

**Dharmsinh Desai University,Nadiad**

**Faculty of Technology,Department of Computer**

**Engineering**

B.Tech CE Semester – VI

Subject: Service Oriented Computing

Project Title:

“BSMS”

Submitted By

Shah Krupal (18CEUOD005) (CE123)

Suvagiya Akash (18CEUOD006)(CE131)

**Guided by**

**Prof. Apurva Mehta**



**Dharmsinh Desai University,Nadiad**

**Faculty of Technology,Department of Computer**

**Engineering**

**CERTIFICATE**

This is to certify **Service Oriented Computing** project entitled “**BSMS**” is the bonofied report of work carried out by **Krupal Shah (18CEUOD005)(CE123)** and **Akash Suvagiya (18CEUOD006)(CE131)** of Computer Engineering,Semester VI, academic year 2019-2020,under our supervision and guidance.

**Guide**

Prof. Apurva Mehta

Assistant Professor of Department of

Computer Engineering,

Dharmsinh Desai

University, Nadiad

**HOD**

Dr. C. K. Bhensdadia

Head of Department of

Computer Engineering,

Dharmsinh Desai

University, Nadiad

**INDEX**

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **Content** | **Page No.** |
| **1.** | **Abstract** | **1** |
| **2.** | **Introduction of Project** | **3** |
|  | 2.1 Brief Introduction | **4** |
|  | 2.2 Technology and Tools Used | **5** |
| **3.** | **Software Requirement Specification** | **7** |
|  | 3.1 Scope | **8** |
|  | 3.2 System Functional Requirements | **8** |
|  | 3.3 Other Non-Functional Requirements | **14** |
| **4.** | **Design** | **15** |
|  | 4.1 Use Case Diagram | **16** |
|  | 4.2 Class Diagram | **17** |
|  | 4.3 Sequence Diagram | **18** |
|  | 4.4 Activity Diagram | **20** |
|  | 4.5 State Diagram | **23** |
|  | 4.6 E-R Diagram | **24** |
|  | 4.7 Data Dictionary | **25** |
| **5.** | **Implementation Details** | **28** |
|  | 5.1 Modules Description | **29** |
| **6.** | **Testing** | **30** |
| **7.** | **Screenshots** | **34** |
| **8.** | **Conclusion** | **41** |
| **9.** | **Limitations and Future Extension** | **43** |
| **10.** | **Bibliography** | **45** |

1. **Abstract**

**Abstract**

BSMS is Client-Server Desktop Application build for managing And handling the Bill and Stock Which are in any store, shopping mall, whole sellers and retailers which have many branches and wide business. BSMS is develop in Wcf and Windows Form App , Wcf used for server side to provide services and client are made by using windows form.

BSMS has functionality like Manage employees and Branch. BSMS can also manage Sales and Stock. BSMS can also generate Bills and maintain brief record of all transaction. BSMS provide accurate fact and figure related to stock and sales and bills which is made by using BSMS.

BSMS has Client side App develop in Window Form one is for Main Admin, second for Branch Level Admin and third for Biller who can able to generate bill. So that BSMS has wider Scope it used in big shops and Malls and etc.

1. **Introduction of Project**
   1. **Brief Introduction**

|  |  |
| --- | --- |
| Project Name: | BSMS |
| Project Definition/Aim: | The main aim of project is to provide better maintainability of bills and stock and sales record of big malls and super stores. |
| Develop For : | Bill Malls ,Whole Sellers |
| Front End : | Windows From Application(with C#) |
| Back End : | Wcf(Windows Communication Foundation) |
| Other Technologies : | * Microsoft Sql Server |
| Documentation tools: | MS Word |
| Submitted To: | Any Mall, super Stores, any store who have multiple Branches |
| IDE: | Microsoft Visual Studio(2015-community) |

* 1. **Tools and Technology Used**

**Windows Form App**:

**Windows Forms** (WinForms) is a [free and open-source](https://en.wikipedia.org/wiki/Free_and_open-source) [graphical](https://en.wikipedia.org/wiki/Graphical_user_interface) (GUI) [class library](https://en.wikipedia.org/wiki/Library_(computing)) included as a part of [Microsoft](https://en.wikipedia.org/wiki/Microsoft) [.NET Framework](https://en.wikipedia.org/wiki/.NET_Framework)  providing a platform to write rich client applications for desktop, laptop, and tablet PCs. While it is seen as a replacement for the earlier and more complex [C++](https://en.wikipedia.org/wiki/C%2B%2B) based [Microsoft Foundation Class Library](https://en.wikipedia.org/wiki/Microsoft_Foundation_Class_Library), it does not offer a comparable paradigm and only acts as a platform for the user interface tier in a [multi-tier](https://en.wikipedia.org/wiki/Multitier_architecture) solution

**Windows Communication Foundation:**

Windows Communication Foundation (WCF) is a framework for building service-oriented applications. Using WCF, you can send data as asynchronous messages from one service endpoint to another. A service endpoint can be part of a continuously available service hosted by IIS, or it can be a service hosted in an application. An endpoint can be a client of a service that requests data from a service endpoint. The messages can be as simple as a single character or word sent as XML, or as complex as a stream of binary data. A few sample scenarios include:

* A secure service to process business transactions.
* A service that supplies current data to others, such as a traffic report or other monitoring service.
* A chat service that allows two people to communicate or exchange data in real time.
* A dashboard application that polls one or more services for data and presents it in a logical presentation.
* Exposing a workflow implemented using Windows Workflow Foundation as a WCF service.
* A Silverlight application to poll a service for the latest data feeds.

While creating such applications was possible prior to the existence of WCF, WCF makes the development of endpoints easier than ever. In summary, WCF is designed to offer a manageable approach to creating Web services and Web service clients.

## **Features of WCF**

**Service Orientation**

**Interoperability**

**Multiple Message Patterns**

**Data Contracts**

**Security**

**Reliable and Queued Messages**

**Durable Messages**

**AJAX and REST Support**

1. **Software Requirement Specifications** 
   1. **Scope**

**The scope of this system is very large so it used in any mall, large scale shop which has many branches and also retailer, Whole sellers etc used this system.**

* 1. **System Functional Requirements**

BILL AND STOCK MANAGEMENT SYSTEM

**MAIN ADMIN**

R1: LOGIN

The admin can give the credentials and login to the portal.

Input:-Enter credentials

Output:-Login Successfully

R2: MANAGE BRANCH ADMIN

The main admin is resides at the server side and can manage the multiple branches(crud operations) and their admins.

R2.1 Add Branch Admin

Input:-Enter Branch Admin Detail

Output:-Branch admin added

R2.2 Remove Branch Admin

Input:-Enter Branch Admin Detail

Output:-Branch admin Removed

R2.3 Update Branch Admin

Input:-Enter Branch Admin Detail

Output:-Branch admin Updated

R2.4 View All Admins

Output:-List of Branch Admins.

R3: MANAGE STOCKS

The main admin can manage the stocks at different branches like available stocks, sold out stocks etc.

