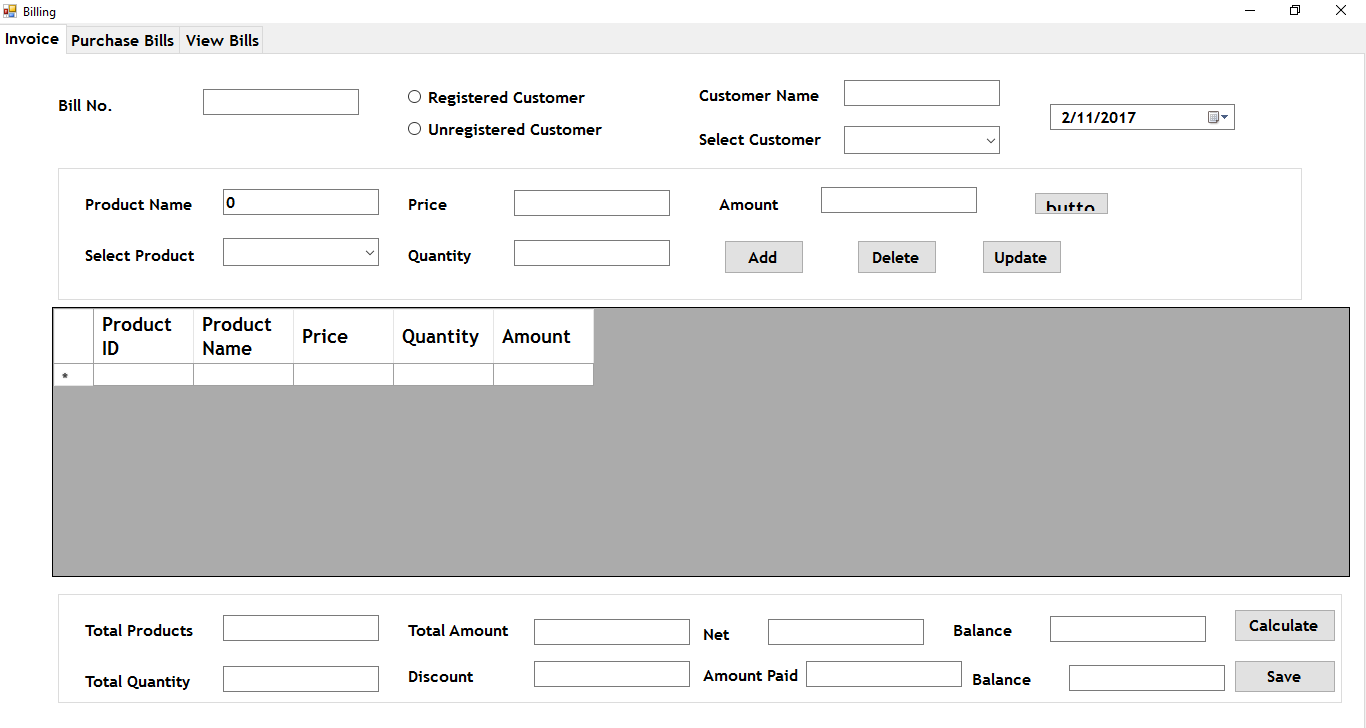
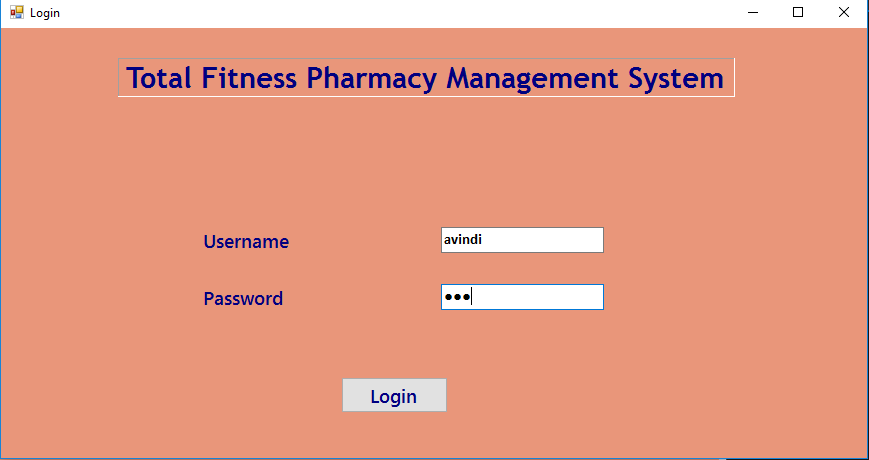


