K.S.R COLLEGE OF ENGINEERING

(AUTONOMOUS)

THOKKAVADI POST, TIRUCHENGODE-637215

NAMAKKAL DIST, TAMILNADU

RECORD NOTE BOOK

REG NO:

Certified that this is the bonafide record of work done by Selvan/Selvi…………………………………of the………….

Semester………………………………………Branch during the year……………..in the............................ laboratory.

Staff-in-charge **Head of the Department**

Submitted for the university practical

Examination on ……………………

Internal Examiner External Examiner

CONTENTS

EX.NO:1
DATE :

PAYROLL SYSTEM

Aim:

To prepare necessary documents and to develop the PAYROLL SYSTEM with UML diagrams using Software Engineering Methodology

PROGRAM ANALYSIS AND PROJECT PLANNING

Problem Statement:

This project PAYROLL SYSTEM is to develop an application to analyze the salary of the employee in a company. Admin can access all details in the application such as salary details of an employee. And can also update the salary details of an employee in a company. If the employee want the salary details the he want to request admin.

SOFTWARE REQUIREMENT ANALYSIS

The Modules in the Project:

1. Admin Login.
2. Change password
3. Add or update record
4. Calculate salary.
5. Delete record.
6. View records.

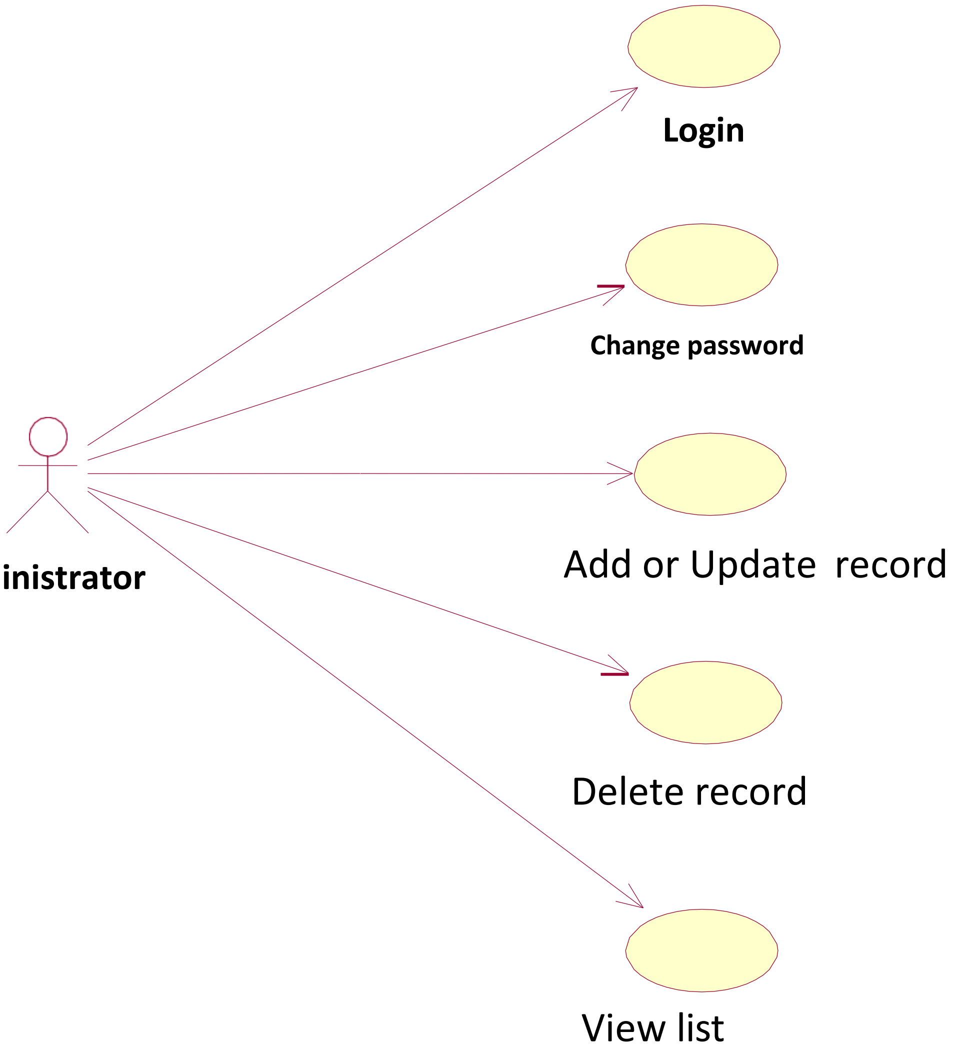
The first module is the user should login as an administrator with his login credentials such as username and password. Then the application verifies the username and password, then allow the user to continue.The second module allows the user to change current password to new password by taking input as current password, new password, confirm password.The third module is to add or update record by taking required employee details.The fourth module is to calculate salary for the entered input. The fifth module allows admin to remove a record from database.

