HOME WORK

STUDY THE CASE STUDY AND MAKE THE CORRECTIONS IN THE FOLLOWING CASE STUDY

Software Requirements Specification

For

All in one e-billing

PROBLEM STATEMENT FOR "ALL IN ONE E-BILLING"

Though India is one of the biggest leaders in the IT world, the software industry has failed to succeed at all levels of its society. To become most powerful in this field, we must emphasize on use of IT for common people. Even though Indian software market is world recognizable, there is big hiatus between use of IT for people's regular use; for example, even today, people stand in queue for long hours to pay electricity bill, wasting their valuable time and at the cost of travelling. Taking such problems into account, there comes requirement of the software which will provide a single gateway for submission of different kind of weekly/monthly bills, using web service.

This document deals with such a software called "All in one e-billing". Using the application user can pay his/her various monthly bills through the system on a single click, saving time as well as physical work. The service facilitates online payment of Electricity bill, Telephone bill, Society bill, LPG gas bill, Cable bill etc through the single gateway.

Once the user completes initial registration procedure, user will get login ID and password, using which a user can pay the bill. The procedure is simple, at first, user need to choose type of bill to be submitted from drop down menu, after selecting the type of bill, now user need to provide area code. Depending upon the information provided, user will diverted to the corresponding system database; where all information will be cross verified, producing appropriate bill. Here, customer can pay the bill through credit/debit card. For the bank related transactions the details regarding credit/debit card are requested and system is transferred to the bank for payment. As soon as successful completion of transaction takes place system's database will get updated for future reference.

Apart from the above, there are two modules that extend the feature of the system further. First one is Automatic update system, that keeps updating the user about any change in the system or any possible prediction about the services provided by the application. The second module called feedback mechanism, is specially designed for user for expressing their opinions/views/suggestions about the product. This mechanism is then handled system admin, taking appropriate action. The bill history section keeps the track of all bills that has been submitted by the user till date.

The "All in one e-Billing " is a simple web based application which will provide single gateway to pay bills of all categories. The application is user friendly and secure. Through the gateway, the customer transferred to appropriate system website. The "All in one e-billing" application will generate an invoice for the same. Thus it enables users to pay bill online without visiting to the shop physically.

Table of Contents

1. Introduction

1.1 Purpose

The "All in one e-billing" is web application intended to provide easy way to pay all kind of bills using Internet as medium. It will enable customers to pay bills without visiting the shop/office physically, by using the application, on a single click. This document explains how the single interface system can be implemented efficiently so that users can easily pay the bills, any time, from anywhere.

1.2 Document Conventions

When writing this document it was inherited that all requirements have the same priority. The general flow of information has been maintained according to standard rules.

- The heading in the document are in Times New Roman with font size 16.
- Sub heading are in Times New Roman with font size 14.
- Description is written in Times New Roman font with font size 12.

1.3 Intended Audience and Reading Suggestions

The document is intended for developers, project managers, testers, and other technical staff. The rest of the SRS contains description about the product details, features and interfaces.

1.4 Product Scope

The "All in one e-billing" provides an easy to handle kind of interface to all its user. Using the application one can pay monthly bills through internet. The application's main module provides the facility of payment of monthly bills such as Electricity bills, Telephone bills, Corporation bills, Cable bill and LPG bill. Other module include feedback mechanism, automatic update policy etc.

1.5 References

The following are the references used,

- Software Engineering : A Practitioner's approach By Pressman, Roger
- IEEE paper on web and project management.
- IEEE computer society.

2. Overall Description

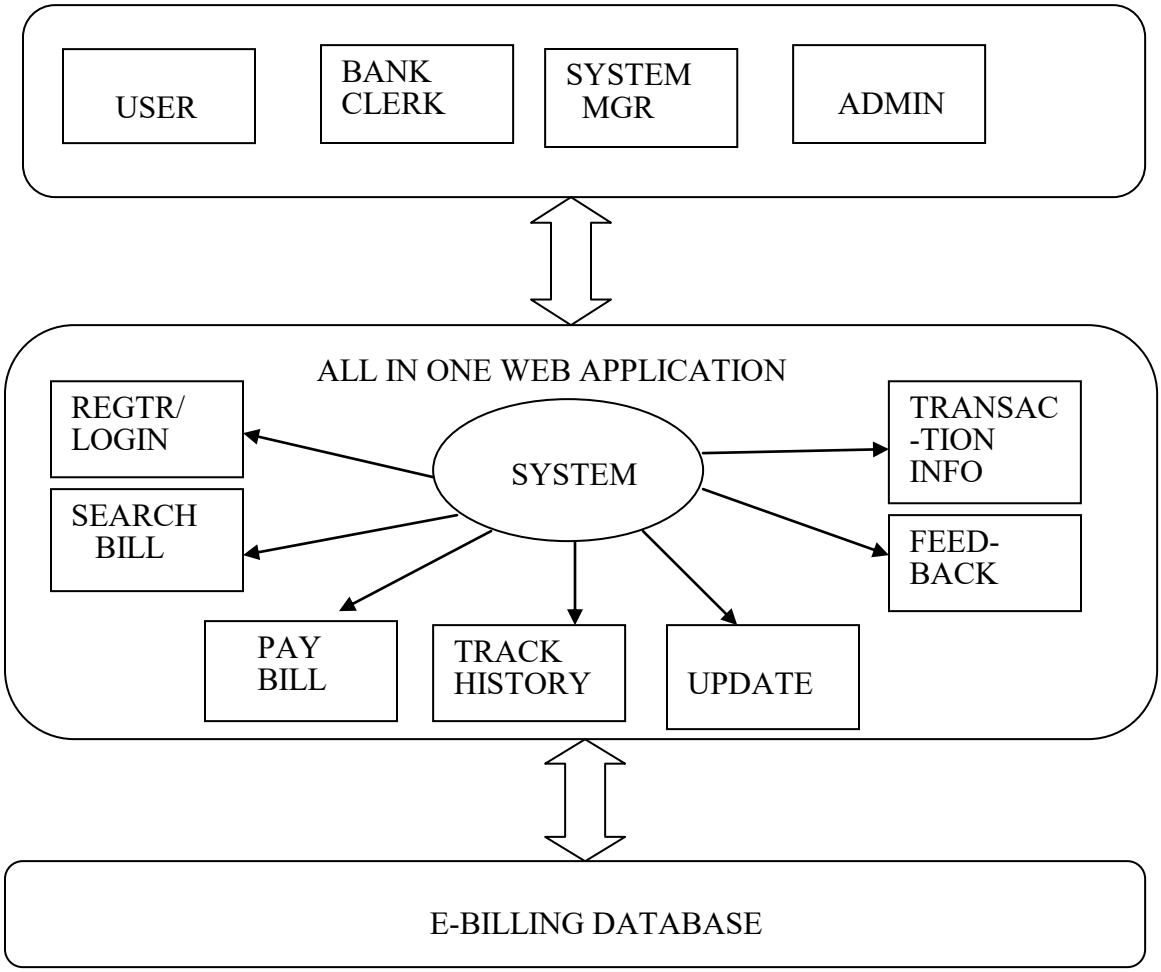
2.1 Product Perspective

The Product is made for the service provider, they give online access to the customer facilitating e-payment of monthly bills. The product is efficient one as compared to other existing products, providing extra features like automatic update and feedback mechanism. Special care has been taken to make the system faster and secure.

2.2 Product Features

The major functions provided by the applications are:

- Online Billing: User can make payments of his/her monthly bills just by one click.
- Automatic Update: This facility alert the user about future issues or any changes in the system.
- Feedback Mechanism: Through this customer can specify his/ her opinion or thinking about the system.



2.3 User Classes and characteristics

The product will use following classes:

- User: This class will contain information about the user, that using the system, a user can be registered user or unregistered user. unregistered user has provided limited access to the system's functionality. The unregistered user can search the different kind of bill information according area but cannot pay the bill. He/she can see the help menu and get more details about the functions provided by the system. However, the access is limited and Unregistered users are not allowed to access the System Update and Feedback mechanism features. On the other hand, the registered user has given the full access to all functionalities, a registered user can search the bill section, and pay the bill. She/he is allowed to see the System Update, can give feedback or can see the previous bill history. Thus, the unregistered user has provided with limited access. To register for the system, user simply has to fill the registration form, by clicking on the Register button. All in information provided by aspiring user will be checked and confirmed by the system admin; and once system admin found all information authenticated, system admin will provide Temporary Password to user, which user can change for further security.
- Admin: admin class will have information about the administrator that are managing the system. As the system is available 24 by 7, the system can have more than one administrator. Thus the class maintains the log of the administrator that are operating on the core functions of the system. This log data can be exchanged between the two or more admin in order to know the details of changes that has been done. The admin class have right to accept /reject the user registration. It keeps track of the System manager and updates that are done to system.
- System manager : This class contains the information about the system manger who looks after the admin. The system manager class is responsible for maintaining updates/response to feedback mechanism. The class maintains the track record of feedback from the user.
- BillInfo: The class contains the information about the bill payment made by the user. This information is available to user in the history section. Thus using the class the user of the system can see the previous paid/unpaid bill.

2.4 Operating Environment

The product can run without any interrupt on 32-bit MS Windows (95/98/NT/2000/XP/07), Linux/Unix. The application requires Internet connection, the browsers like IE 6.0,Opera 7.0, Mozilla Firefox 2.0 or their next versions can be used. The printer machine must be attached to the system, in order to take printout of the receipt generated.

2.5 Design and Implement Constraints

- As the application is purely for online use, the internet access is most important requirement to run the application successfully. Thus the application is dependent on Internet service provider.
- For making the payment , the system is diverted to proper database of corresponding channel, thus the application is dependant of their system to work.
- No multilingual support is provided.
- Real life credit card validation and banking system is not implemented.

2.6 User Documentation

The product contains the user manual. The manual contains the complete basic information on using the application, which will be the great help for users. Apart from the user manual, online help is also provided which expose the technical as well as basic details. Still if anyone has doubt, he/she can mail to the admin, e-mail facility is also included.

2.7 Assumptions and Dependencies

- The successful working of the application depends upon the availability of Internet.
  It is assumed that Internet connection is available all the time.
- It is assumed that the end user must have appropriate versions of the Operating system and Browser accessed.
- It is assumed that user providing all the correct information while filling up the form.

3. System Features and Functional Specifications

An enhanced atomized system is developed to maintain Customer, Product, bill , bill details data and produce Bill with following features.