Figure - View Bill

Figure 70 - Delete

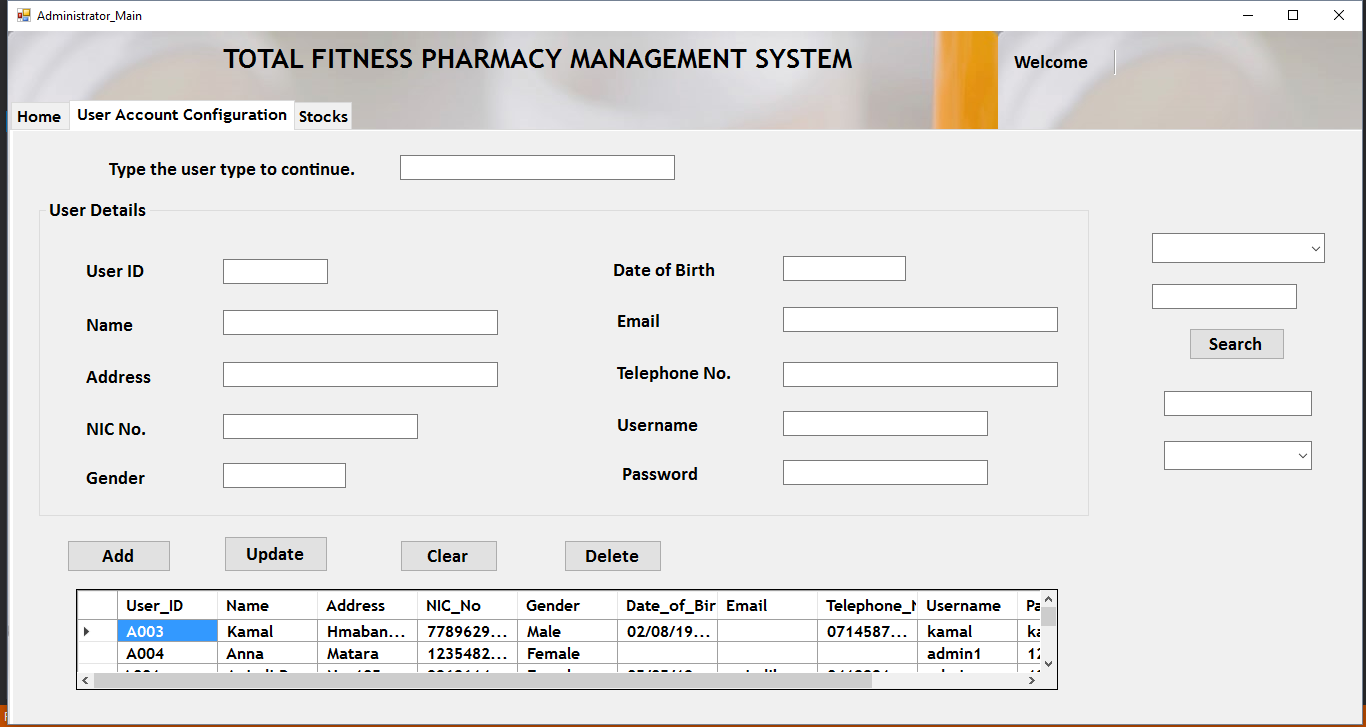
### User Manual

**TOTAL FITNESS PHARMACY MANAGEMENT SYSTEM USER MANUAL**



1. **HOW TO LOGIN TO THE SYSTEM?**

* First type the username and password.
* If the details are correct you will get a message “Login Successful”. And you will be directed to a main window
* Else you would get an error message



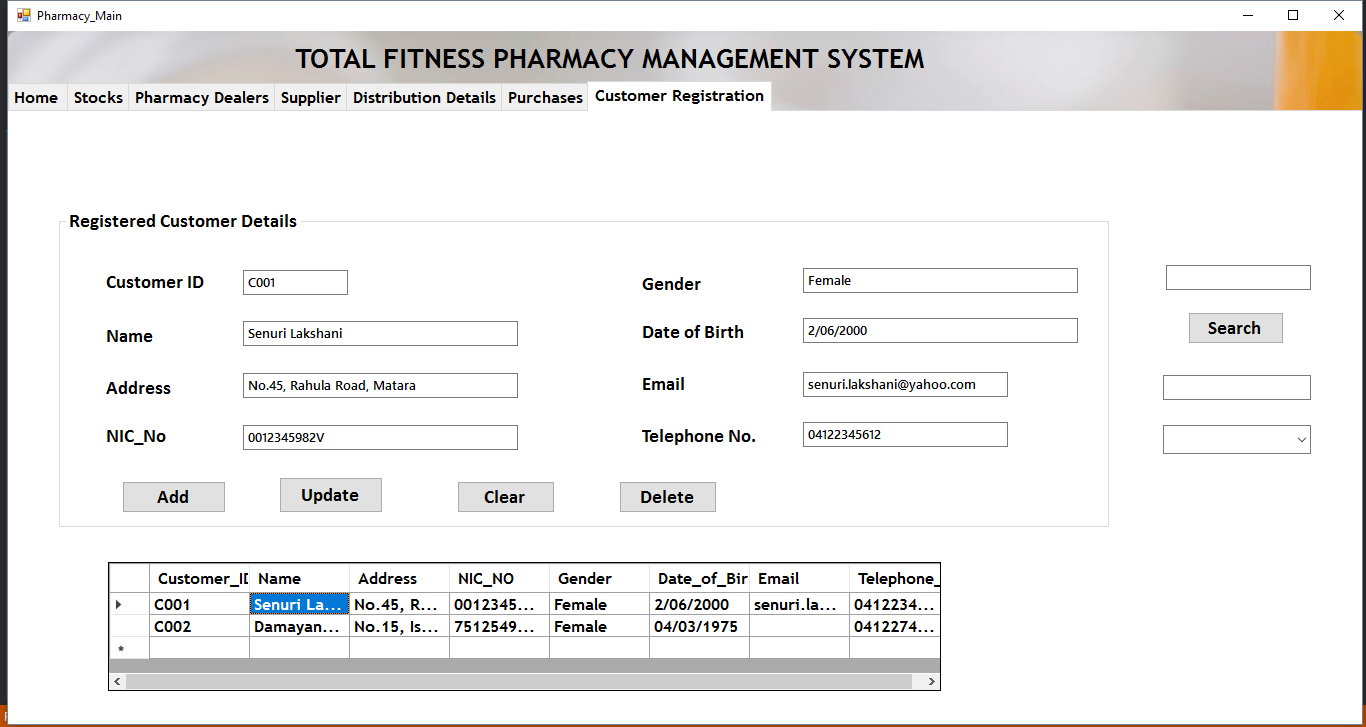
1. **HOW TO ADD, UPDATE, DELETE AND SEARCH USERS?**

* **To Add New User:** Add all the user details and click “Add” button. The details will be loaded in the table.
* **To Update User:** Select a user from the table or using search option. Then update the details and click “**Update”** button.
* **To Delete User:** Select a user from the table or using search option. Then click “**Delete”** button.
* **To Search User:** Type the User ID and click **“Search”**