DATA MODELING

Data Dictionary:

Field Name	Description	Data Type	Field size	Default value	Validation
EmpId	Identification Number	Integer	10	NULL	(0-9)
EmpName	Name of the Employee	String	20	NULL	(A-Z) or (a-z)
Experience	Experience of the Employee	String	10	NULL	(0-9)
BasicSalary	Basic salary of the Employee	Number	10	NULL	(0-9)
Allowances	Other than Basic Salary	Number	10	NULL	(0-9)
Nod	Number of days per month	Integer	5	NULL	(0-9)
Nol	Number of leave taken by Employee	Integer	5	NULL	(0-9)
pf	Provident fund of the Employee	Integer	10	NULL	(0-9)
Tax	Tax of basic salary	float	10	NULL	(0-9)
Salary	Take Home Salary	Integer	10	NULL	(0-9)

USE CASE DAIGRAM:



Use case Diagram:

Use case diagram is a graph of actors, set of use cases enclosed by a system boundary, communication (participation) association between the actors and the use cases and a generalization among the use cases.

Actor:

An actor represent a set of roles that user of a use case play when interacting with the use cases. Actor identified here is Administrator and Staff.

Use case:

A use case is a description of a set of sequence of actions that a system performs to yield result of value to an actor.

The Use Cases described are,

i. Login
ii. Display Salary
iii. Calculate Current Salary
iv. Add New Records
v. Update Records

The Login use case is to describe that, the user should choose his/her category whether he/she is a administrator or staff.

The use case Display Salary describes that, the system displays the salary detail of the staff from the database.

The use case Calculate Current Salary is to describe that, the administrator or the staff can calculate the current salary of basic salary and daily allowance available with the database.

The Add New records use case describes that; the administrator can add new records to the database.