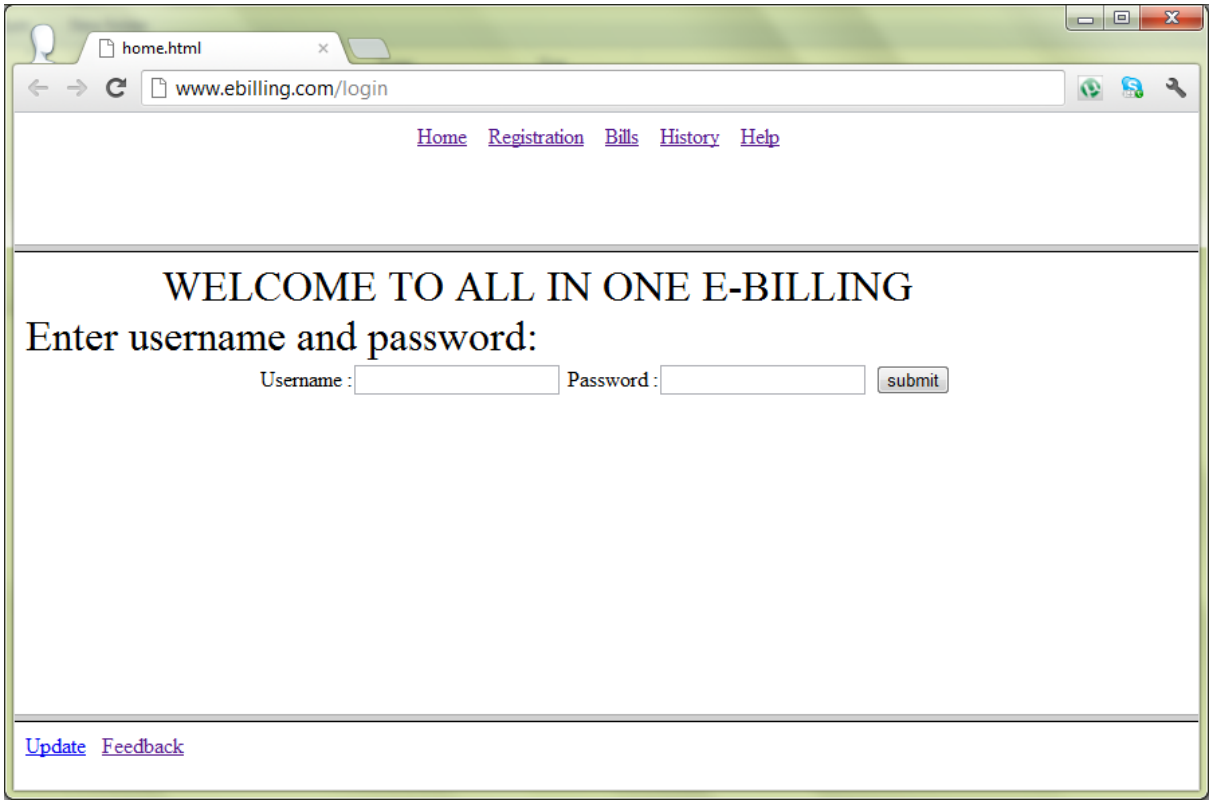
- The customer will be able to Add, Edit own account details and search the bill . The customer enter the bill type, choose the area code. The systems search engine then start searching the different offices located in the area provided by the customer. Customer then choose one from the drop down menu , and provide the date/month for which he wants to pay the bill. The system then asks for mode of payment, payment details will be verified by the bank clerk. For the bank related task the system will be forwarded to the banks database, and upon processing the request the control comes back to the system.
- Improved search capabilities of bill information will be available; It will display all the details of the particular product with current stock quantity.
- By using their own account, system manager will be able to place orders through the system. In addition, system manager will be able to apply discounts in accord with the current sales policy. Prices and order information will be expressible in international currency and date/time formats.
- After paying the bill system automatically print the bill submission receipt with all the details.
- Searching of previously paid bills is also available; Customer can search the previously saved /paid bills by entering system generated unique bill number.
- Apart from these features, the system has two sub-module; first on is "Automatic update", this facility inform about any updates regarding the system that have done or that might be possible. It also inform about the deadlines for filling the bills, when deadline comes very close. The other sub-system is "Feedback mechanism ", which enable user of the system to submit their suggestions/complaints/opinion about the system. Thus using the feature, system can attain further level of perfection
- System maintains the user master and security features.
- Following table describes various functional specification of the system.

Unregistered User	Search any information about bills Register Read the Updates, Help Give Feedback
Registered User	Check Updates Give Feedback Search the information about bills Search the bill section Fill the form details Pay the amount Print the Form Subscribe the Weekly Updates Submit the Query
System Manager	Provide with the Earlier Mistakes Check the filled form Check and handle the feedback Answer the query
Bank Clerk	Enter the Transaction Details Checks the transactions Enter paid Details Update user account information
Administrator	Add or Remove User/Manager Register Users Manage Database Update Information Mail the answers to the Users mail id Allow the Access to the Bank

4.External Interface Requirement

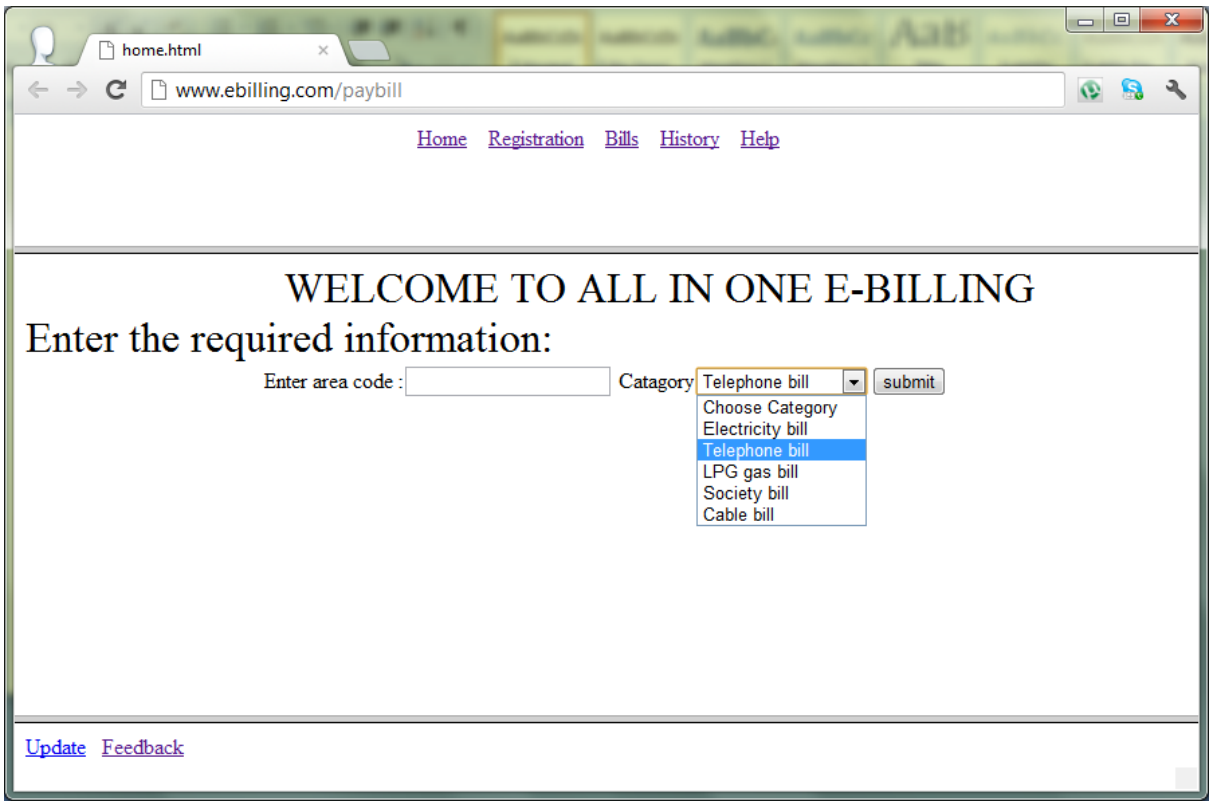
4.1 User Interface

- The user interface shall be implemented using any tool or software package like Java Applet, MS Front Page, EJB etc.
- Below the screen shows the home page, it is the first screen user can see. Through this interface user can login, can do their activity.



- Following is the registration form, to create the new account. User has to fill all the fields correctly and then submit the form.
- Only after all information provide by user is verified and found true; admin provide temporary password to the user, which user can change for futher security purpose.

- Following screen shows the different bill user can choose fom the drag down menu, and according to the area, the request is diverted.



4.2 Hardware Interface

As the product fully depends upon the internet connection, all the corresponding hardware is required to run the application, such as

- Modem
- LAN/ WAN connection
- Ethernet cross -cable

4.3 Software Interface

- The e-Billing system will communicate to the bank database for verifying all the transactions related to money transfer and account details.
- The system database will have all the information of the registered customer, Thus the system database acts as the software interface.
- The application is built using JAVA as a programming language and Oracle 11g is used for storage purpose.

4.4 Communication Interface

The "All in one e-billing " system shall use the HTTP protocol for communication over the Internet and the intranet communication will be trough TCP/IP protocol suite.

5. Other Nonfunctional Requirements

5.1 Performance Requirement

- The product depends on Web server to run successfully. Access to the Web server must be provided as per the requirements.
- The application will take initial load time depending upon the Internet connection bandwidth.
- The performance will also depend on the hardware and software component available at the user end.

5.2 Safety Requirement

All system data must be backed up every 24 hours and the backup copies stored in another server at different building or location for disaster recovery.

5.3 Security Requirement

Following are the security measures provided :

- The system shall use secure sockets in all transactions that include any confidential information.
- The system shall automatically log out all customers after a period of activity.
- The system shall confirm all transactions with the customer's web browser.
- The system shall not leave any cookies on the customer's computer containing the user's password.
- The system shall not leave any cookies on the customer's computer containing any of the user's confidential information.