1. **HOW TO ADD, UPDATE, DELETE AND SEARCH ITEMS?**

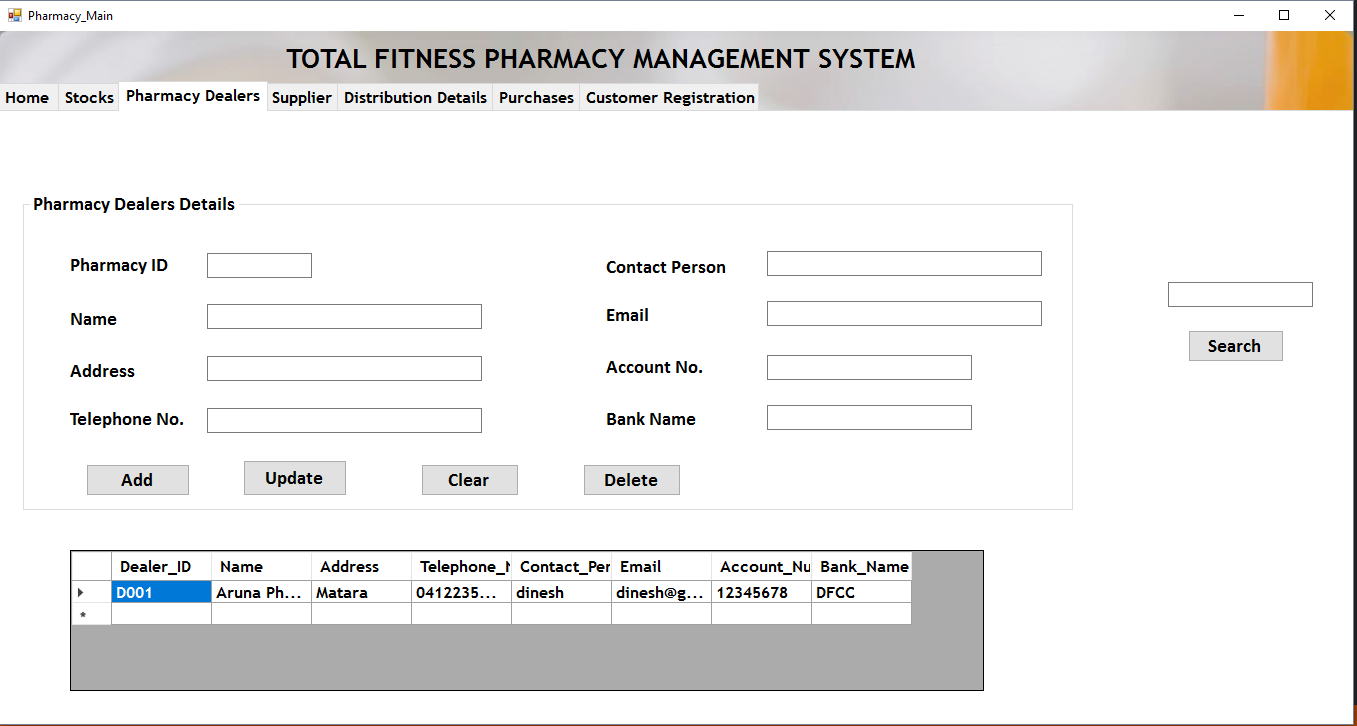
* **To Add New Item:** Add all the item details and click “Add” button. The details will be loaded in the table.
* **To Update Item:** Select a item from the table or using search option. Then update the details and click “**Update”** button.
* **To Delete Item:** Select an item from the table or using search option. Then click “**Delete”** button.
* **To Search Item:** Type the Stock ID and click **“Search”**

****

1. **HOW TO ADD, UPDATE, DELETE AND SEARCH DEALERS?**

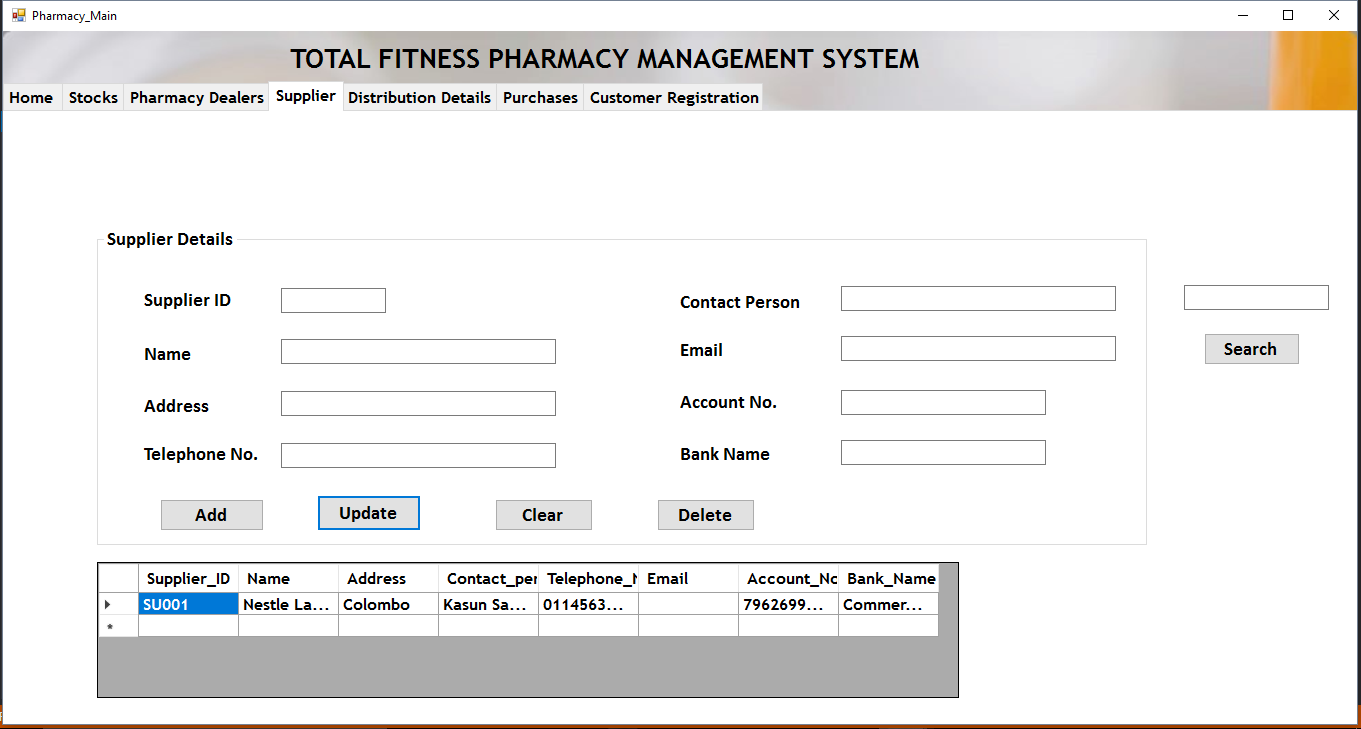
* **To Add New Customer:** Add all the customer details and click “**Add**” button. The details will be loaded in the table.
* **To Update Customer:** Select a customer from the table or using search option. Then update the details and click “**Update”** button.
* **To Delete Customer:** Select a customer from the table or using search option. Then click “**Delete”** button.
* **To Search Item:** Type the Customer ID and click **“Search”**