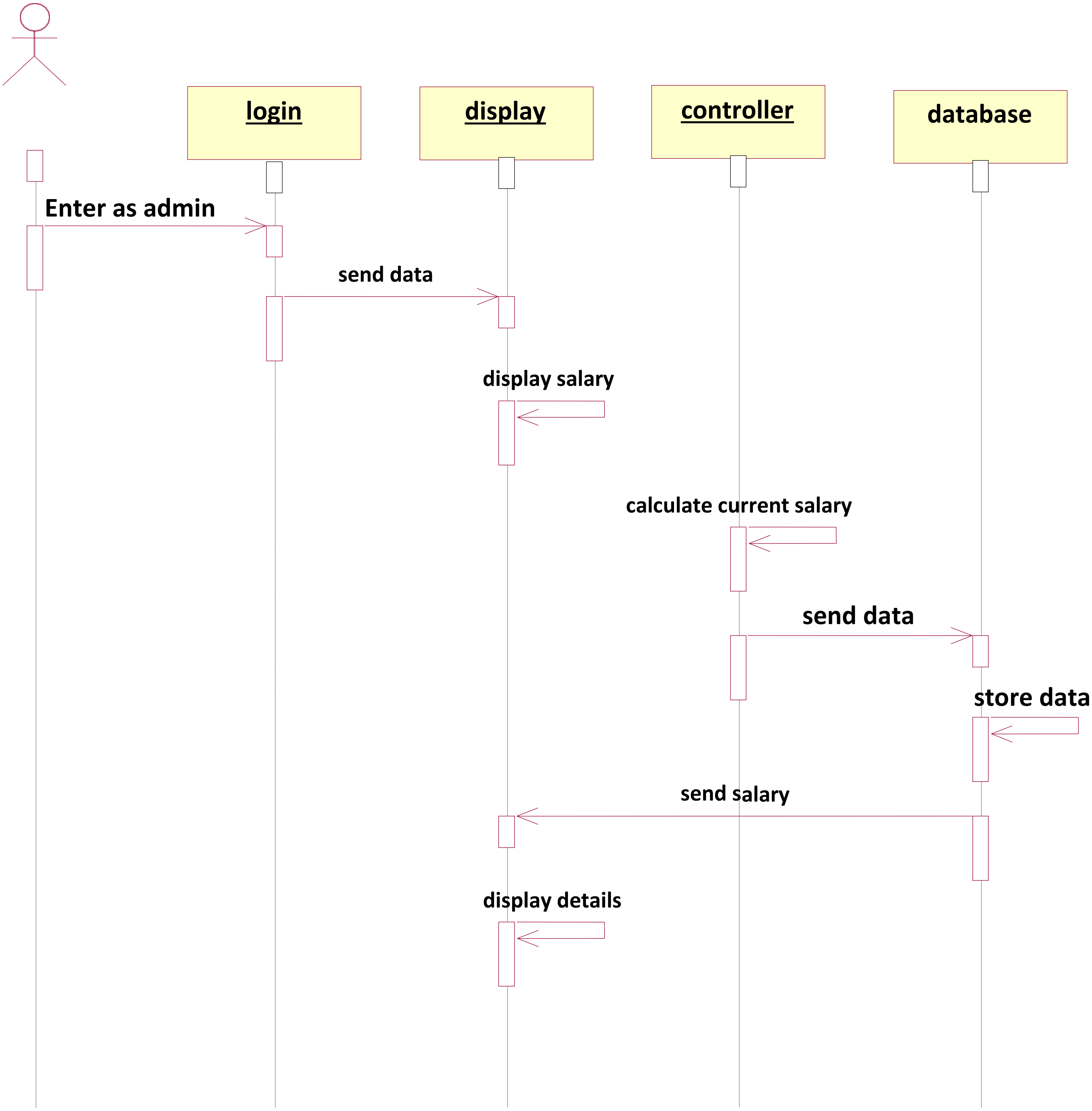
The Update Records use case describes that the Administrator can update the records.

SEQUENCE DIAGRAM

Sequence diagrams are easy and intuitive way of describing the behavior of a system by viewing the interaction between the system and its environment. A sequence diagram shows an interaction arranged in a time sequence.