R3.1 Add Stock on Particular Branch

Input:-Enter Stock Details

Output:-Stocks Added Successfully

R3.2 Remove Stock On Particular Branch

Input:- Enter Stock Details

Output:-Stock Removed Successfully

R3.3 Update Stock on Particular Branch

Input:-Enter Stock Details

Output:-Stock Updated Successfully

R3.4 View All Stocks

Output:-Statistical Information of Total Stock available at all branch

R4: MANAGE BILL

It can manage the bills which are being generated at all the branch and can do some statistical analysis.

R4.1 Remove Bills generated at Different Branch

Input:- Enter Bill No.

Output:-Bill Removed Successfully

R4.2 Update Bill

Input:- Enter Bill No and Details to update.

Output:-Bill Updated Successfully

R4.3 View All Bills

Output:-Information of all Bills of All Branches

R5: MANAGE BRANCH

The admin can manage branch like to add branch delete branch etc.(crud operations for branch).

R5.1 Add Branch

Input:-Enter Branch Detail

Output:-Branch added

R5.2 Remove Branch

Input:-Enter Branch Detail

Output:-Branch Removed

R5.3 Update Branch

Input:-Enter Branch Detail

Output:-Branch Updated

R2.4 View Branch Information

Input:-Enter Branch Id

Output:-Display Branch Information(To show selling, cost of selling etc.)

**BRANCH ADMIN**

R1: LOGIN

The branch admin can give the credentials and login to the portal

Input:-Enter credentials

Output:-Login Successfully

R2: MANAGE BILLER

The branch admin is resides within the particular branch and can manage the multiple billers(crud operations) .

R2.1 Add Biller

Input:-Enter Biller Detail

Output:-Biller Added Successfully

R2.2 Remove Biller

Input:-Enter Biller Detail

Output:-Biller Removed

R2.3 Update Biller

Input:-Enter Biller Detail

Output:-Biller Updated

R2.4 View All Billers

Output:-List of Billers.

R3: MANAGE STOCKS OF BRANCH

The branch admin can manage the stocks at their particular branches like available stocks, sold out stocks, which stock is required etc.

R3.1 Add Stock on Particular Branch

Input:-Enter Stock Details

Output:-Stocks Added Successfully

R3.2 Remove Stock On Particular Branch

Input:- Enter Stock Details

Output:-Stock Removed Successfully

R3.3 Update Stock on Particular Branch

Input:-Enter Stock Details

Output:-Stock Updated Successfully

R3.4 View All Stocks

Output:-Statistical Information of Total Stock available at all branch

R4: SELLING AT BRANCH

It can display the selling of the stocks at the branch e.g. how much selling is done last month, this month etc.

R4.1 Show Selling

Output:-Display the total selling of the Branch and Related Information

R4 MANAGE BILL

It can manage the bills which are being generated at the branch and can do some statistical analysis.

R4.1 Remove Bills generated at Branch

Input:- Enter Bill No.

Output:-Bill Removed Successfully

R4.2 Update Bill

Input:-Enter Bill No and Details to update.

Output:-Bill Updated Successfully

R4.3 View All Bills

Output:-Information of all Bills of All Branches

**BILLER**

R1: LOGIN

The biller can give the credentials and login to the portal to generate bills.

Input:-Enter credentials

Output:-Login Successfully

R2: GENERATE BILLS

After login the biller can enter the product id and can generate the bills for the different products for the customers.

Input:-Enter Product Information

Output:-Bill Generated

* 1. **Other Non-Functional Requirements**

**Performance and scalability.**How fast does the system return results. How much will this performance change with higher workloads.

**Portability and compatibility.**Which hardware, operating systems, browsers, and their versions does the software run on? Does it conflict with other applications and processes within these environments.

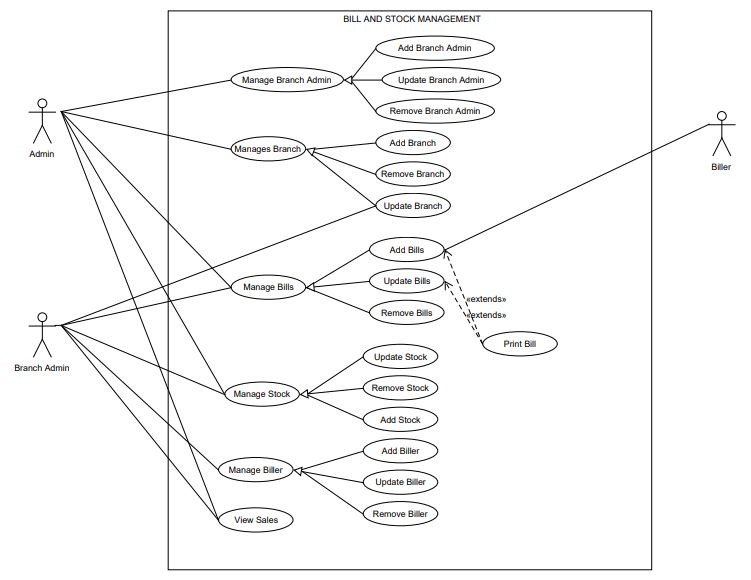
**Reliability, availability, maintainability.**How often does the system experience critical failures? and how much time is it available to users against downtimes.

**Security.**How are the system and its data protected against attacks.

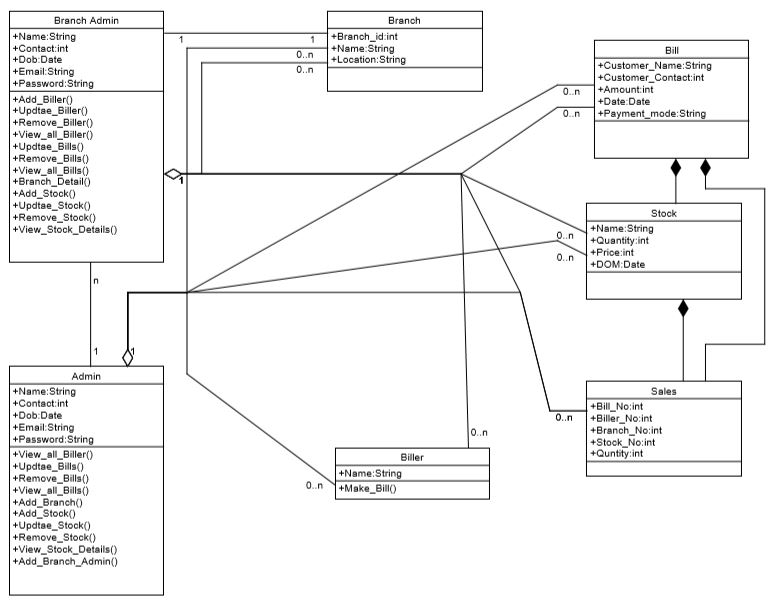
**Localization.**Does the system match local specifics.

