Pharma Management System

Integrated Capability Test

Contents

[**Pharma Management System** 1](#_Toc43986459)

[Table of contents 2](#_Toc43986460)

[**1.0** **Introduction** 3](#_Toc43986461)

[**1.1** **Purpose of this document** 3](#_Toc43986462)

[**1.2** **Project Overview** 3](#_Toc43986463)

[**1.2.1** **Use Case Diagram** 3](#_Toc43986464)

[**1.3** **Scope** 3](#_Toc43986465)

[**1.4** **Software Requirement** 3](#_Toc43986466)

[**2.0 Technical Guidelines** 4](#_Toc43986467)

[**3.0** **Design for adding new Pharmaceutical items to the database** 5](#_Toc43986468)

[**3.1** **Requirement flow** 5](#_Toc43986469)

[**4.0** **Design for viewing pharmaceutical Items** 6](#_Toc43986470)

[**4.1** **Requirement flow** 6](#_Toc43986471)

[**5.0** **Database schema.** 7](#_Toc43986472)

[**5.1 The following data base schema is used for the application.** 7](#_Toc43986473)

[**5.2 Configuring connection string in app.config** 7](#_Toc43986474)

[**6.0** **Standards and Guidelines** 7](#_Toc43986475)

[**7.0** **Change Log** 8](#_Toc43986476)

# **Introduction**

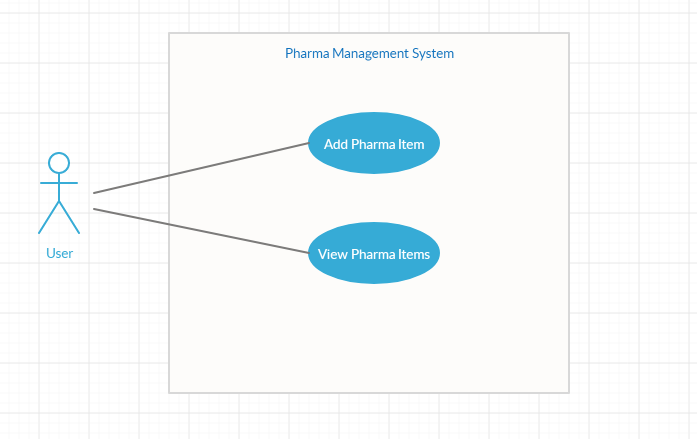
## **Purpose of this document**

The purpose of this document is to define the scope and requirement of the Pharma Management System application.

## **Project Overview**

The project involves creating a pharma management system. A user can enter the details of pharmaceutical items which will be stored in the database. Details of the items stored can be viewed from the database.

## **Use Case Diagram**

****

## **Scope**

Creation of an application for Pharma management system

## **Software Requirement**

|  |  |
| --- | --- |
| **Tools/APIs** | .Net Framework 4.6  ADO.NET Framework  SQL Server 14.x |
| **Server** | Embedded |

# **2.0 Technical Guidelines**

The Pharma Management application consists of three projects.

* 1. A console application named **PharmaManagement.Client.**

This console application contains a Program class with the following static methods apart from the Main method.

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | **Input Argument** | **Return Type** | **Description** |
| AddNewPharmaItem | N/A | N/A | To add a pharma item to the database |
| ViewPharmaItems | N/A | N/A | To view pharma items |
| ViewProductsAfterDiscount | Decimal cost, double discount | N/A | To get the discounted price details of the pharma items which has a price greater that the cost that is provided |

* 1. A class library project named **PharmaManagement.Model.**

This class library project should contain the following classes with properties.

* + 1. **PharmaItem**

|  |  |
| --- | --- |
| **Property Name** | **Data Type** |
| Id | string |
| Name | string |
| Cost | double |
| QuantityInStock | int |

* + 1. **PharmaItemType**

|  |  |
| --- | --- |
| **Property Name** | **Data Type** |
| Id | string |
| Name | string |

* + 1. **Supplier**

|  |  |
| --- | --- |
| **Property Name** | **Data Type** |
| Id | string |
| Name | string |
| Address | string |

* 1. A class library project named **PharamaManagement.DAL.**

This class library project consists of a class named **PharmaDAL** with the following methods.

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | **Input Arguments** | **Return Type** | **Description** |
| InsertPharmaItem | |  |  | | --- | --- | | **Name** | **Type** | | id | string | | name | string | | cost | double | | quantityInStock | int | | typeId | int | | supplierId | string | | bool | Inserts a new pharma item |
| GetPharmaItemDetails | N/A | List<PharmaItem> | Gets all pharma item details |
| GetPharmaTypes | N/A | List<PharmaItemType> | Gets all PharmaItemTypes |
| GetSuppliers | N/A | List<Supplier> | Gets all supplier details |
| GetPharmaItemDetailsAfterDiscount | |  |  | | --- | --- | | **Name** | **Type** | | Cost | decimal | | Discount | Double | | List<PharmaItem> | Gets all pharma items whose cost is greater than the input cost and then return the cost after applying the discount. For example, if the cost is 1000 and discount is 13 display all pharma items whose cost is greater than 1000 and return them after applying the discount of 13 % on cost. |