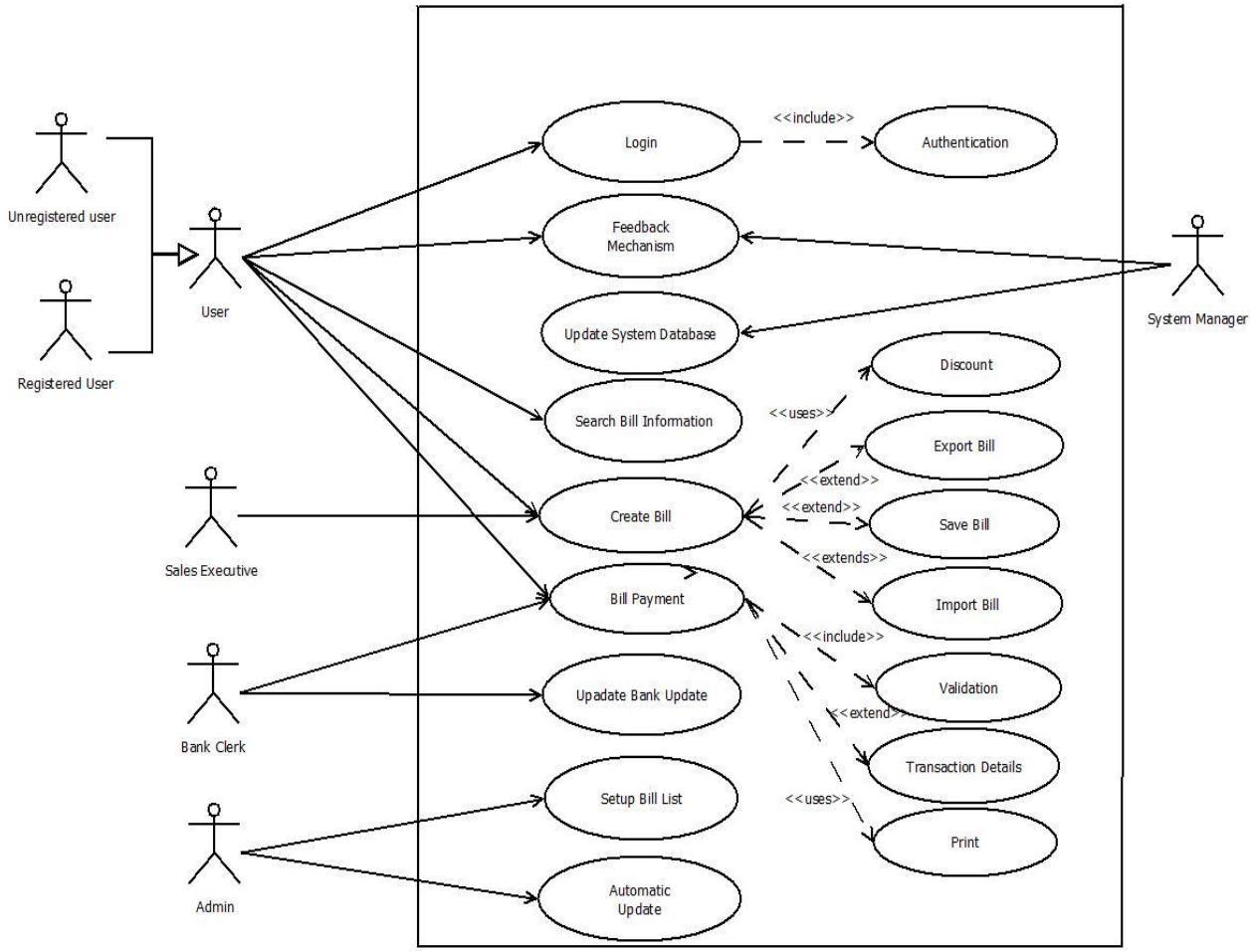
- The customer's web browser shall never display a customer's password. It shall always be echoed with special characters representing typed character.
- The system's back-end server shall only be accessible to authenticated administrators.
- The system's back-end databases shall be encrypted.

5.4 Software Quality Attributes

The quality of the software is maintained in such a way that , the system is so user friendly and efficient. Other qualities that includes :

- Availability : The system is available for use 24 by 7, thus having no timing restriction for accessing the application. Thus providing flexibility for using the system.
- Robustness : If the connection between the user and the system is broken prior to bill paid being either confirmed or canceled, the e-billing system shall enable the user to recover from an incomplete transaction.
- Consistency : The "All in one e-billing" system automatically updates all the transactions data performed by user in database, so the most recent data is responded to query.

1. USE CASE DIAGRAM



Description of Use Case Diagram :

Various Actors in the system includes

1.User : User of the system can be registered or unregistered. Registered user has been provided with full set of functionality however unregistered user has given limited access to the system.

2.Sales Executive: The actor creates the bill submitted by user.

3. Bank clerk: Deals with the bill payment, checks transaction details, and updates the bank database.

4.Admin: This actor setup the bill lists according to different categories and month. Also it deals with Automatic update mechanism, which notifies the customer about the deadlines of the bill payment.

5.System manager : The actor updates the system data such as billing information , registration and user information. It also deals with the feedback mechanism.

The use cases used in the use case diagram are as below :

1.Login

Overview:

To acces the billing functionality user need to login using the username and password. If the user is not registered one the he/she has to first register to the system, after registration user will have username and password.

Precondition:

User must be the registered member of the system.

Main flow of event:

1.Provide user name.

2.Provide password.

3.If user is authenticated, acess is provided or redirected to the registration page.

Postcondition:

None.

2.Feedback mechanism

Overview:

The use case provide the facility of providing feedback to the system. The user can suggest/complain/quetions to the system regarding the functions provided by the billing system.

Precondition:

User must be the registered member of the system.

Main flow of event:

1.Login to the system.

2.Provide feedback.

3.Feedback is then handled by the System manager.

4.Reply from System manager to user regarding feedback.

Postcondition:

Feedback submission ID generated for future reference.

3. Update System Database

Overview:

Whenever the user pays the bill or update information about himself/herself the usecase is used to update the information in the system. The system manager will update the information in the system.

Precondition:

The system manager must be autherized one to update the changes made by user.

Main flow of event:

1.User pays the bill/modify self data.

2.System manager checks the update, and if necessary make changes.

Postcondition:

If updates are verified then only changes are made.

4.Search bill information

Overview:

This use case helps in finding out the information about the bills, Intially user need to select type of bills from the drop down menu and provide location ID, then according to the information provided the bill will be searched.

Precondition:

1.User must be the authenticated/registered user.

2.It must provide valide data about the bill i.e. bill type and area /location code.

Main flow of events:

1.User selects the category of bill to be paid and Location ID.

2.The appropriate bill according to the information provided is thn displayed.

3.User then can choose pay bill or pay bill later option.

Postcondition:

Bill generated and diasplayed has a unique bill ID which can be used for future reference.

5. Create bill

Overview

The sales executive generate the bill according to the information provided by the user. It verifies the Bill type and location ID. According to the month, the bill is

created and displayed. The past unpaid bills are also taken into account and are also displayed at the same time.

Precondition:

1.Choosing valid type of bill.

2.Providing valid Location ID.

Main flow of events:

1.After getting the bill type and Location ID/ area code, bill is created.

2.The System displays the created bill.

3.System asks to the user for bill payment and redirected to the next module depending upon the response from the user.

Postcondition:

None.

6. Bill payment

Overview :

Bill payment use case deals with validation part of the payment. It checks the bank details provided by the user.The usecaseis handled by the bank clerk.

Precondition :

User must provide valid credit/debit card details.

Main flow of events

1.User choose the bill to pay.

2.User enters the payment details.

3.Details are then checked by the bank clerk for validation.

Postcondition :

Amount must be transfered to the proper channel.

7. Update bank database

Overview:

This usecase is also controlled by the Bank clerk, and it updates the bank database according the transaction.

Precondition:

Valid transaction by user.

Main flow of events:

1.Bank clerk checks for the transaction details provided by user.

2.If the payment details are correct, then transaction is successful and system is updated othewise the transaction failure message is displayed.

Postcondition:

Valid transaction must be updated.

8. Set up bill list

Overview:

The usecase deals with listing up the different categores of bill accoding to the area and applicaton. It is updated/modified every time by the Admin.

Precondition:

New bill after some specific period when arrives, it must be updated.

Main flow of events :

1.After a period of month/week the bill list is updated and displayed.

2.The user then choose the bill from the drop down menu and proceed to next level.

Postcondition:

The updation must be made according to the proper channel.

9.Print :

Overview:

To take the printout of the paid bill.

Pre Condition :

User must have duly filled and paid the form.

Main flow Events:

1.Fill the form required for billing.

2.Submit the form

3.Click on save form button

4.Get the bill verified

5.Click on print button

Post Condition :

User will be provided with the printout of the bill paid.

10. Save bill

Overview: To save the paid bill.

Pre Condition : User must have paid the bill.

Main flow Events:

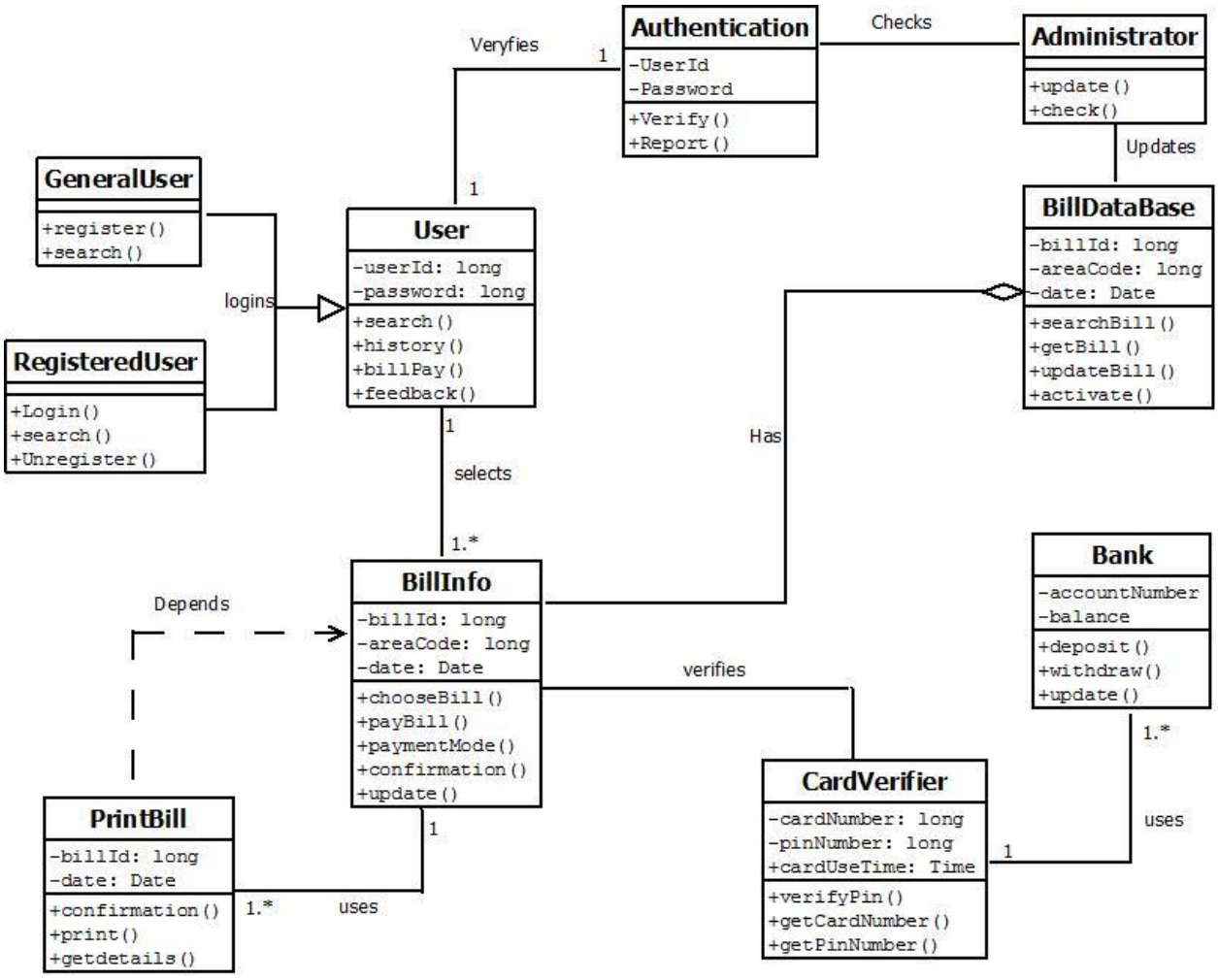
1.Fill the form required for billing and submit it.