**Usability.**How easy is it for a customer to use the system.

1. **Design**
   1. **Use Case Diagram**

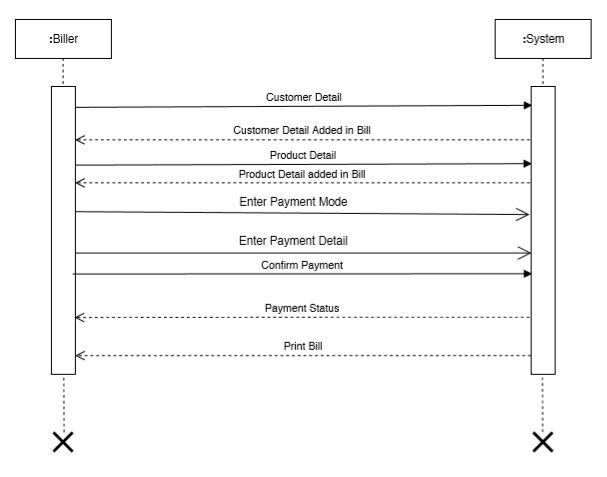
****

* 1. **Class Diagram**

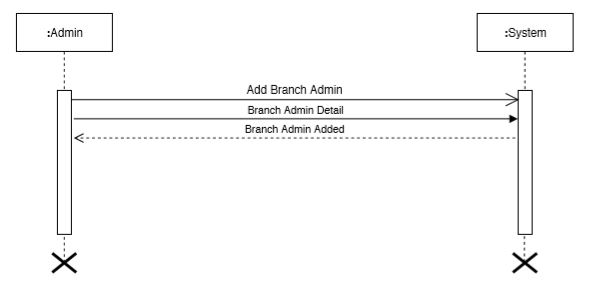
****

* 1. **Sequence Diagram**

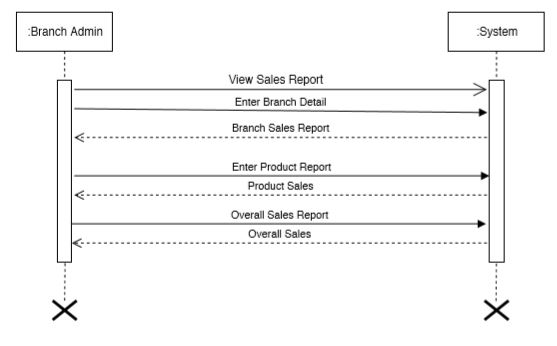
1. **Generate Bill**

****

1. **Add Branch**

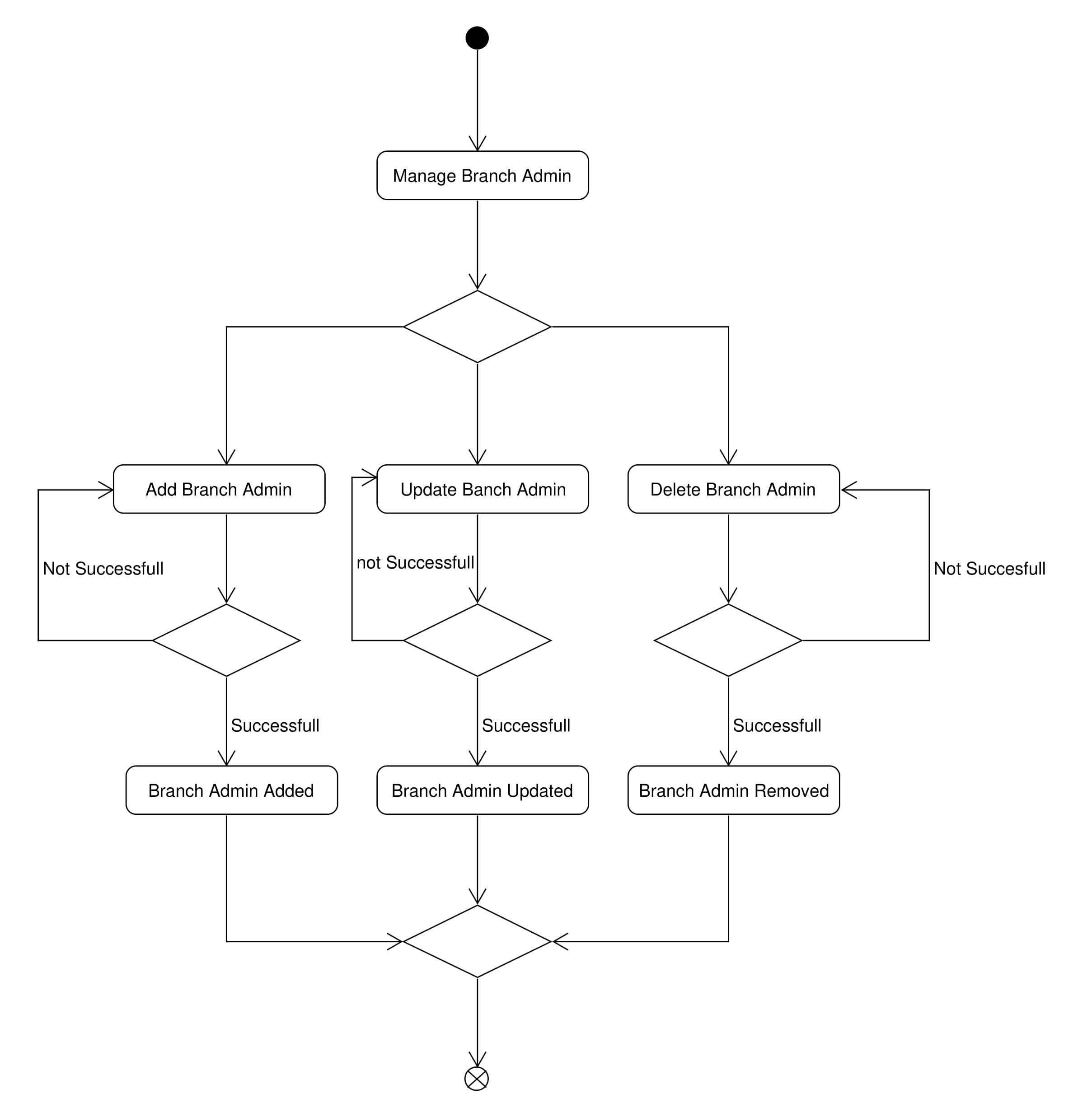
****

1. **View Sales**

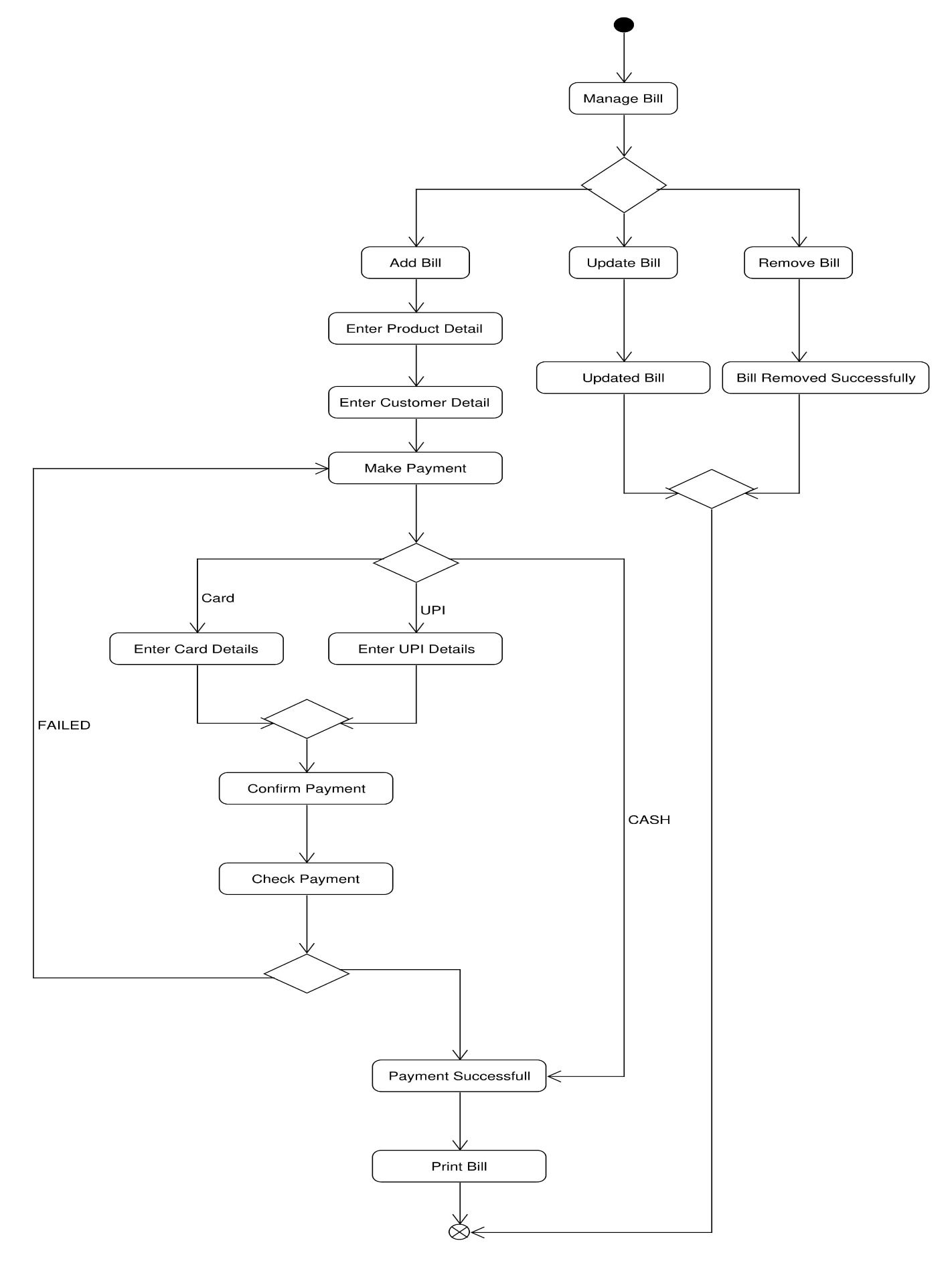
****

* 1. Activity Diagram

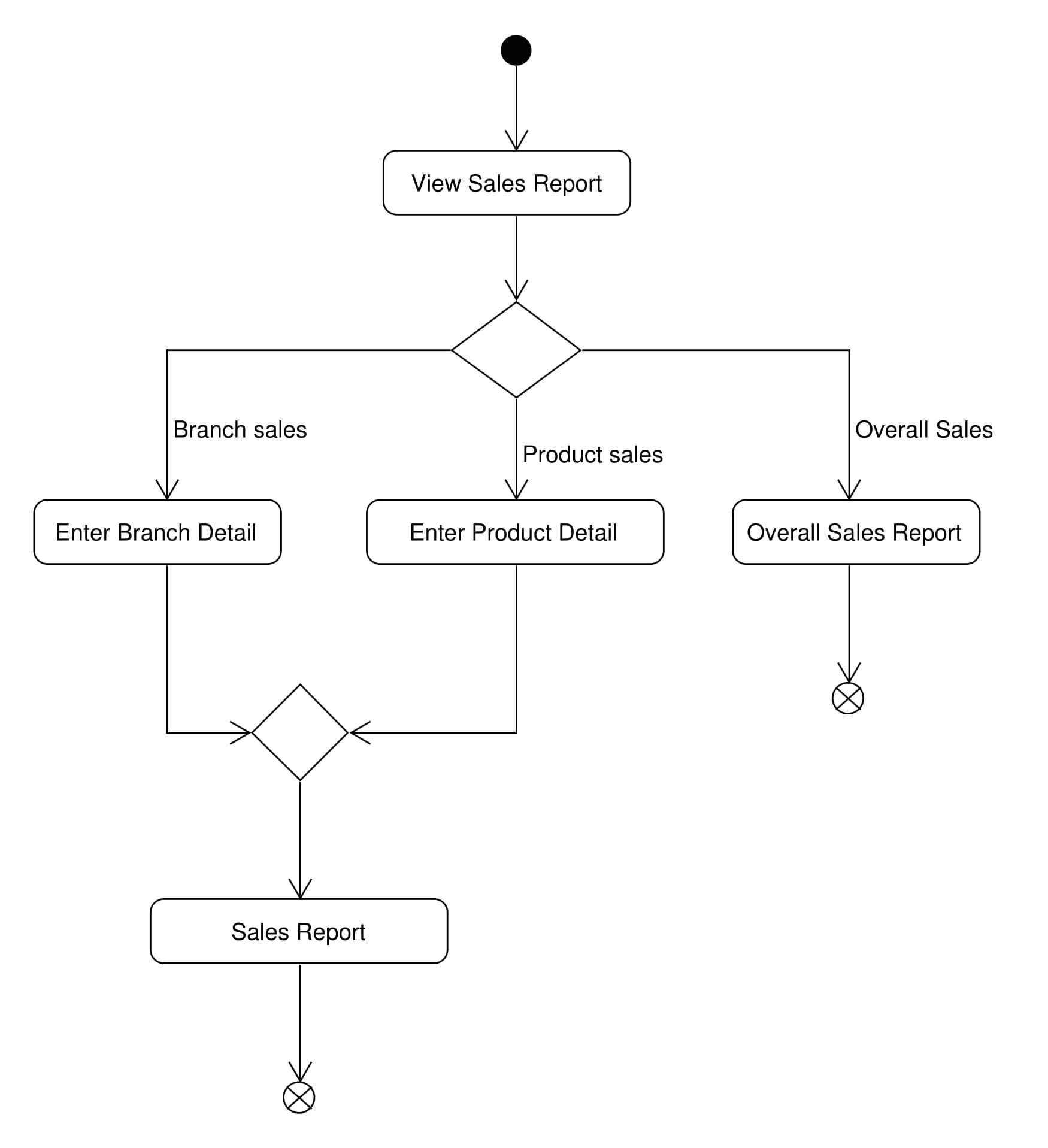
1. Crud Activity

****

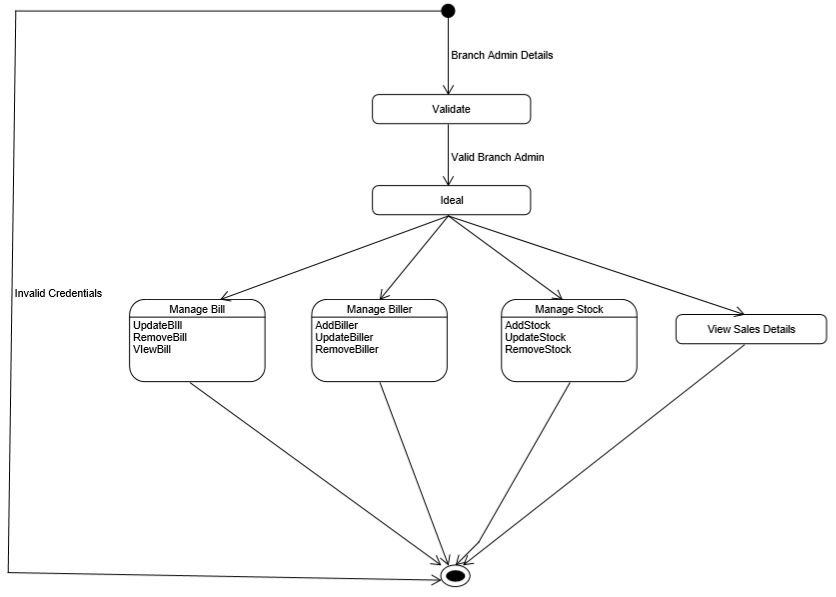
1. Bill Activity



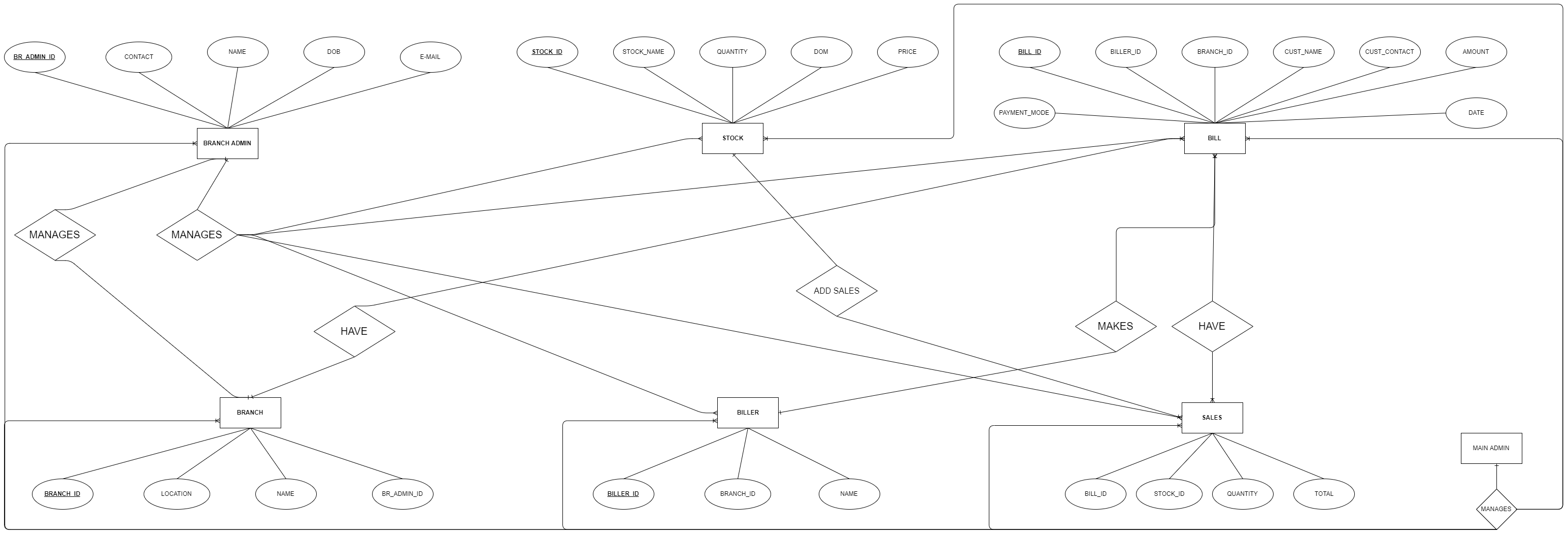
1. Sales Activity



* 1. State Diagram



* 1. E-R Diagram



* 1. **Data Dictionary**

1. **BRANCH ADMIN**

|  |  |  |  |
| --- | --- | --- | --- |
| NAME | DATATYPE | SIZE | DESCRIPTION |
| Br\_admin\_id | Int | 6 | Primary key |
| Name | Varchar | 30 |  |
| Contact | Int | 10 |  |
| Dob | Date |  |  |