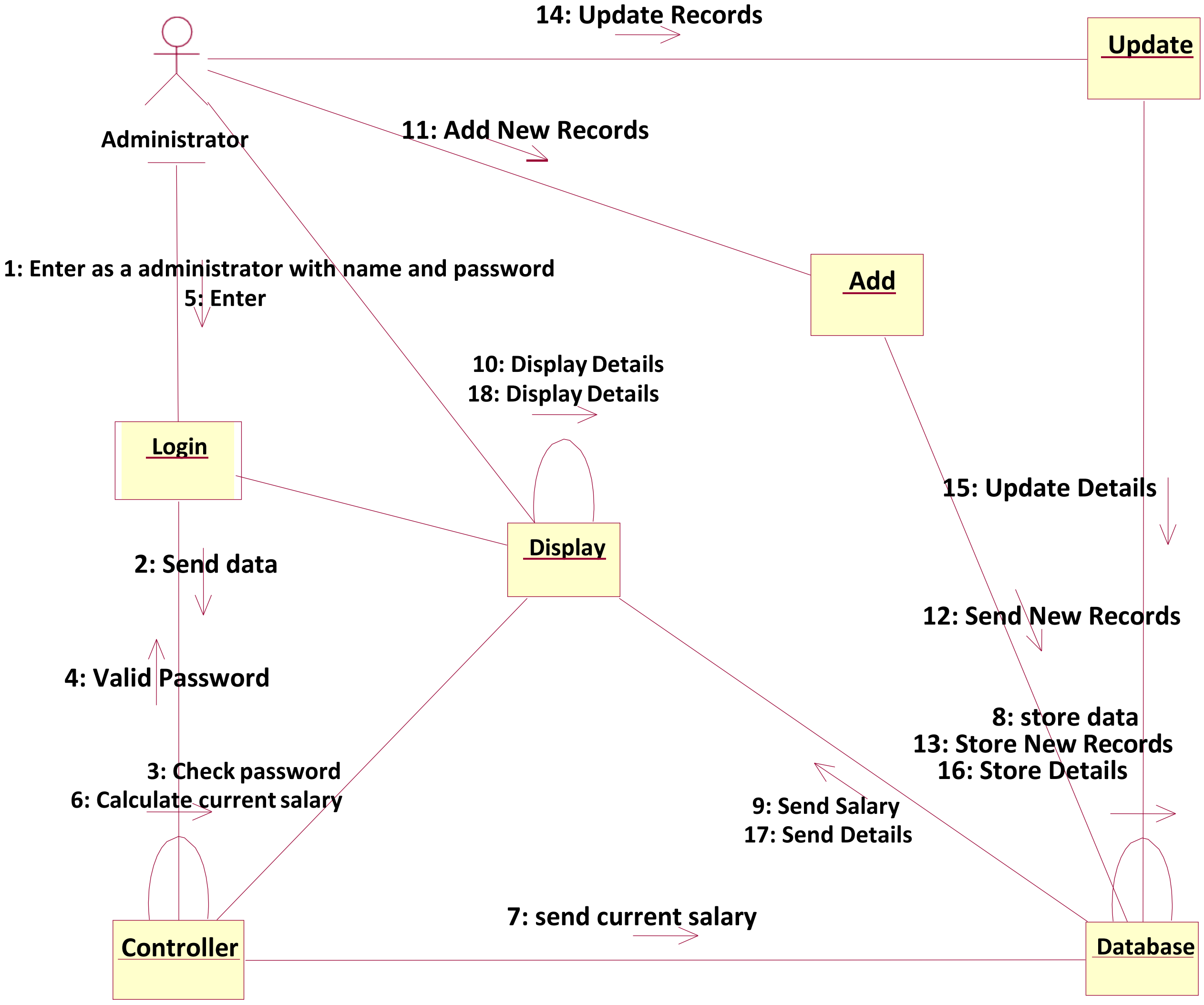
The objects used in this sequence diagram are,

1. Login
2. Display
3. Add
4. Update
5. Controller
6. Database


login
display
controller
database
Enter as admin
send data
display salary
calculate current salary
send data
store data
send salary
display details

COLLOBORATION DIAGRAM :