3.Provide the account details for payment

4. If successful payment, Bill will be saved, generating unique bill number which can be used for future reference.

Post Condition : Bill will be saved.

2. CLASS DIAGRAM



Class Diagram

NOTE :ADD USER HAS BANK ACCOUNT AND CARD

Description of class diagram :

1. Identify the classes

- General user

- Registered user
- User
- Authentication
- Administrator
- BillDataBase
- Billinfo
- PrintBill
- CardVerifier
- Bank

2. Identify association Between the Object

- User can be Unregistered or Registered; registered user login to the system using loginID and Password in the User class.
- The login Id and Password Provided by the user; is then checked by Authentication class.
- BillInfo class maintains the current record of the bill which the user interested in paying. The class is associated with other many classes for payment.
- PrintBill is the class which is dependant of the BillInfo class. Once the user pays the bill the PrintBill class prints the payment reciept.
- CardVerifier Class verifies the Credit/debit card details provided by the user.
- Bank class has account details of the customer. When customer enters the payment details, CardVerifier class checks the details with the help of Bank class. If Payment is successful then the Bank class updates the corresponding information in the account of user.

- BillDataBase class has the list of all bills. Once the bill Id and area code is provided by the user; the bill is displayed.

3. Identify Attributes of Objects

- class : User

  Attribute: 1.UserID

  2.Password

- class : Authentication

  Attribute: 1.UserID

  2.Password

- class :Administrator

  Attribute: 1.Admin ID

- class :BillDataBase

  Attribute: 1.BillID

  2.AreaCode

  3.Date

- class :BillInfo

  Attribute: 1.BillID

  2.AreaCode

  3.Date

- class :PrintBill

  Attribute: 1.BillID

2.Date

- class :CardVerifier

  Attribute: 1.Card number

  2.Pin number

  3.Card use time

- Class: Bank

  Attribute: 1.Account number

  2.Balance

4.Identify operations of objects

- class : User

  Operation: 1.Search()

  2.history()

  3.billpay()

  4.feedback()

- class : Authentication

  Operation: 1.Verify()

  2.Report()

- class :Administrator

  Operation: 1.Update()

  2.Check()

- class :BillDataBase

Operation: 1.Searchbill()

2.GetBill()

3.UpdateBill()

4.Activate()

- class :BillInfo

  Operation: 1.ChooseBill()

  2.PayBill()

  3.PaymentMode()

  4.Confirmation()

  5.Update()

- class :PrintBill

  Operation: 1.Confirmation()

  2.Print()

  3.GetDetails()

  4.VerifyPin()

- class :CardVerifier

  Operation: 1.GetCardNumber()

  2.GetPinNumber()

  3.VerifyPin()

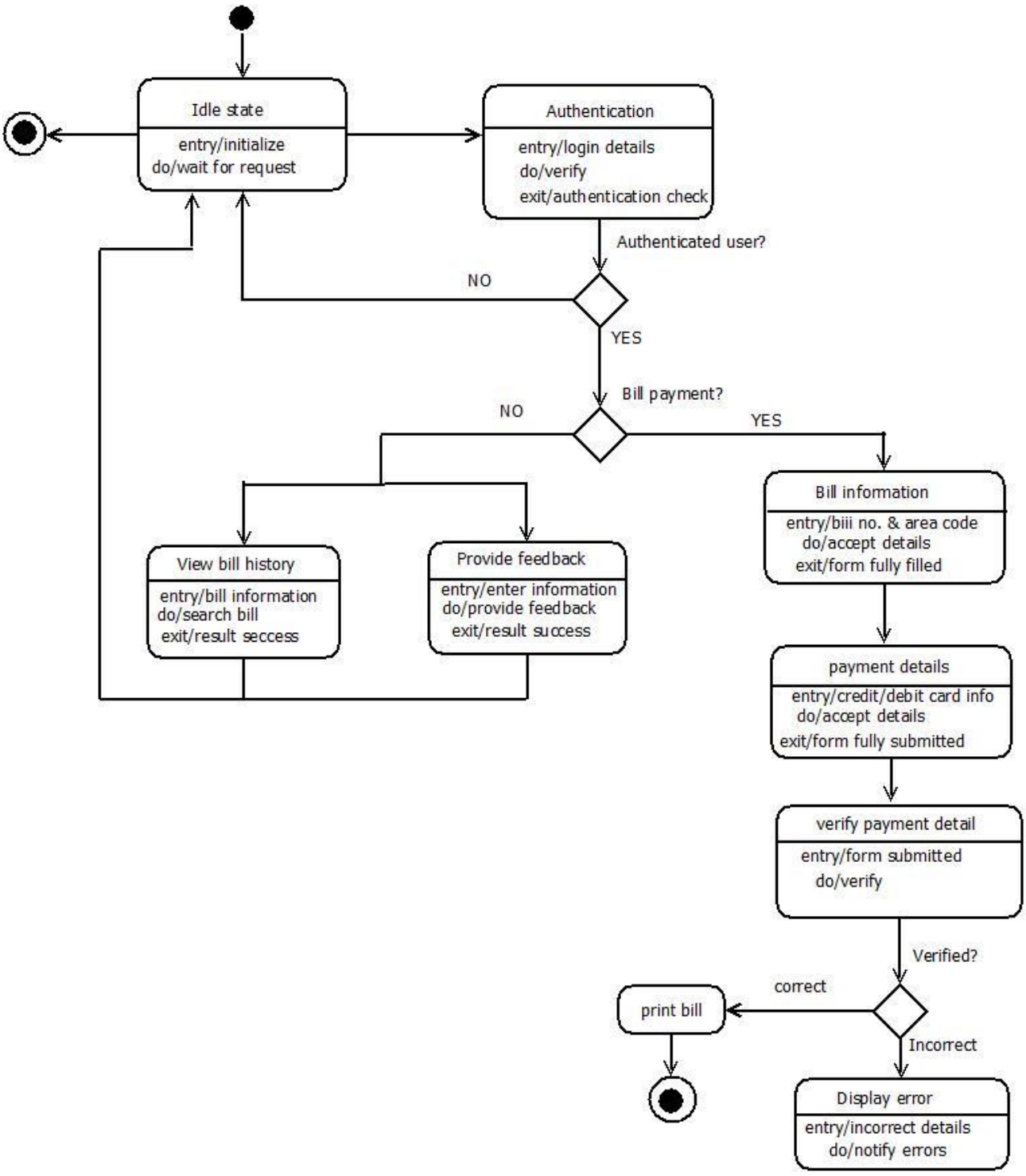
- Class : Bank

  Operation: 1.Deposite()

  2.Withdraw()

3.Update()

3. STATE DIAGRAM



Description of State Chart Diagram :

1. **State: Authentication**

The state verifies the authenticity of user. User provide username and Password, which is then checked/

- **Entry Action:** Login details are provided by the user.
- **Do Action:** Verifies the login and password.

2. **State: View Bill History**

Using the state the user can see the past paid bills. The bills can be searched by choosing from the dropdown menu or by simply providing unique bill ID.

- **Entry Action:** Bill information .
- **Do Action:** Search and displays the bill.

3. **State: Provide Feedback**

Using the state user can provide feedback about the system. A feedback can be complaint, suggestion or inquiry.

- **Entry Action:** Feedback submission.
- **Do Action:** Generate the Feedback submission number for future reference.

4. **State: Bill Information**

This state deals with displaying the bills; user enter the bill id and area code. According to the information provided the bill is displayed.

- **Entry Action:** Bill id and area code..
- **Do Action:** Display the corresponding bill.

5. **State: Payment Details**

In the State the user enters the debit/credit card details for the payment.

- **Entry Action:** payment details from the user
- **Do Action:** Accept the details for verification

6. State: Verify Payment Details

In this state the system checks the payment details and verifies it.

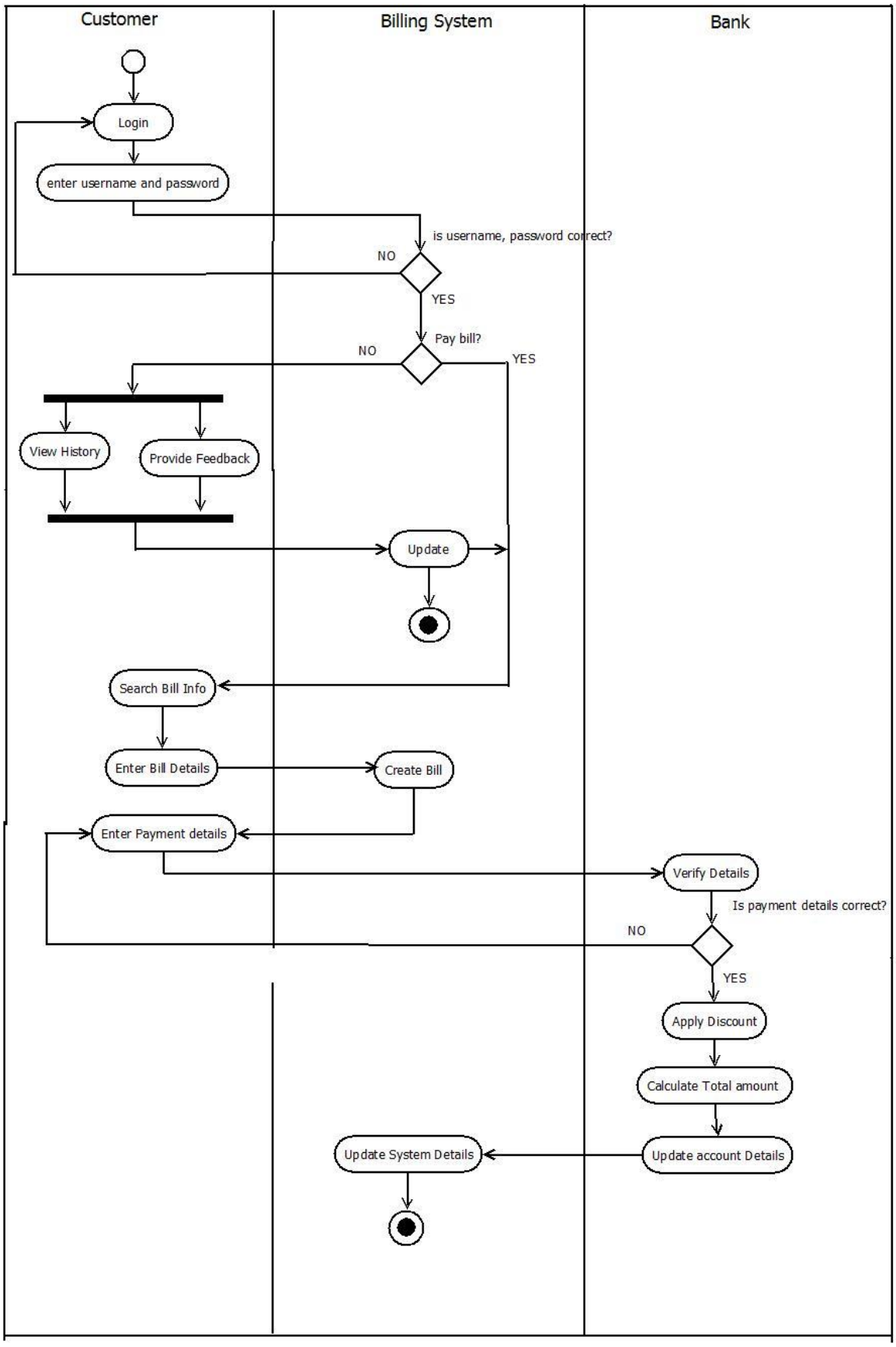
- **Entry Action:** payment form and details
- **Do Action:** Verifies the details