| e-mail | Varchar | 40 |  |
| Password | Varchar | 8 |  |

1. **BRANCH**

|  |  |  |  |
| --- | --- | --- | --- |
| NAME | DATATYPE | SIZE | DESCRIPTION |
| Branch \_id | Int | 6 | Primary key |
| Br\_admin\_id | Int | 6 | Ref(BRANCH ADMIN) |
| Location | Varchar | 100 |  |

1. **BILL**

|  |  |  |  |
| --- | --- | --- | --- |
| NAME | DATATYPE | SIZE | DESCRIPTION |
| Bill\_id | Int | 6 | Primary key |
| Biller\_id | Int | 6 | Ref(BILLER) |
| Branch\_id | Int | 6 | Ref(BRANCH) |
| Customer\_name | Varchar | 40 |  |
| Customer\_Contact | Int | 10 |  |
| Amount | Int | 6 |  |
| Date | Date |  |  |
| Payment\_mode | Varchar | 10 | Values(card,cash,upi) |

1. **STOCK**

|  |  |  |  |
| --- | --- | --- | --- |
| NAME | DATATYPE | SIZE | DESCRIPTION |
| Stock\_id | Int | 6 | Primary key |
| Name | Varchar | 30 |  |
| Quantity | Int | 6 |  |
| Price | Int | 6 |  |
| Dom | Date |  |  |

1. **SALES**

|  |  |  |  |
| --- | --- | --- | --- |