1. **HOW TO ADD, UPDATE, DELETE AND SEARCH CUSTOMER?**
2. **HOW TO ADD, UPDATE, DELETE AND SEARCH DEALERS?**

****

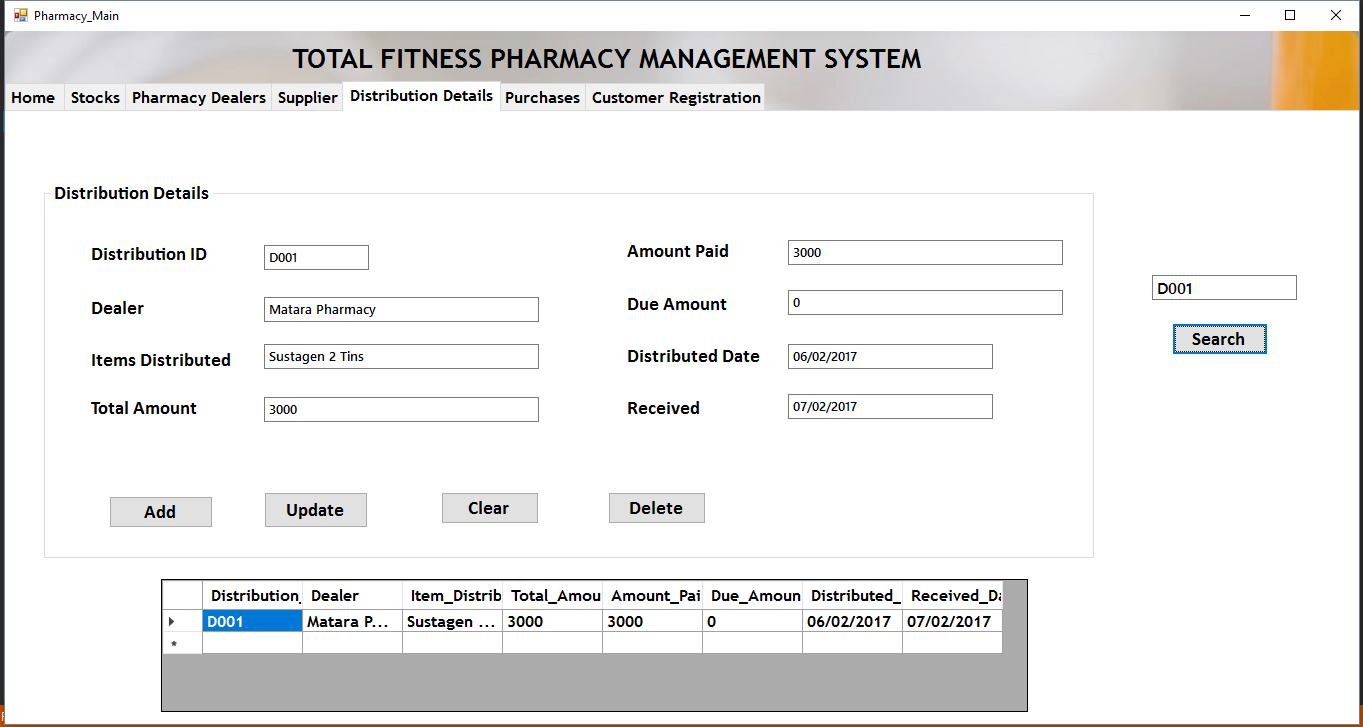
* **To Add New Dealer:** Add all the dealer details and click “**Add**” button. The details will be loaded in the table.
* **To Update Dealer:** Select a dealer from the table or using search option. Then update the details and click “**Update”** button.
* **To Delete Dealer:** Select a dealer from the table or using search option. Then click “**Delete”** button.
* **To Search Dealer:** Type the Dealer ID and click **“Search”**