7. State : Display Error

If the entered information about the credit/debit card are correct then the system displays the payment successful massage otherwise the error.

- **Entry Action:** Incorrect payment details
- **Do Action:** Display error massage.

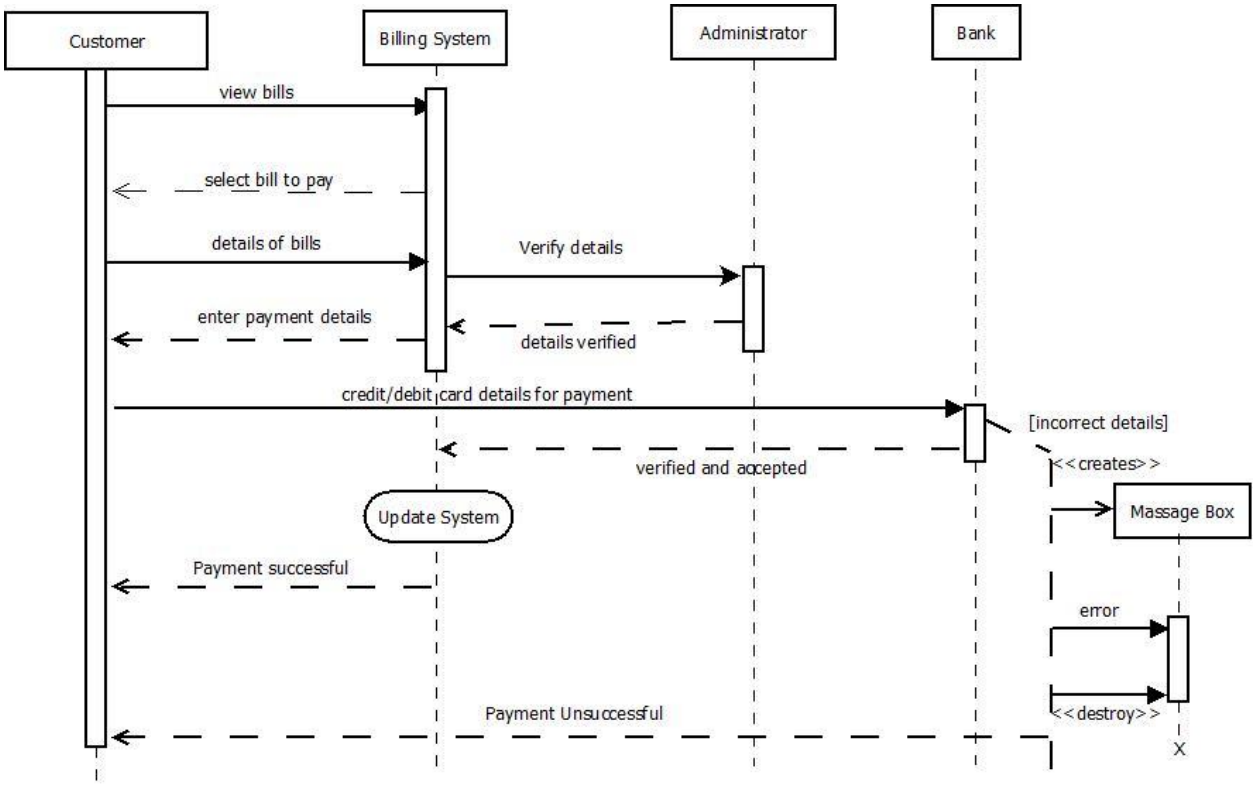
4. ACTIVITY DIAGRAM


Customer
Billing System
Bank
Login
enter username and password
is username, password correct?
NO
YES
Pay bill?
NO
YES
View History
Provide Feedback
Update
Search Bill Info
Enter Bill Details
Create Bill
Enter Payment details
Verify Details
Is payment details correct?
NO
YES
Apply Discount
Calculate Total amount
Update account Details
Update System Details

Description of Activity Diagram:

- The activity diagram to explain the all in one e-billing system is shown above and it is divided into three swimlanes as customer, billing system and bank.
- The activities in the diagram are arranged in vertical zones under a particular object i.e. under particular swimlane.
- The activity starts with customer entering the login details, this details are verified by the billing system.
- If user name and password provided by the user is correct then the user is authenticated and moves to next level otherwise the user is redirected to the login activity.
- If user do not want to pay the bill then the activity like feedback mechanism or view history comes into the action. After completing the activity user can either end or choose to pay activity.
- If the user wants to pay the bill then, it first searches the bill, enters the bill details the corresponding bill is generated by the billing system followed payment details from the user.
- The bank object verifies the payment details provided by the user, if the details are correct the discount will be added and final bill payment is done. Otherwise user is directed to enter the payment details activity.
- The billing system updates the data regarding the transaction and billing.

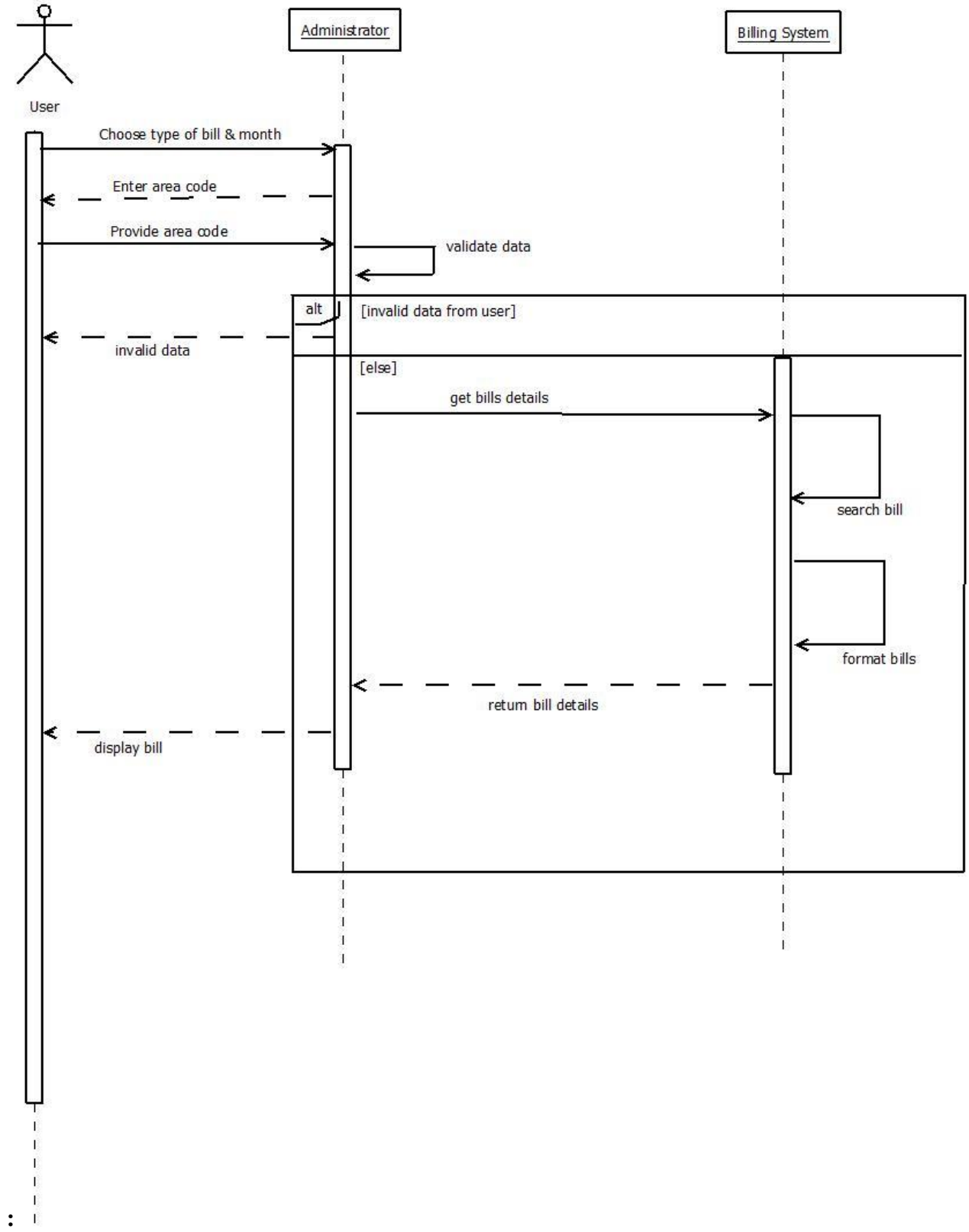
5. SEQUENCE DIAGRAM :



Description of Sequence diagram :