USER: ADMINISTRATOR



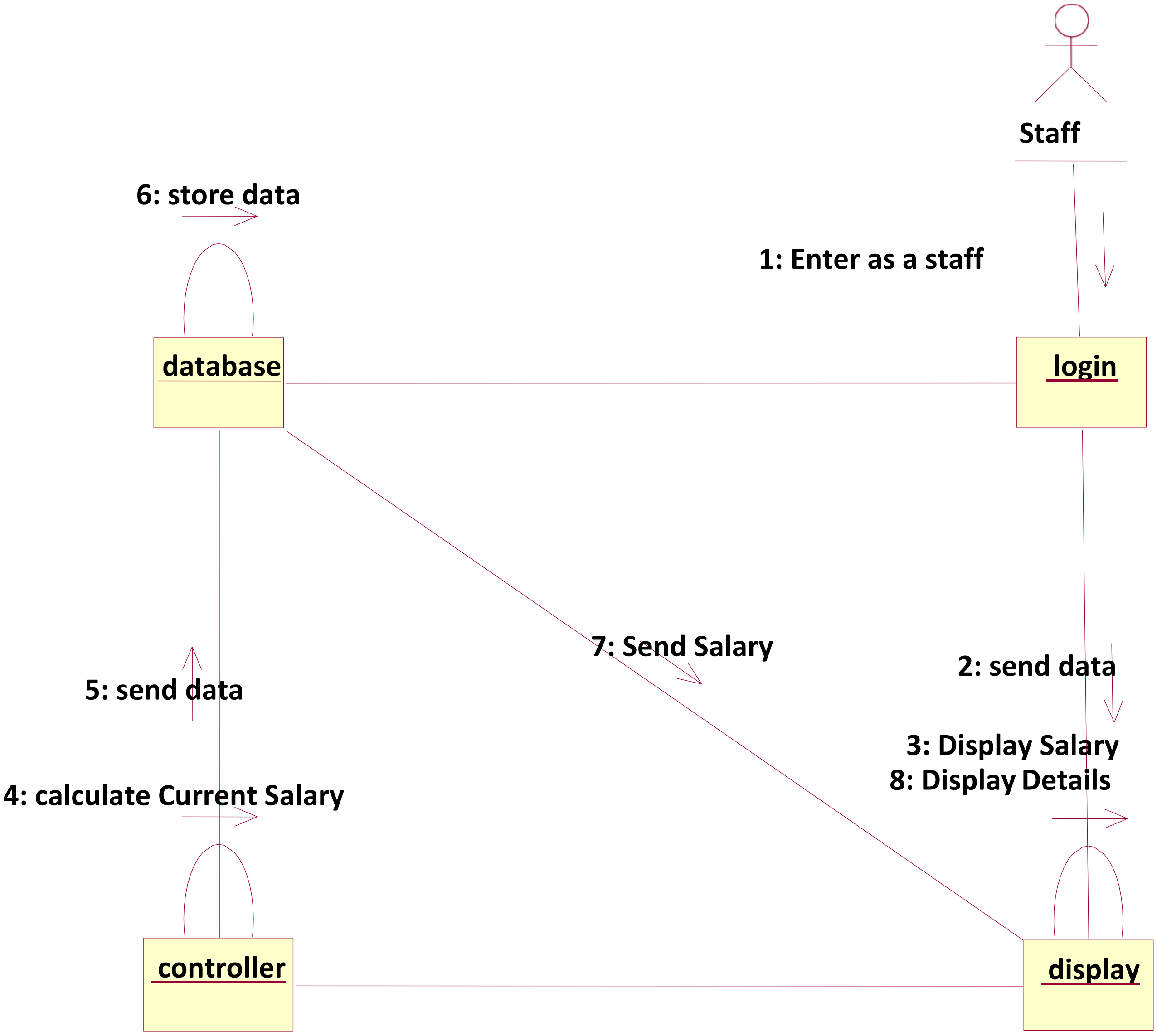
A collaboration diagram represents a collaboration, which is a set of objects related in a particular context, and interaction, which is a set of messages exchanged among the objects within the collaboration to achieve a desired outcome.

In this collaboration diagram, the objects are represented as rectangle, the actors are stick figures. Whereas the sequence diagram illustrates the object and actor interaction overtime, the collaboration diagram shows the object and actor interaction without reference to time.