**6. HOW TO ADD, UPDATE, DELETE AND SEARCH SUPPLIER**

****

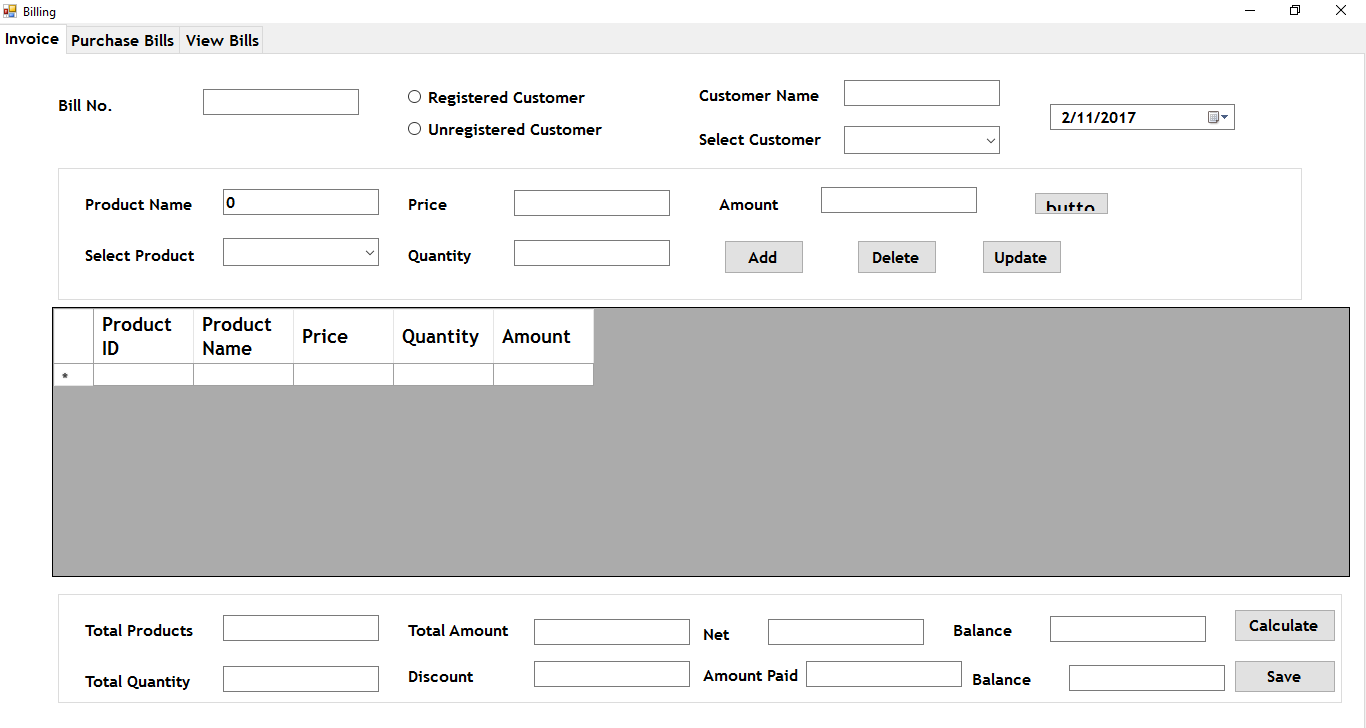
* **To Add New Supplier:** Add all the supplier details and click “**Add**” button. The details will be loaded in the table.
* **To Update Supplier:** Select a supplier from the table or using search option. Then update the details and click “**Update”** button.
* **To Delete Supplier:** Select a supplier from the table or using search option. Then click “**Delete”** button.
* **To Search Supplier:** Type the Supplier ID and click **“Search”**

**7. HOW TO ADD, UPDATE, DELETE AND SEARCH DISTRIBUTION DETAILS?**

****

* **To Add New Distribution:** Add all the details and click “**Add**” button. The details will be loaded in the table.
* **To Update Distribution detail:** Select a one from the table or using search option. Then update the details and click “**Update”** button.
* **To Delete Distribution Detail:** Select a distributor from the table or using search option. Then click “**Delete”** button.
* **To Search Distribution detail:** Type the ID and click **“Search”**

**8. HOW TO ADD, UPDATE, DELETE AND SAVE BILLS?**

****

* **To Add Bill:** Add all the details product name, price quantity and click “**Add**” button. The details will be loaded in the table.
* **To Update Bill:** Select a one from the table. Then update the details and click “**Update”** button.
* **To Delete Bill:** Select from the table. Then click “**Delete”** button.
* **To save bill:** After you have done all the calculations click **“Save”.**
* **To view bill:** Click **View Bills.**

# Task 05



## System Testing

Before implementing the system to the Total Fitness Pharmacy, the system needs to be tested properly. System testing means testing each and every components of the system whether the expected output is received or not when giving the input to the system. There are different types of testing techniques.

1. Black Box Testing
2. White Box Testing
3. Gray Box Testing
   * 1. **Black Box Testing**

Black box testing means examining requirement functionality of the application without testing the interior part of it. Interior part means the structure of the program, codes, etc. In here we check the interface of the application, performance errors, and check whether we get the expected output from the application when data inserted. It’s very easy to test than the white box testing. In here programming knowledge isn’t required to test. Also, we can start the test cases too using this technique.

* + 1. **White Box Testing**

White Box testing means, examining the interior part of the application like examining the codes, design, conditions etc...as mentioned before. This testing is known as the Clear Box Testing, Open Box Testing, Glass Box Testing. In here the GUI isn’t needed for testing like black box testing. Testing is more thorough, with the possibility of covering most paths. Also we need programming knowledge is required to test the codes. Also, this testing is quite expensive and consume more time.

1. **Grey Box Testing**

Grey box testing is a combination of white box and black box testing. In here the tester knows only the interior part to some extent only

So among these three testing techniques we have to choose a one for our application to test. When we consider about black box testing technique, we have to test only the exterior part of the application. Also its easy to start and we can start test cases too, programming knowledge isn’t required for this. But in white box testing, we have to test only the interior part of the application. We need the programming knowledge for this and we have to examine the code, interior structure and design error handlings etc.... So it takes time when doing this type of testing. Grey box testing is a combination of both black box and white box testing. Though it’s a hybrid of both the techniques the tester can’t get access to test the source code. Therefore, by comparing these three techniques the leaner chooses Black Box Testing for this application. It is because, as mentioned before this technique is easy and we can start the test case too and many other benefits mentioned above.