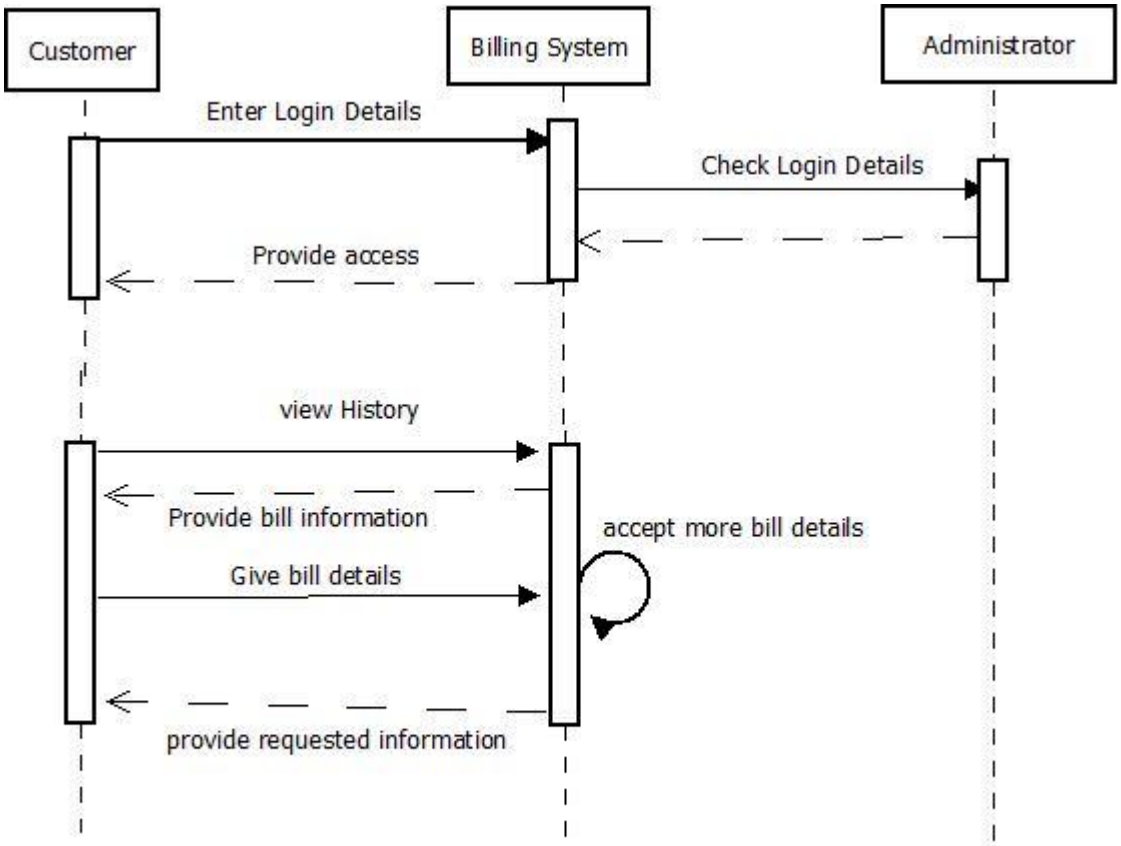
- A **sequence diagram** in a interaction diagram that shows how processes operate with one another and in what order.
- A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario.
- Sequence diagrams typically are associated with use case realizations in the Logical View of the system under development.
- The sequence diagram to explain the overall system is shown above, mainly focusing on the payment for the bill.
- The request is shown in synchronous arrow while the reply in asynchronous arrow.
- Customer views the bill and provide the details of the bill.
- The administrator object checks and verifies the bill details and ask for the payment details to user object.
- The entered details regarding the payment is the verified by the bank object.
- If the payment details are correct then the Bank object updates the bank as well as the system database and pass the payment successful massage to the user object.
- If the payment details are incorrect then the bank object creates the object displaying the payment unsuccessful massage to the user.

SEQUENCE DIAGRAM



:

SEQUENCE DIAGRAM :

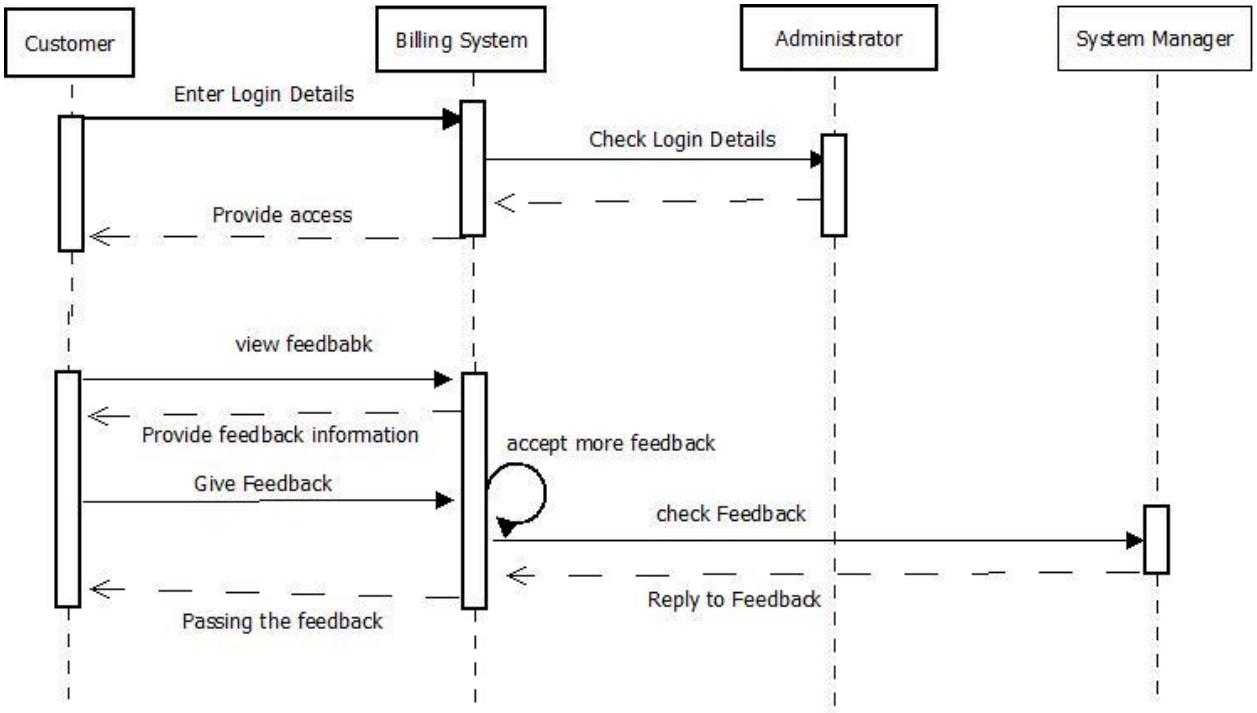


Sequence diagram : Billing History

Description of Sequence diagram :

- The sequence diagram explain the interaction between the object when a user want to see the bill history.
- By entering the bill number user can see the previous activities related to bill, such as whether bill is paid or unpaid, or to just confirm the payment.
- If the bill is unpaid, the here user came to know that can pay it through the channel.

SEQUENCE DIAGRAM :

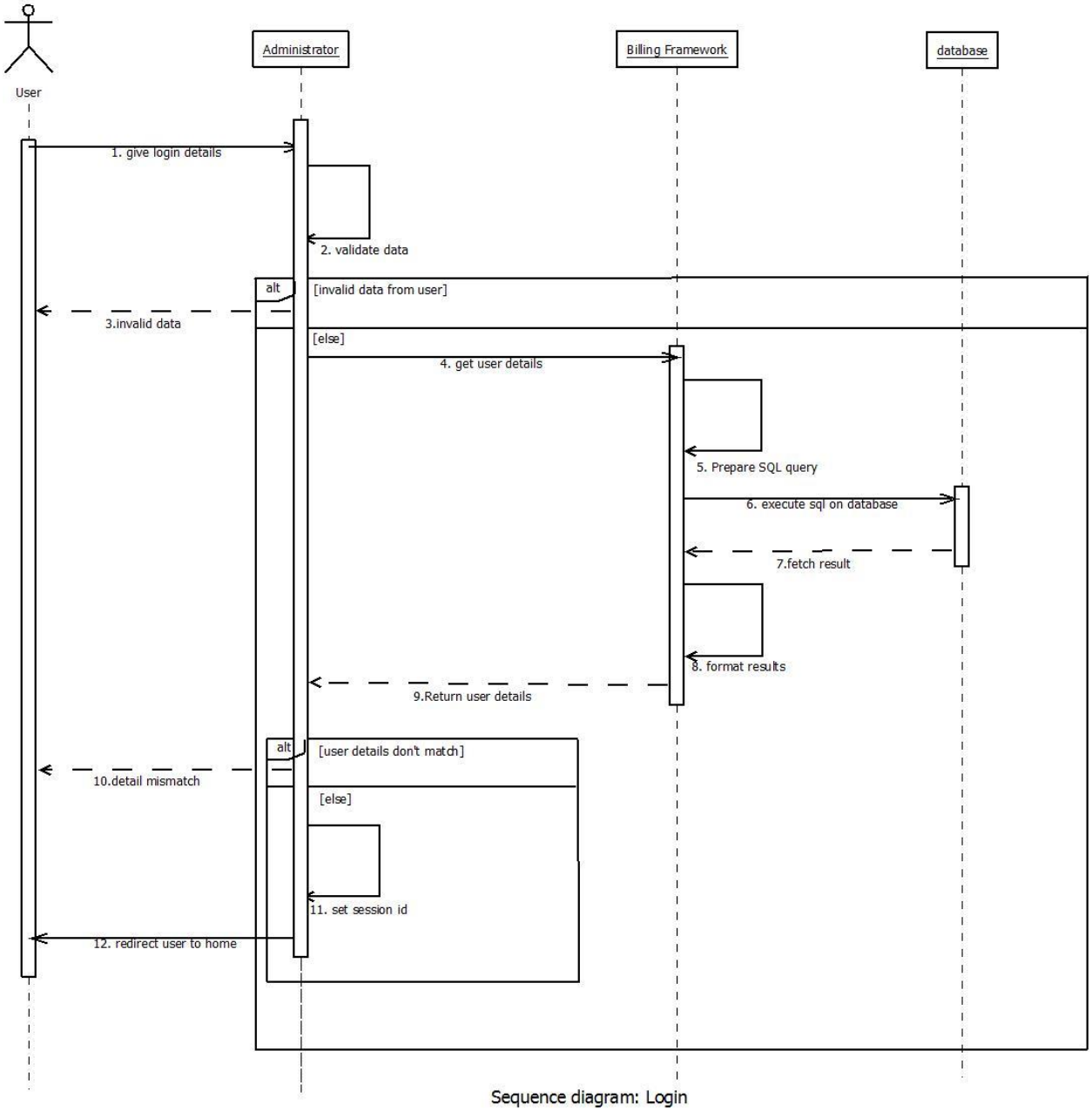


Sequence Diagram : Feedback Mechanism

Description of Sequence diagram :

- The feedback mechanism offers the facility of feedback from the user. the feedback can be query, complaint or suggestions.
- The system manager handles the feedback provided by the user. The detailed description is shown in the diagram.
- User submit the feedback to the system, system manager checks it and reply to the user/customer.