| NAME | DATATYPE | SIZE | DESCRIPTION |
| Bill\_id | Int | 6 | Ref(BILL) |
| Stock\_id | Int | 6 | Ref(Stock) |
| Quantity | Int | 6 |  |
| Total | Int | 6 |  |

1. **BILLER**

|  |  |  |  |
| --- | --- | --- | --- |
| NAME | DATATYPE | SIZE | DESCRIPTION |
| Biller \_id | Int | 6 | Primary key |
| Branch\_id | Int | 6 | Ref(BRANCH) |
| Contact | Int | 10 |  |
| Dob | Date |  |  |
| e-mail | Varchar | 40 |  |
| Password | Varchar | 8 |  |

1. **Implementation Details**
   1. **Modules Description**
2. **Admin**

In this module we are provide functionality like manage branches, manage branch admin. And other functions of this modules like manage stock, provide details of sales and bills and other etc.

1. **Branch Admin**

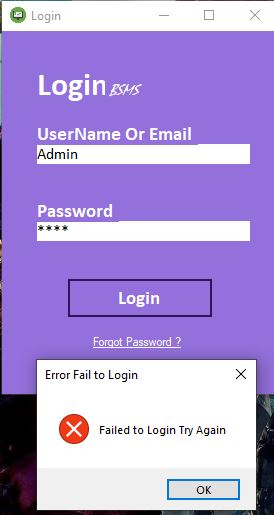
In this module we are provide functionality like Manage Biller , branch Admin can add ,update and delete branch Biller.also manage stock at branch, and view Bills and view sales in Branch.