## Test Case

“A test case is a set of conditions or variables under which a tester will determine whether a system under test satisfies requirements or works correctly. The process of developing test cases can also help find problems in the requirements or design of an application.” (Software Testing Fundamentals, 2016)

Table 1 - Test Case

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test No.** | | **Test name** | **Changes made** | **Expected result** | **Actual result** | **Pass/Fail** |
| 1 | 1.1 | User Login | Give the correct username and password. | Log on to the system. | Log on to the system. | Pass |
|  | 1.2 |  | Give an incorrect username or password | Displays an error message “Invalid Username or Password” | Displayed an error message “Invalid Username or Password” | Pass |
| 2 | 2.1 | Create User | Give all the user details and click “Add” button | Displays a message “New user added into the table” | Displayed a message “New user added into the table” | Pass |
|  | 2.2 |  | Giving other details except Name and User ID and click “Add” button | Displays an error message “You are not allow to add a new record without User ID and Name” | Displays an error message “You are not allow to add a new record without User ID and Name” | Pass |

Table 2 - Test Case

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test No.** | **Test name** | **Changes made** | **Expected result** | **Actual result** | **Pass/Fail** |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 3 | 3.1 | Update User | Selecting a user from the table, updating details and click “Update” button | Displays a message “Selected Record Updated” | Displays a message “Selected Record Updated” | Pass |
|  | 3.2 |  | Click the update button without selecting a user | Displays an error message | Displayed an error message | Pass |
| 4 | 4.1 | Delete User | Selecting a user and click “Delete” | Displays a message “Record deleted from the table” | Displayed a message “Record deleted from the table” | Pass |
|  | 4.2 |  | Click the “Delete” button without selecting user | Record will not be deleted | Record will not be deleted |  |
| 5 | 5.1 | Add Item | Give all the item details and click “Add” button | Displays a message “New item added into the table” | Displayed a message “New item added into the table” | Pass |
|  | 5.2 |  | Giving other details except Name and Stock ID and click “Add” button | Displays an error message “You are not allow to add a new record without Stock ID and Name” | Displays an error message “You are not allow to add a new record without Stock ID and Name” | Pass |
| 6 | 6.1 | Update Item | Selecting an item from the table, updating details and click “Update” button | Displays a message “Selected Record Updated” | Displays a message “Selected Record Updated” | Pass |
|  | 6.2 |  | Click the update button without selecting an item | Displays an error message | Displayed an error message | Pass |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 7 | 7.1 | Delete Item | Selecting an item and click “Delete” | Displays a message “Record deleted from the table” | Displayed a message “Record deleted from the table” | Pass |
|  | 7.2 |  | Click the “Delete” button without selecting item | Record will not be deleted | Record will not be deleted | Pass |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 8 | 8.1 | Add Dealer | Give all the pharmacy dealer details and click “Add” button | Displays a message “New dealer added into the table” | Displayed a message “New dealer added into the table” | Pass |
|  | 8.2 |  | Giving other details except Name and Pharmacy ID and click “Add” button | Displays an error message “You are not allow to add a new record without Pharmacy ID and Name” | Displays an error message “You are not allow to add a new record without Pharmacy ID and Name” | Pass |
| 9 | 9.1 | Update Dealer | Selecting an dealer from the table, updating details and click “Update” button | Displays a message “Selected Record Updated” | Displays a message “Selected Record Updated” | Pass |
|  | 9.2 |  | Click the update button without selecting a dealer | Displays an error message | Displayed an error message | Pass |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 10 | 10.1 | Delete dealer | Selecting an dealer and click “Delete” | Displays a message “Record deleted from the table” | Displayed a message “Record deleted from the table” | Pass |
|  | 10.2 |  | Click the “Delete” button without selecting dealer | Record will not be deleted | Record will not be deleted | Pass |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 11 | 11.1 | Add Supplier | Give all the supplier details and click “Add” button | Displays a message “New supplier added into the table” | Displayed a message “New supplier added into the table” | Pass |
|  | 11.2 |  | Giving other details except Name and Supplier ID and click “Add” button | Displays an error message “You are not allow to add a new record without Supplier ID and Name” | Displays an error message “You are not allow to add a new record without Supplier ID and Name” | Pass |
| 12 | 12.1 | Update Supplier | Selecting an supplier from the table, updating details and click “Update” button | Displays a message “Selected Record Updated” | Displays a message “Selected Record Updated” | Pass |
|  | 12.2 |  | Click the update button without selecting an supplier | Displays an error message | Displayed an error message | Pass |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 13 | 13.1 | Delete Supplier | Selecting an supplier and click “Delete” | Displays a message “Record deleted from the table” | Displayed a message “Record deleted from the table” | Pass |
|  | 13.2 |  | Click the “Delete” button without selecting supplier | Record will not be deleted | Record will not be deleted | Pass |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 14 | 14.1 | Add Distribution details | Give all the distribution details and click “Add” button | Displays a message “New distribution added into the table” | Displayed a message “New distribution added into the table” | Pass |
|  | 14.2 |  | Giving other details except Dealer and Distribution ID and click “Add” button | Displays an error message “You are not allow to add a new record without Distribution ID and Dealer” | Displays an error message “You are not allow to add a new record without Distribution ID and Dealer” | Pass |
| 15 | 15.1 | Update Distribution | Selecting a detail from the table, updating details and click “Update” button | Displays a message “Selected Record Updated” | Displays a message “Selected Record Updated” | Pass |
|  | 15.2 |  | Click the update button without selecting an item | Displays an error message | Displayed an error message | Pass |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 16 | 16.1 | Delete Item | Selecting a detail and click “Delete” | Displays a message “Record deleted from the table” | Displayed a message “Record deleted from the table” | Pass |
|  | 16.2 |  | Click the “Delete” button without selecting a distribution detail | Record will not be deleted | Record will not be deleted | Pass |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 17 | 17.1 | Add Customer | Give all the customer details and click “Add” button | Displays a message “New customer added into the table” | Displayed a message “New customer added into the table” | Pass |
|  | 17.2 |  | Giving other details except Name and Customer ID and click “Add” button | Displays an error message “You are not allow to add a new record without Customer ID and Name” | Displays an error message “You are not allow to add a new record without Customer ID and Name” | Pass |
| 18 | 18.1 | Update Customer | Selecting an customer from the table, updating details and click “Update” button | Displays a message “Selected Record Updated” | Displays a message “Selected Record Updated” | Pass |
|  | 18.2 |  | Click the update button without selecting an customer | Displays an error message | Displayed an error message | Pass |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 19 | 19.1 | Delete Customer | Selecting an Customer and click “Delete” | Displays a message “Record deleted from the table” | Displayed a message “Record deleted from the table” | Pass |
|  | 19.2 |  | Click the “Delete” button without selecting Customer | Record will not be deleted | Record will not be deleted | Pass |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 20 | 20.1 | Add Item to bill | Give all the item details and click “Add” button | Items will be added to table | Items added to the table | Pass |
|  | 20.2 |  | Click “Add” button by missing some details | Items will be not added | Items not added | Pass |
| 6 | 6.1 | Update bill | Selecting an item from the table, updating details and click “Update” button | Displays a message “Selected Record Updated” | Displays a message “Selected Record Updated” | Pass |
|  | 6.2 |  | Click the update button without selecting an item | Displays an error message | Displayed an error message | Pass |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 7 | 7.1 | Delete bill | Selecting a bill and click “Delete” | Displays a message “Record deleted from the table” | Displayed a message “Record deleted from the table” | Pass |
|  | 7.2 |  | Click the “Delete” button without selecting bill | Record will not be deleted | Record will not be deleted | Pass |