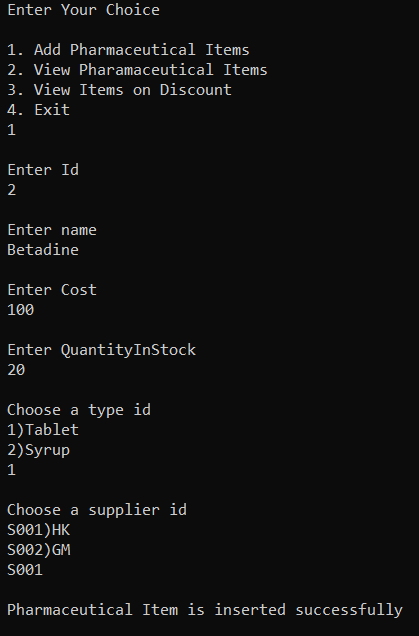
# **Design for adding new Pharmaceutical items to the database**

On launching the application, the user will be asked to enter one of the following options.

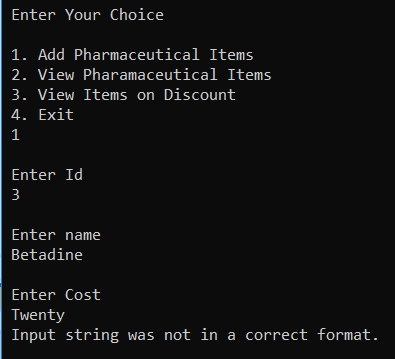
1. Add Pharmaceutical Items
2. View Pharmaceutical items
3. Exit

## **Steps for adding pharmaceutical items to database**

* + 1. Accept input from user from the screen
    2. Choose option 1 for adding new Pharmaceutical item details and then enter Id, name, cost, items in stock.
    3. Choose a pharma item type from the listed item type
    4. Choose supplier id from the given list.
    5. Display the message ‘Pharmaceutical Item is inserted successfully’.



* + 1. If an error occurs, exception message should be displayed in the console.

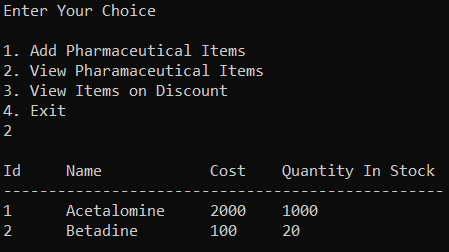


# **Design for viewing pharmaceutical Items**

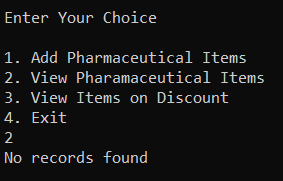
## **Requirement flow**

**Steps for viewing Pharmaceutical Items**

* + 1. Accept input from user from the screen
    2. Choose option 2 for viewing pharmaceutical items
    3. Display the Id, name, cost and quantity in stock of the pharmaceutical item.



* + 1. If no records are found display ‘No records found’ in the console.

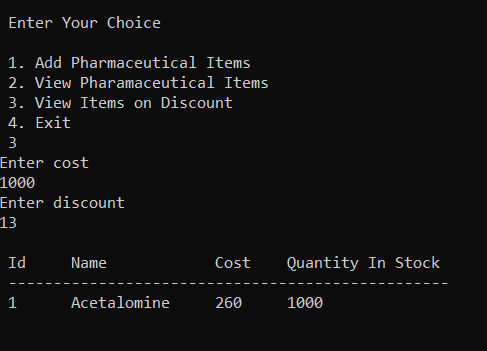


# **Design for viewing pharmaceutical after discount**

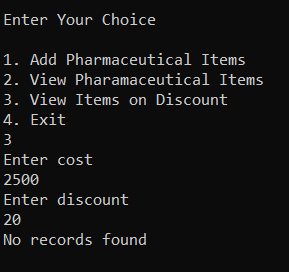
* 1. Requirement flow

**Steps for viewing Pharmaceutical Items**

* + 1. Accept input from the user from the screen.
    2. Choose option 3 for viewing the pharma items after discount.
    3. Accept first cost and then discount as input
    4. Display the Id, name, cost and quantity in stock of the pharmaceutical item.



* + 1. If no records are found display ‘No records found’ in the console.



# **Database schema.**

## **5.1 The following data base schema is used for the application.**

****

## **5.2 Configuring connection string in app.config**

Create a connection string section in the app.config of the application. Connection string name should be named as **PharmalManagementConnectionString.** The sample given below can be used as reference.

<connectionStrings>

<add name="PharmalManagementConnectionString" connectionString="Data Source=localhost; Initial Catalog=PharmaDb;Integrated Security=True"/>

</connectionStrings>

# **Standards and Guidelines**

* 1. Appropriate naming conventions should be following while naming the different types in the application.
  2. Use exception handling in the data access layer and client project
  3. Connection string should be set in the app.config file

# **Change Log**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Changes Made** | | | |
| V1.0.0 | Initial baseline created on <date> by <author> | | | |
|  | <Please refer the configuration control tool / change item status form if the details of changes are maintained separately. If not, the template given below needs to be followed> | | | |
| **Section No.** | **Changed By** | **Effective Date** | **Changes Effected** |
|  |  |  |  |