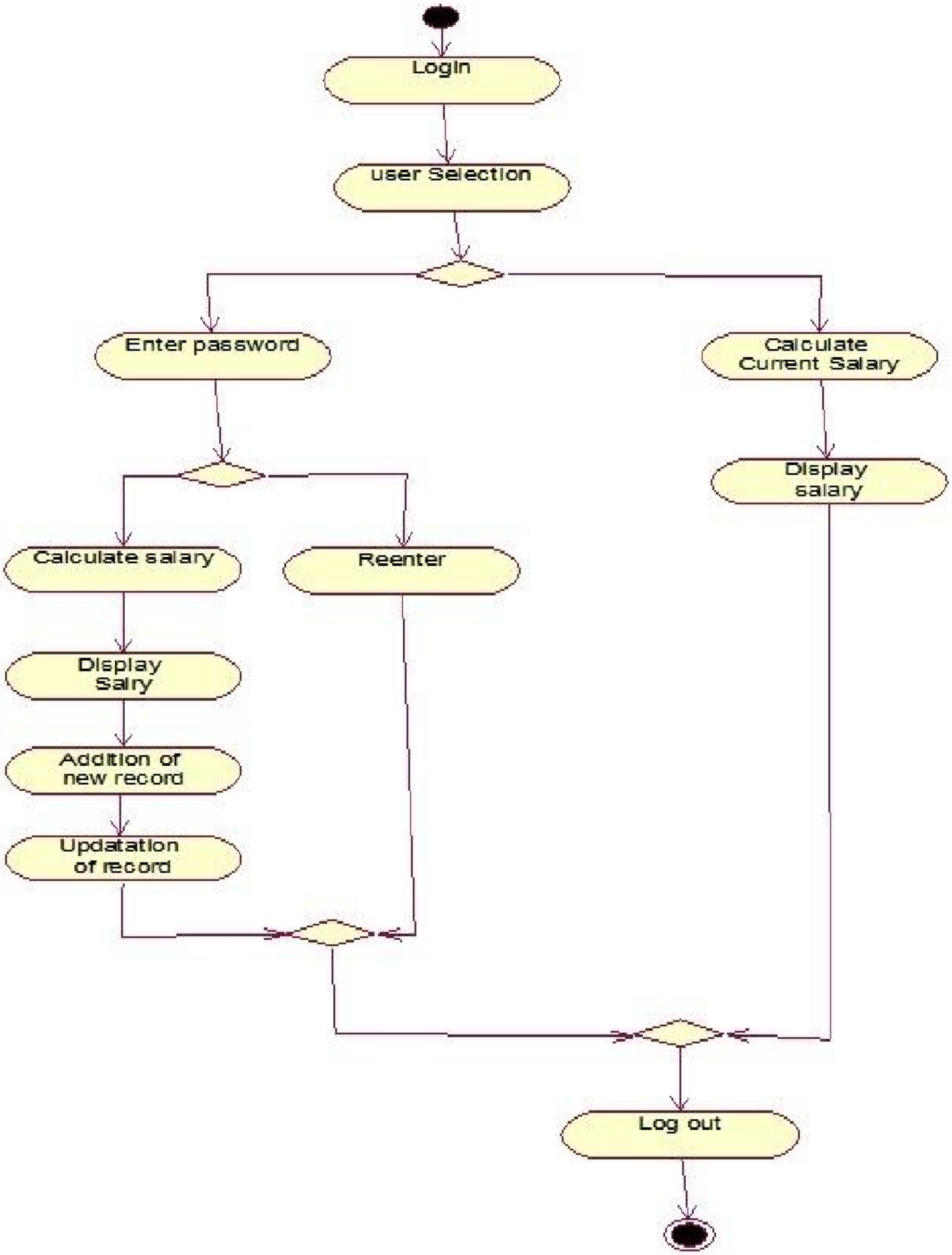
In our PAYROLL SYSTEM each object interacts with each other or collaborates with each other; it gets represented by the solid line drawn between them.

COLLOBORATION DIAGRAM:

User: STAFF



ACTIVITY DIAGRAM:



The activity diagram describes the sequencing of activities with support for both conditional and parallel behavior.

The Activity diagram is used to describe the various activities taking place in an application. Here in our PAYROLL SYSTEM, we have various activities starting from login.

After login, the user selection activity gets performed, where the user can be an administrator or staff.

If the user is a administrator, then they have to enter their name and password and only when those details are valid they can access the system. They can calculate the current salary obtained by the staffs, they can add new records, and they can update the values of the records which gets stored in the database.

If the user is a staff then they can view their salary detail and they can calculate their current salary.