SEQUENCE DIAGRAM :



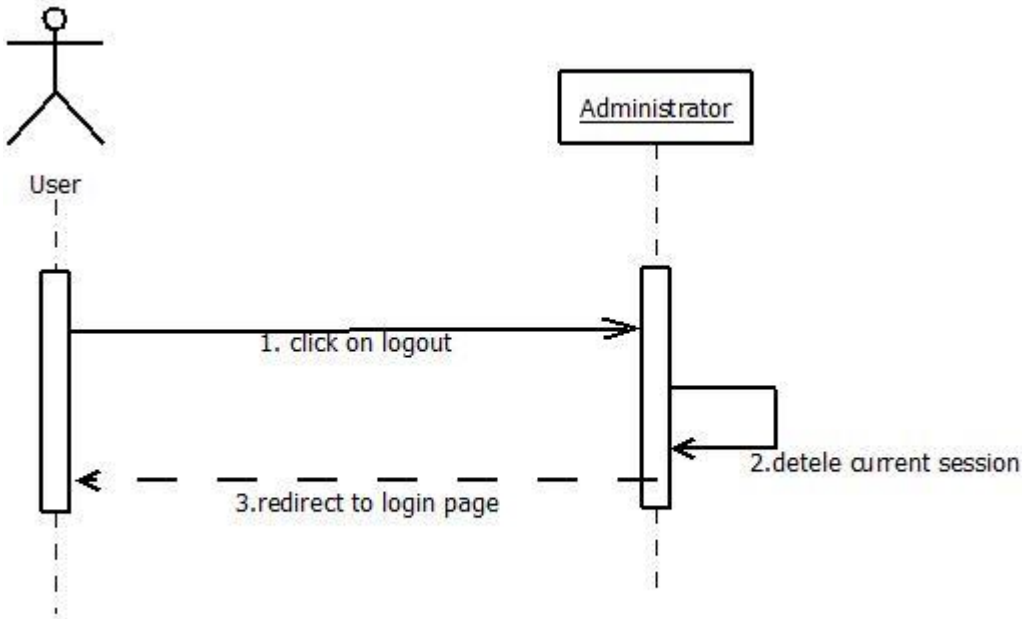
Sequence diagram: Login

Description of Sequence diagram :

- The above diagram shows the simple login process by the user.
- The login details are checked by the administrator and are verified.

- if the user is authenticated by the system then the login is successful and session is created.

SEQUENCE DIAGRAM :

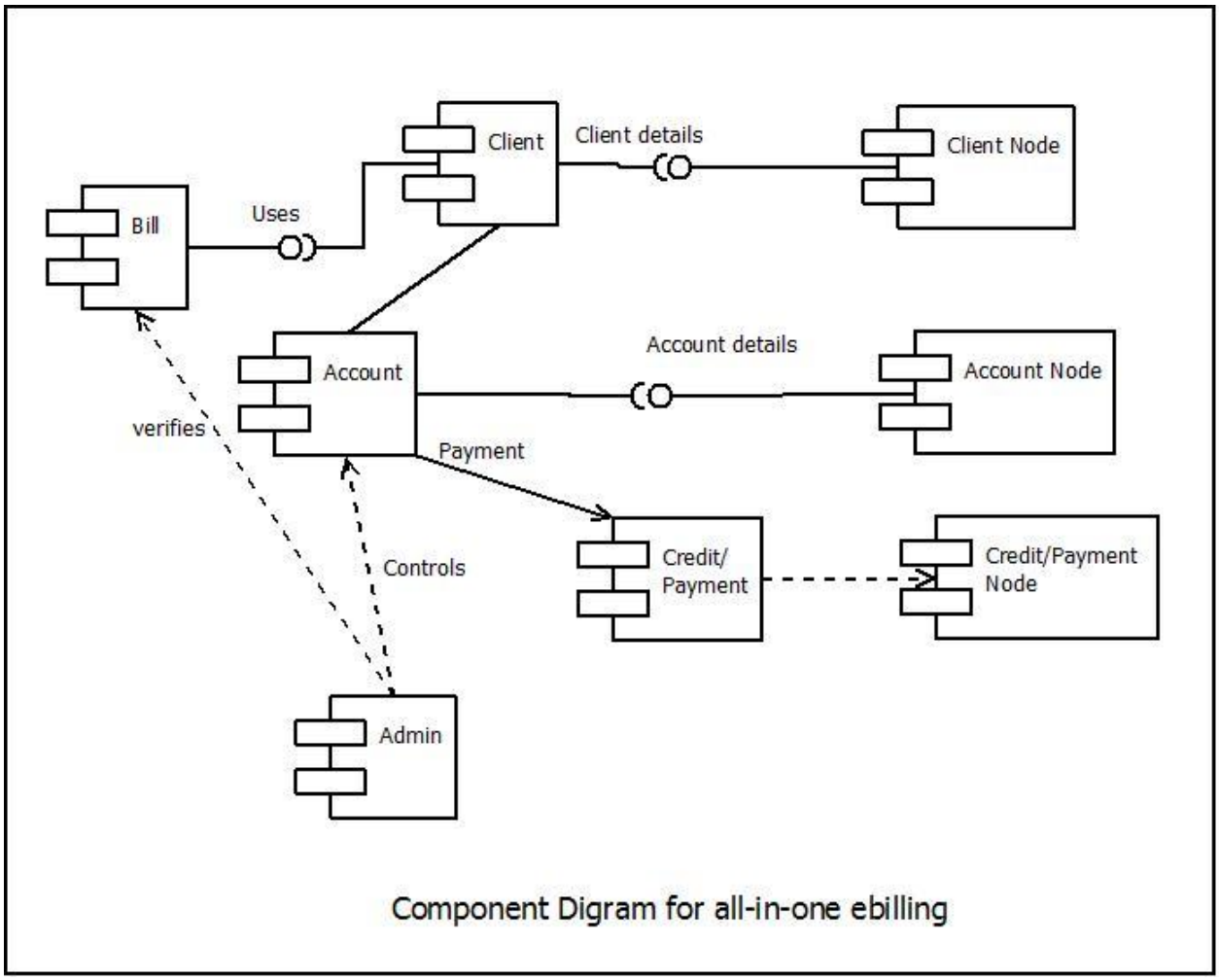


Sequence diagram: Logout

Description of Sequence diagram :

- The above sequence diagram shows the simple logout process.
- whenever user wants to logout to the system, he/she choose logout option and the session is deleted.

6.COMPONENT DIAGARM



Description of Component diagram :

- **Component diagram** shows various components and dependencies between them. This type of diagrams is used in Component-Based Development (CBD) to describe systems with Service-Oriented Architecture(SOA).

- Component-based development is based on assumptions that previously constructed components could be reused and that components could be replaced by some other "equivalent" or "conformant" components, if needed.

- The artifacts that implement component are intended to be capable of being deployed and re-deployed independently, for instance to update an existing system.

- Components in UML could represent logical components (e.g., business components, process components), physical components (e.g., CORBA components, EJB components, COM+ and .NET components, WSDL components, etc.),along with the artifacts that implement them and the nodes on which they are deployed and executed.

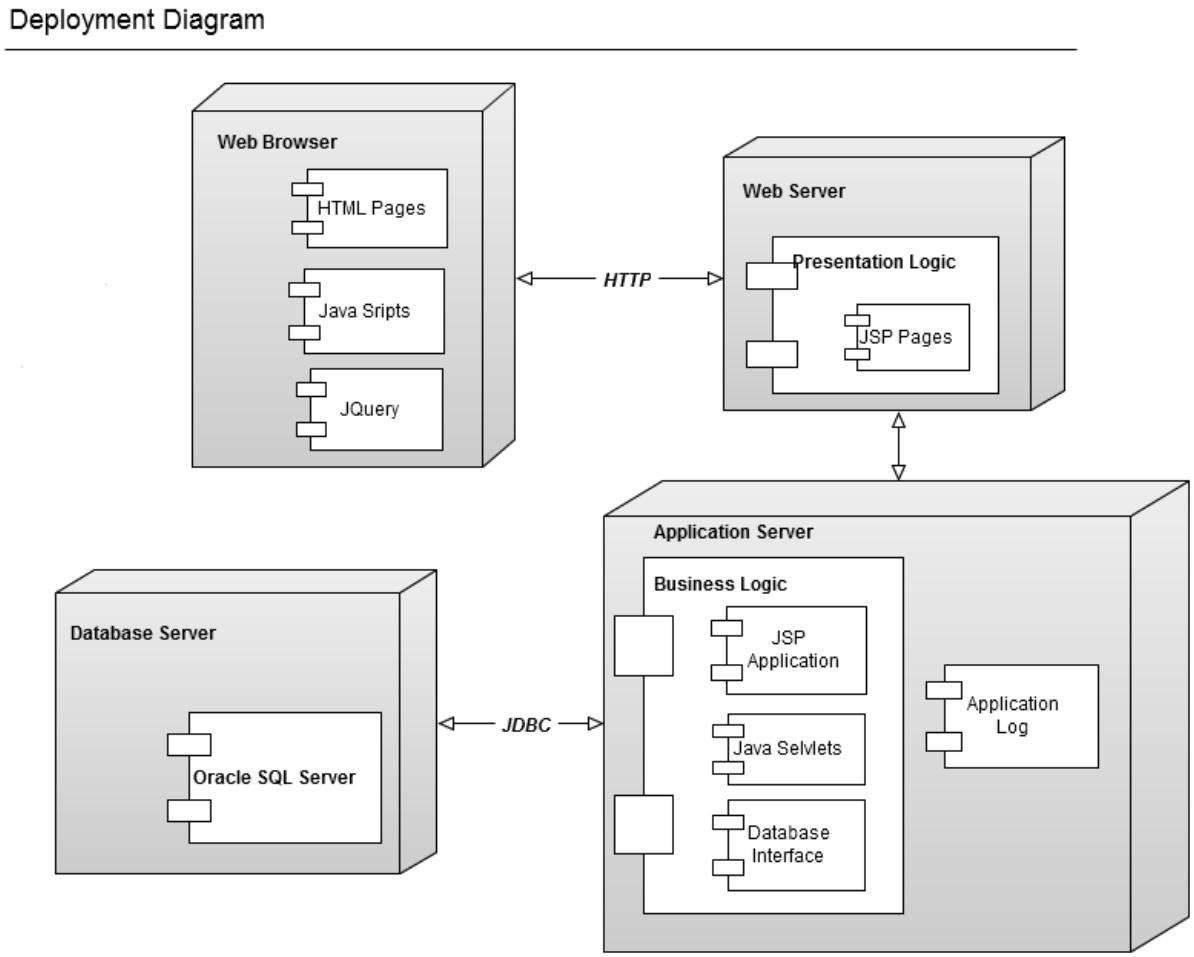
- component diagram to explain all in one billing system is given above in which different components of the system are shown.

- The three different nodes are client/customer, account node and payment/credit node.

- Node uses the different components like client, account, Admin, Bill, Payment.

- Admin controls and verifies the bill payment services for the customer.