1. **Biller**

In this module Biller can login and generate bill and view biller’s added bills.

1. **Testing**
2. **Login Page**

In Login page Whenever Biller, Admin and Branch Admin type wrong password than it would handle by system, system prompt Incorrect username and password



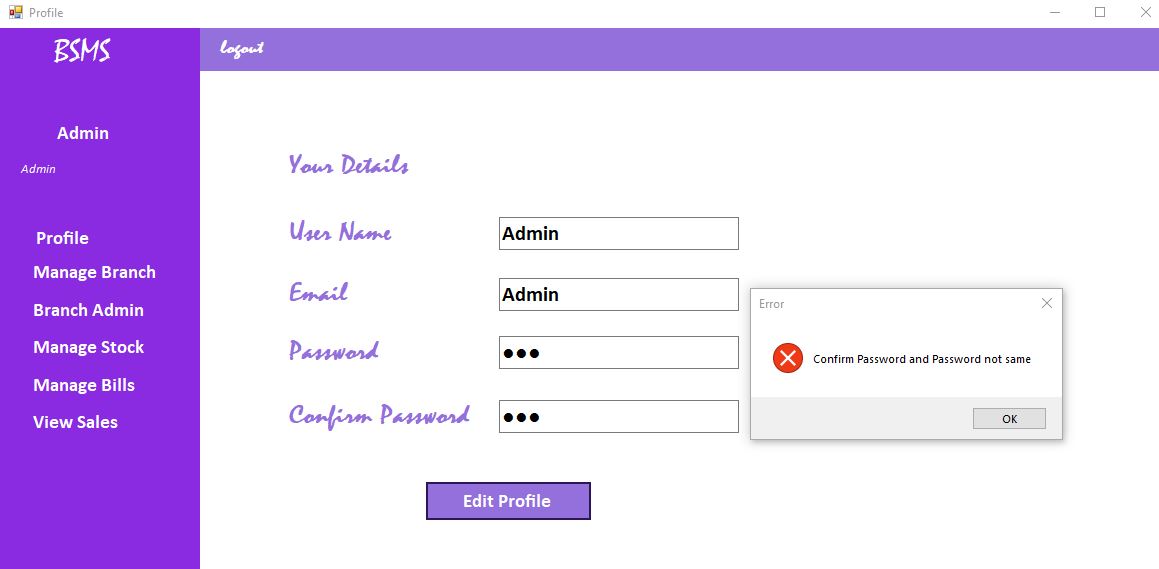
6.1 picture login

Any unauthorized user can prevent to access system.so that security of system is maintain.

1. **Profile page**

In this page system handle password and confirm password error

If the user want change profile than also password and confirm password is required.

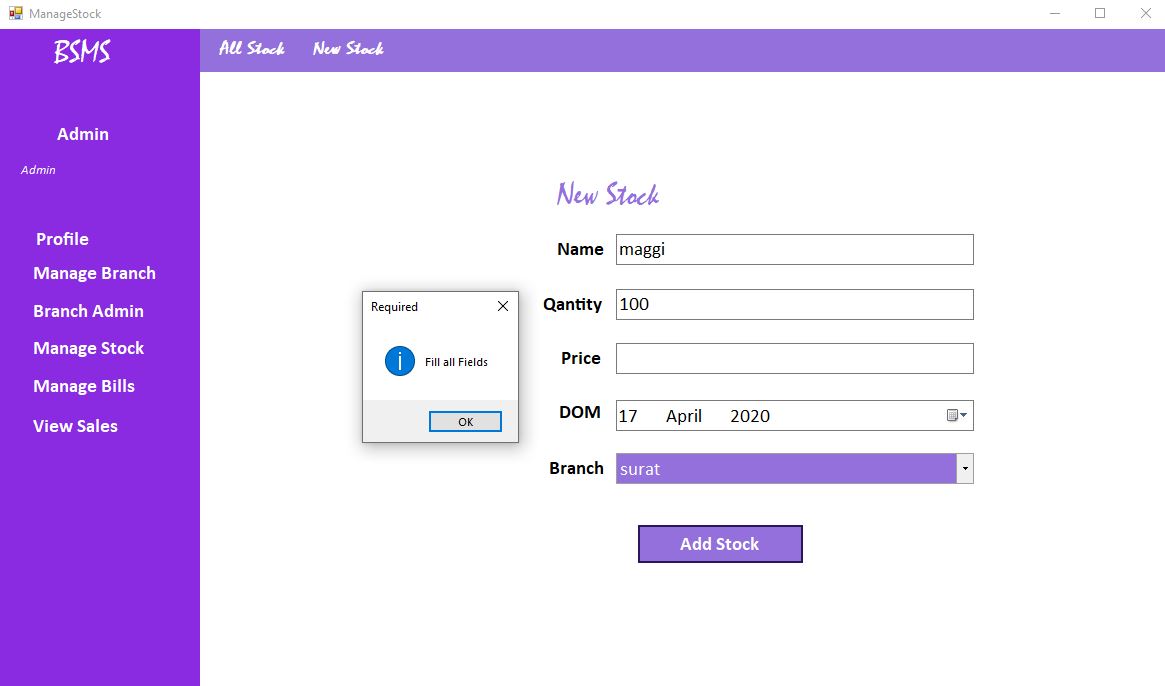


* 1. picture password & con-password error

User needs to enter password for update profile than only profile can edit.

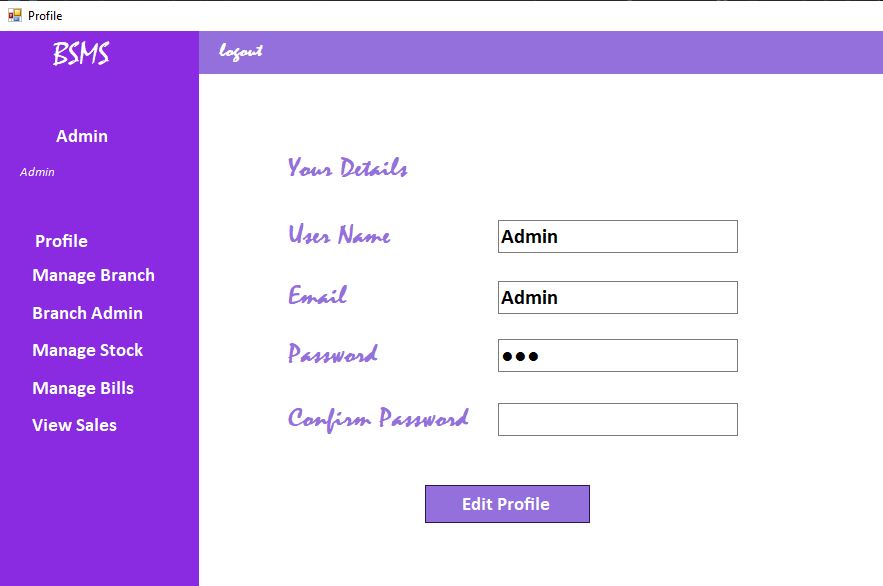
1. **All Page Required field error**

Whenever any system user need to manipulate data like edit or add new data in system and user may forgot enter some data than system handle error and prompt for required fields.