# Task 06



## Deployment Techniques

After testing and correcting errors of the application, it should be deployed to the Total fitness pharmacy. It means the application should be ready to use for the organization in this deployment stage. After the application is ready to use it should be installed to the computer and should be tested to check whether it is operating accurately. There are different types of deployment techniques.

* Direct implementation
* Parallel Implementation
* Pilot Implementation
* Phased Implementation

1. **Direct Implementation**

This type of deployment or implementation technique removes the existing system, whether its manual or computerized and implements the new system for the use. The advantages of this system are it takes the minimal time and effort to implement, the new system is up and running immediately. This method is quite hard because the users are used for the old system so it would be quite harder for them to use the new system at the beginning. Also if there is a fault or error in the new system we can’t go back again to the old system.

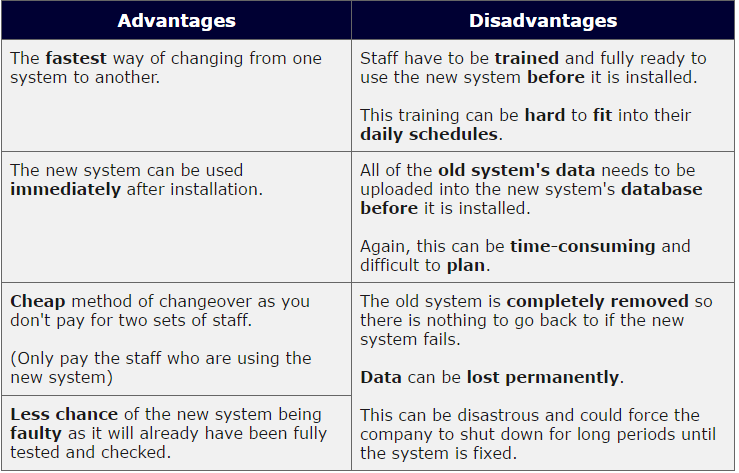


Table - Direct Advantages and Disadvanatges

1. **Parallel Implementation**

Parallel implementation means implementing the existing system and the new computerized system parallel for a particular duration. If the new system is successful, the old system can be removed and we can continue the new system. If there is a fault in the new system, we can remove it easily because the data will be too input to the old system and new system at the same time. So, there won’t be a data loss.



Table 4 - Parallel Advantages and Disadvantages

1. **Pilot Implementation**

In this technique a selected part of the old system is directed to the new system and implemented on by one and then rolling it out to the other departments. By doing this technique we can identify errors easily one by one each component, so don’t have to remove the whole system. But this technique consumes more time because we are implementing only selected parts of the components for a short period of time and checks whether it works fine. Doing this one by one takes a lot of time.

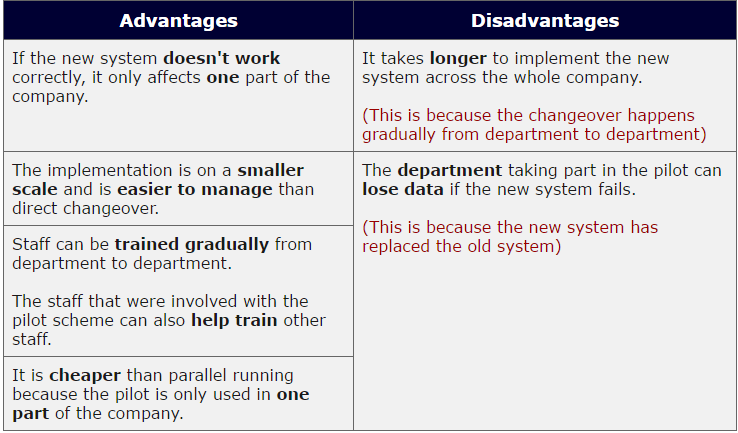


Table - Pilot Advantages and Disadvanatges

1. **Phased Implementation**

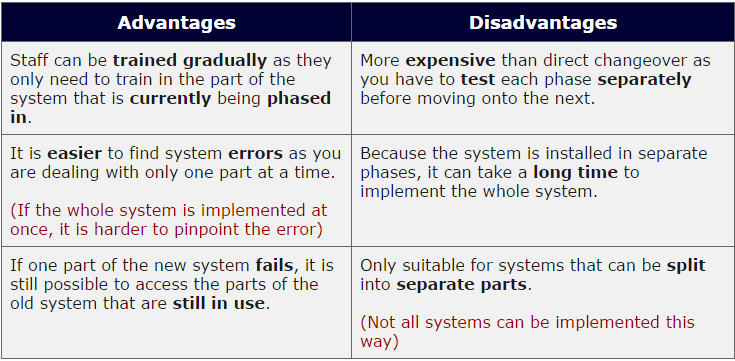
The selected parts of the old system are established in the new system step by step. The new system is implemented one part at a time in phases.