7.DEPLOYMENT DIAGRAM



Description of Deployment diagram :

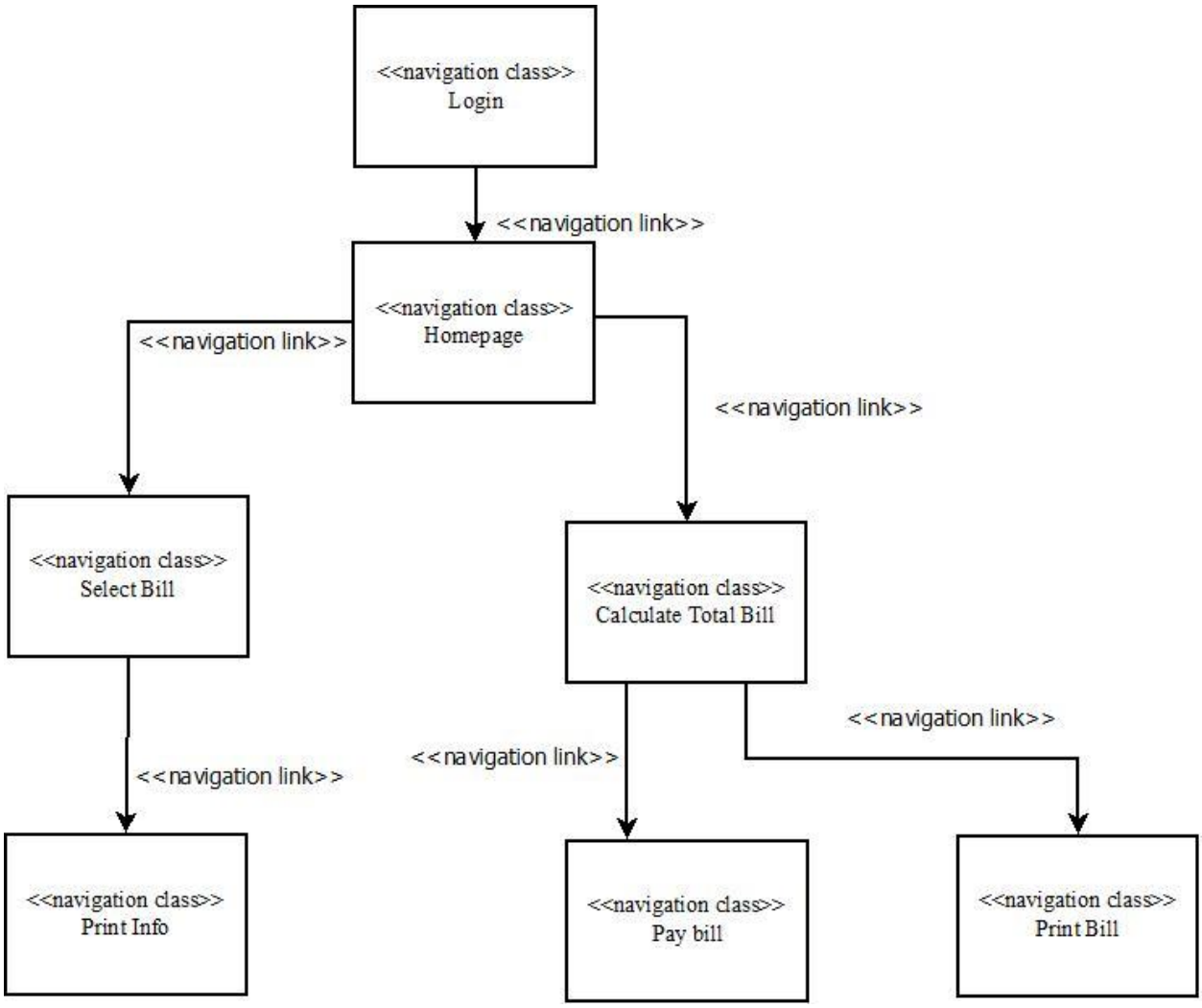
- A deployment diagram in the Unified Modeling Language models the physical deployment of artifacts on nodes.
- Database server node has the Oracle, Application server has Java, JSP, and Interface to connect to the database.
- The Web browser has JSP pages for presentation. The HTML pages for the front view.

TOWARDS OPEN APPLICATIONS

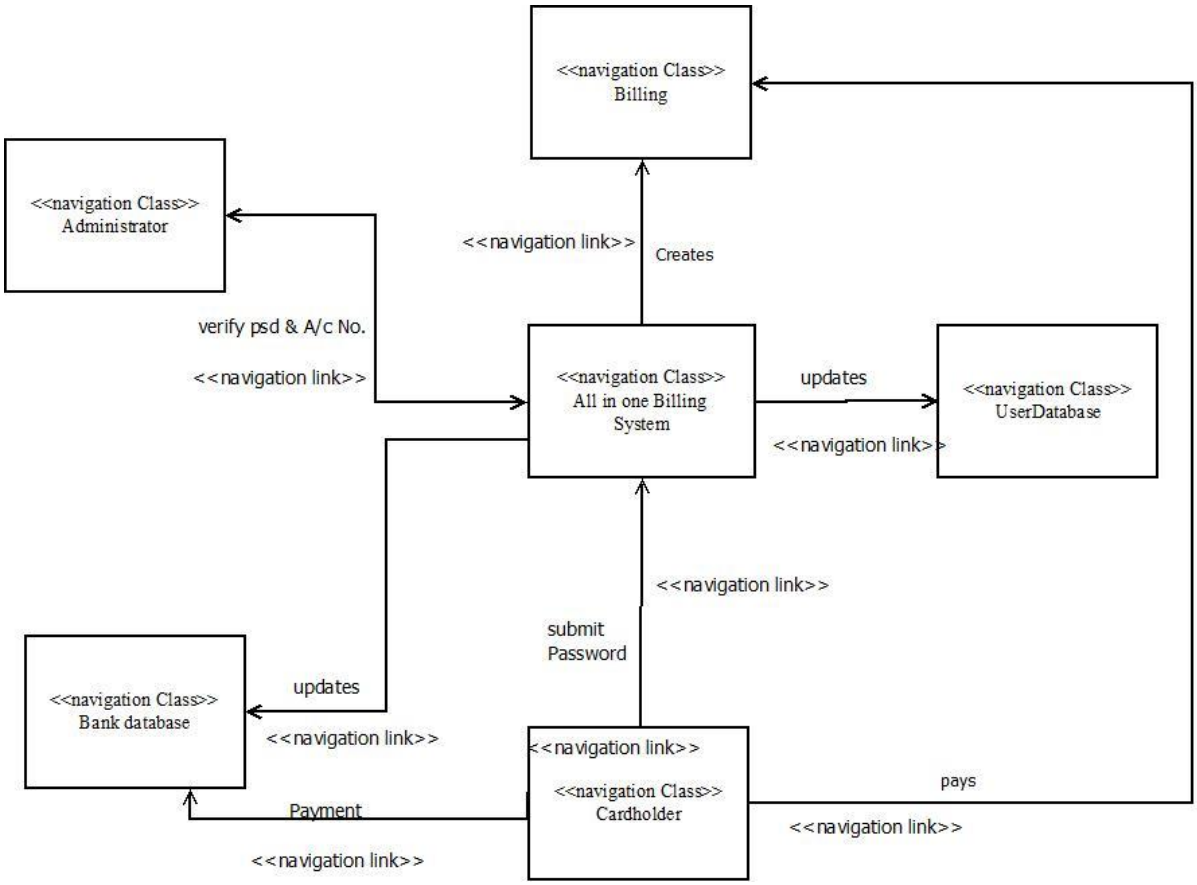
WEB ENGINEERING

ALL IN ONE E-BILLING

8. HYPERTEXT MODEL



HYPERTEXT MODEL



Software Secure Life cycle

Phases	Threats	
Analysis	• Business goals e.g. allowing 24hrs payment of fees via web can yield DOS • Privileges can be given to Manger, Account Manager and organiser account only so that anyone else can not misuse those privileges**.**	
Design	Input Validation	During designing of online form do proper validation of name, email, suggestions section, exhibition section, file upload etc. Otherwise this section can be used by attacker for input validation.
	Authentication	Only authorised user should get access to website otherwise hacker can perform any of the following activity Network eavesdropping, brute force attacks, dictionary attacks, cookies reply, and credential theft.
	Authorization	Identify all roles (admin, account manager, manager) properly then give privileges according to the roles.
	Sensitive mgmt	Access to sensitive data like payment details, papers should be given to manager and account manager only; otherwise there is possibility of network eavesdropping, data tampering.
	Session mgmt	Once the user login till that user is going to logout his session should get maintain properly.
	Cryptography	For sorting password in database we need encrypted password so keys for encryption and decryption should be strong so that password will get revel.
	Parameter manipulation	Parameters passed by clients like login name, password, email should be manipulated properly by removing suspicious special characters.

	Exception mgmt	Error should get handled properly, it should not reveal any program information
Implementation	Input Validation	Validate all text boxes, remove special characters like < , >, from otherwise SQLIA,XSS and BOF is possible
	Authentication	Instead of using only login name and password use 3D password otherwise hacker can get access to your database.
	Authorization	Identify all roles (admin, account manager, manager) properly then give privileges according to the roles. Make proper use of grant command.
	Sensitive mgmt	Access to sensitive data like payment details, papers should be given to account manager and organisers only , otherwise there is possibility of network eavesdropping , data tampering
	Session mgmt	Once the user login till that user is going to logout his session should get maintain properly using Http Session Objects or any other session tracking technique. Otherwise attacker hijack any session.
	Cryptography	For sorting password in database we need encrypted password so keys for encryption and decryption should be strong so that password will get revel. Use proper DES or AES systems.
	Parameter manipulation	Parameters passed by clients like login name, password, email should be manipulated properly by removing suspicious special characters and spaces should get replace with hexadecimal symbols.
	Exception mgmt	Error should get handled properly, it should not reveal any program information
Testing	Improper test data: Most published literatures introduce techniques for generating test cases from UML models, such as sequence diagrams or activity diagrams, and so on. Any diagram-based test method is based on path traversing. A run driven by one test case may not detect the modelled threats, so various runs taking different paths may be necessary to find a path which can activate the threat behaviour. Killing criteria should get defined properly. Design test data such a way that it will cover paths.	

Deployment	**Network Threats**: All network guards like firewall, application firewall, honey-pot and IDS should be updated otherwise following threats are present Information gathering, Sniffing or eavesdropping , spoofing, Session hijacking, Denial of service **Server Threats** : Server on which you're going to deploy "E-Bombay Museum Mgmt System" should be secure otherwise following threats are possible Viruses , Trojan horse and worms Foot printing Password cracking Denial of service Arbitrary code execution Unauthorized access
Maintenance	All tables used in "All in one e-billing" should be updated properly.

The start……