6.3 picture required field error page

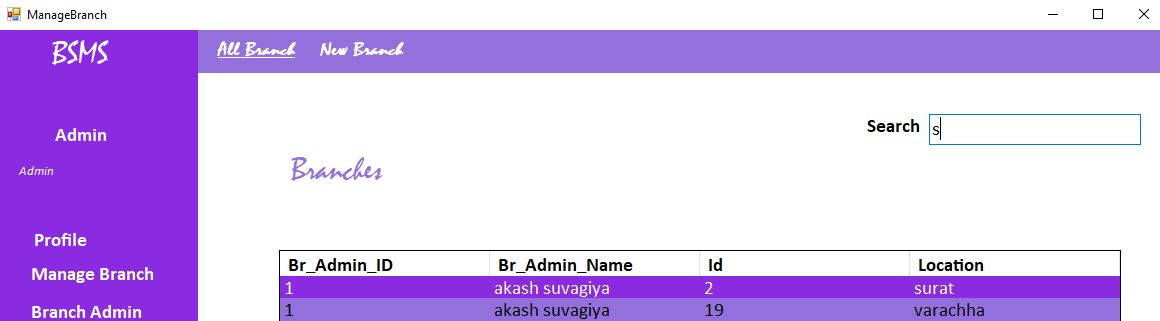
1. **Screenshots**
2. **Profile page**



7.1 picture Profile page

All system used can edit it’s own profile.

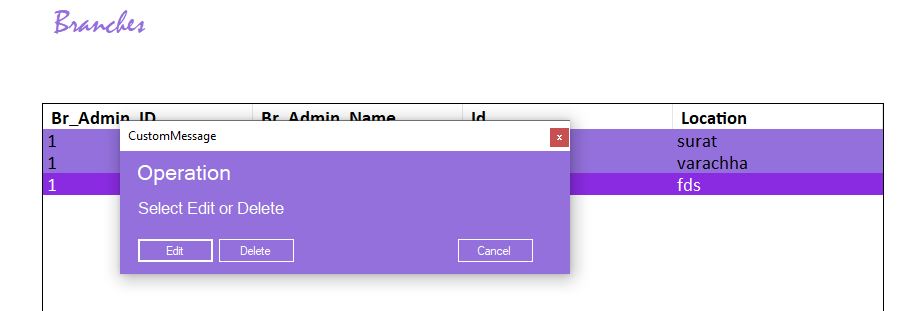
1. **Search**



* 1. Picture live Search functionality for all like branch, bill, stock etc.

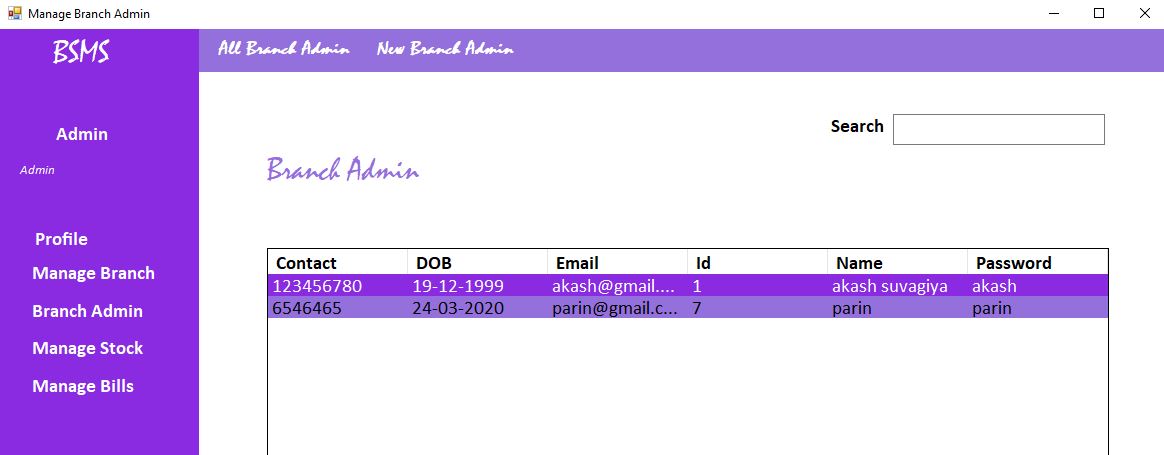
1. **Prompt**

Custom Message Box Design by our staff for custom dialog like edit and delete.



7.3 Picture Custom Message Box

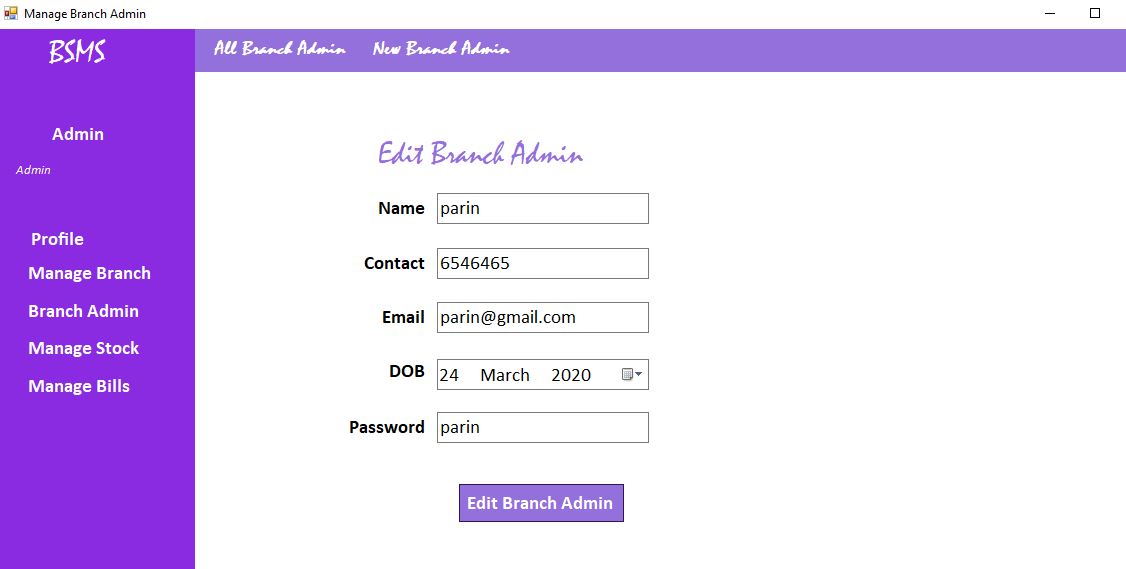
1. **Branch Admin page**



* 1. Picture Branch Admin Manage

1. **Edit Branch Admin**

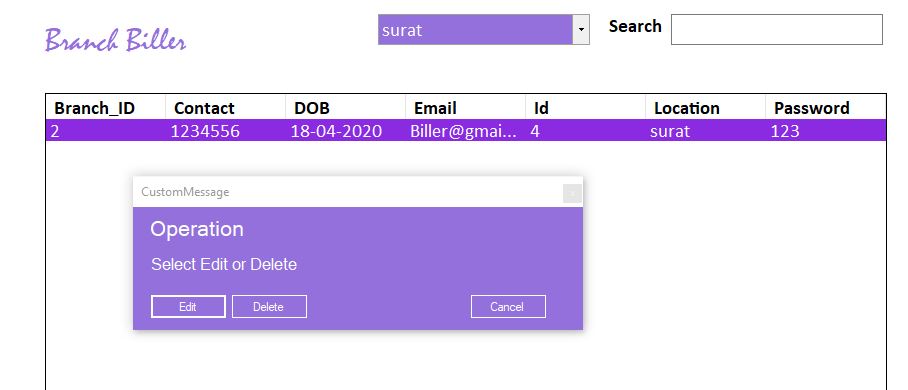
Add new Branch Admin and edit page .