Table - Phased

So, we compared each and every deployment technique. Now, we have to select a one of these techniques for our application to deploy or implement for the Total Fitness pharmacy. When we consider about Direct Implementation it’s the fastest and he cheapest implementation technique. We have to remove the old system and implement the new computerized system. But this quite time consuming because all the old system’s data should be transferred to new system. The staff needs to be trained for the new system before it’s been installed. Unfortunately, if there was an error in the new system the data will be permanently will be lost. So, direct technique will be a risky one for the deployment of the application.

If we look at parallel implementation, in this both the systems are implemented parallel for short period. All the data inputs and outputs are done by both the systems. So if something happens to the new system, we don’t need to be worried about the data because we have used both the systems to input data. Also, the staff can be trained gradually. But this technique is quite expensive as we have to handle two systems.

Then in pilot implementation a selected part of the old system is replaced by the new system. If there’s any fault in the application, it will be affected to one part of the organization only. Also, this is in small scale and easy to manage. We have to train the staff gradually and this technique is cheap too. But there are some disadvantages too in this technique. It’s time consuming because you have to replace part by part of the new system to the whole company. The risk is if there was any fault found on one of the selected part which have been implemented, the data will be lost on the users who used them.

Finally, when we look at phased implementation, step by step selected parts are implemented. It’s quite similar to the previous technique. It’s easy to find errors because we are dealing one part at a time. But this technique isn’t suitable for systems which cannot be separated into parts. Also its time consuming.

So after examining these four techniques the learner decides to choose Parallel deployment technique for the Total Fitness Pharmacy application deployment. It is because we are deploying the new system without removing the old one and running both parallel. So the data won’t be lost if the new system malfunctions.

# Conclusion

Finally, we have created an application for the Total Fitness pharmacy. Before creating the system, we had to select a suitable platform for the application. We compared some operating systems like Microsoft Windows, Mac OS and Linux and selected Microsoft Windows as the suitable platform. Then we compared programming languages and IDE’s’ and selected C#.net and Visual Studio as the programming language and the IDE.

Next, we drew Entity Relationship Diagram to identify relationships, attributes and entities and also we drew the Wireframe Design to get a sketch of the interfaces that we are going to design. Then we developed a suitable application based on the designs using C#.net language and Visual Studio. We showed how the application has been developed by adding screenshots of the interface and source codes. We demonstrated the system too.

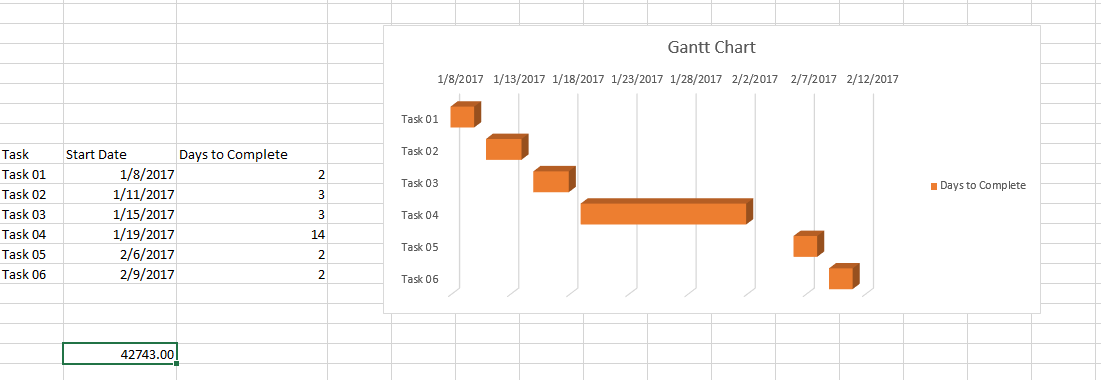
Then we created a user manual for the users who are going to use this system. Then, we test the whole system. We mentioned about system testing techniques and drew the test case. Finally, we discussed about deployment techniques for the system which should be implemented for the pharmacy and finally choose parallel technique for the system deployment.

Now we are satisfied that we have built up a good application for the Total Fitness pharmacy.

# Recommendation

In this section we are going to discuss some recommendations for the application. So the learner would like to recommend some ideas for his system. As we haven’t built up reports the learner suggest to build reports using report building tool.

# Gantt Chart



# References

Webopedia (no date) *What is C# programming language? Webopedia definition*. Available at: http://www.webopedia.com/TERM/C/C\_sharp.html (Accessed: 8 January 2017).

Technopedia (2017) *What is an integrated development environment (IDE)? - definition from Techopedia*. Available at: https://www.techopedia.com/definition/26860/integrated-development-environment-ide (Accessed: 29 January 2017).

BBC Bitesize (2017) *Programming software and the IDE*. Available at: http://www.bbc.co.uk/education/guides/zgmpr82/revision/ (Accessed: 29 January 2017).

RJ Systems (2003) *Windows pros and cons*. Available at: http://www.rjsystems.nl/en/3200.php (Accessed: 31 January 2017).

Soffar, H. (2016) *Linux operating system advantages and disadvantages | science online*. Available at: http://www.online-sciences.com/computer/linux-operating-system-advantages-and-disadvantages/ (Accessed: 11 February 2017).

Inc, B.H. (2010) *A look at the advantages and disadvantages of Mac OS*. Available at: http://www.brighthub.com/computing/mac-platform/articles/73326.aspx (Accessed: 31 January 2017).

Parker, J. (2013) *Types of system testing - Enfocus solutions Inc*. Available at: http://enfocussolutions.com/types-of-system-testing/ (Accessed: 9 February 2017).

Software Testing Help (2016) *Types of software testing and definitions of testing terms — software testing help*. Available at: http://www.softwaretestinghelp.com/types-of-software-testing/ (Accessed: 9 February 2017).

Software Testing Fundamentals (2016) *Black box testing*. Available at: http://softwaretestingfundamentals.com/black-box-testing/ (Accessed: 9 February 2017).

Software Testing Fundamentals (2016) *Test case*. Available at: http://softwaretestingfundamentals.com/test-case/ (Accessed: 9 February 2017).