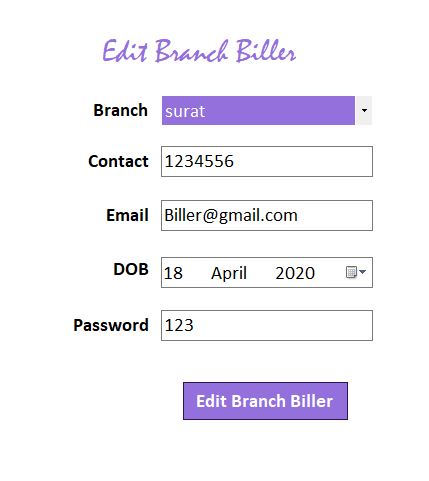
* 1. Picture Branch Admin Add & Edit

1. **Biller Add & Update and Delete**

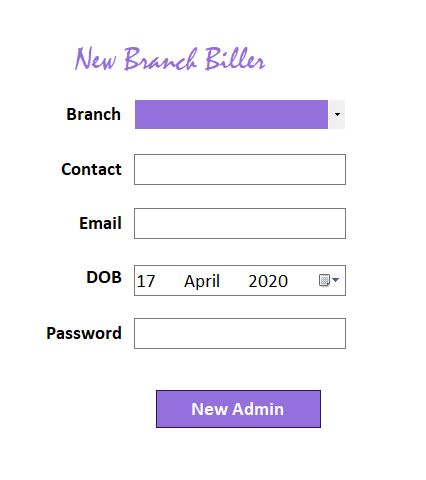
The snapshot contain add Bill, Update Biller ,View Biller and Delete Biller.



7.6.1 Picture Show bill and delete



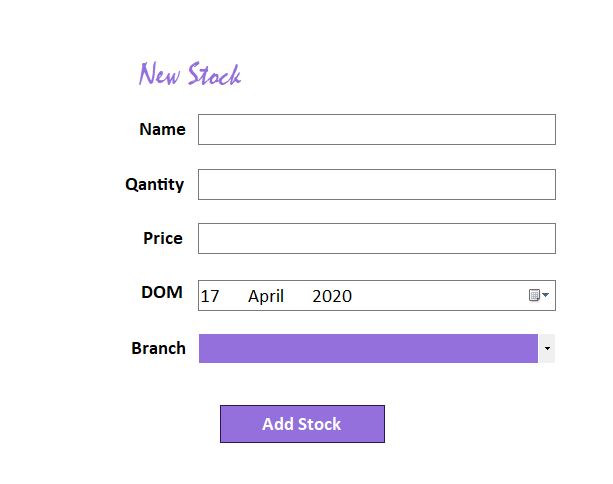
7.6.2 Picture Edit Biller page



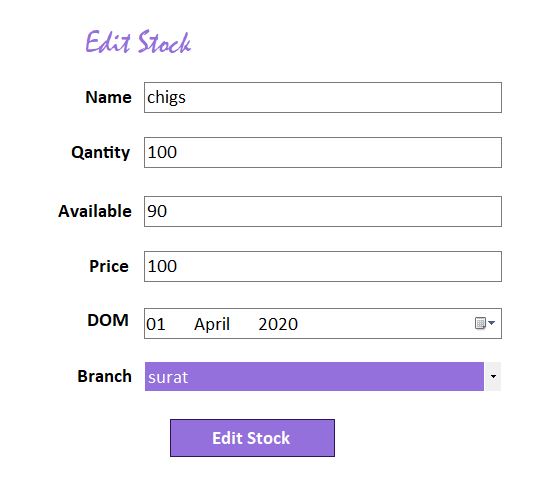
7.6.3 Picture Add Biller page

1. **Stock Management**

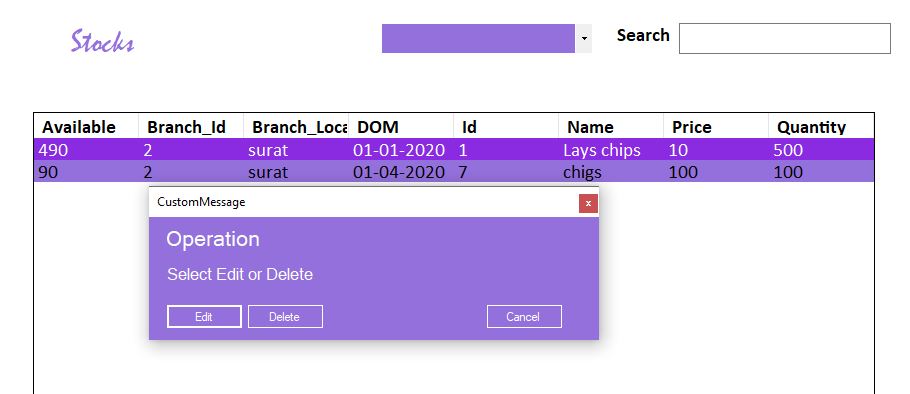
Add Stock, Modify Stock and View and Delete stock function snapshot.



7.7.1 Picture Add Stock



7.7.2 Picture Update Stock



7.7.3 Picture List & Delete Stock

1. **Conclusion**

**Conclusion**

The Functionality implemented in the system was done after understanding all the system modules according to the requirements.

Functionalities that are successfully implemented are:-

Login

Generate Bill

Manage Branch

Manage Bill

Manage Stock

Manage Biller

View Sales

Edit Profile of Biller, Admin, Branch Admin

After the implementation and coding of system, Comprehensive testing was performed on the system to determine the loopholes and possible flaws in the system.

1. **Limitations and Future Extension**

**Limitations and Future Extension**

#### **Limitations:-**

Internet Connectivity is required

Admin is required to handle the app

Admin should have basic knowledge of handling app

Application is only for management of Bill and Stock

It is desktop base Application Hardware dependent.

#### **Future Extensions:-**

Payment Gateway Can be added

Communicate through SMS and Email

Chat facility added for communication of employee

1. **Bibliography**

**Bibliography**

Web:-

<https://stackoverflow.com>

<https://www.c-sharpcorner.com>

<https://en.wikipedia.org/wiki/Windows_Forms>

<https://docs.microsoft.com/en-us/dotnet/framework/wcf/whats-wcf>