CLASS DAIGRAM:4

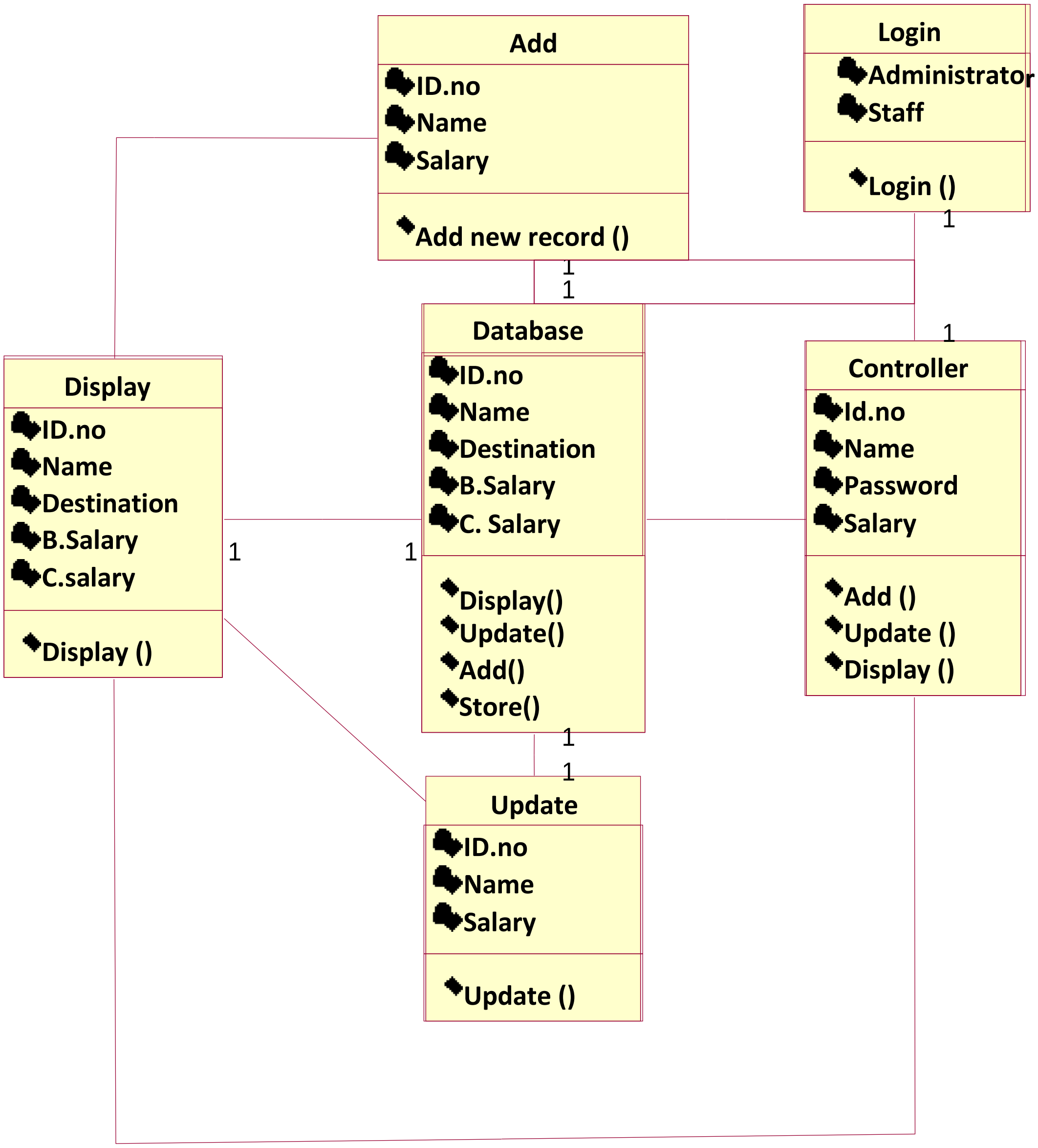
Class diagrams show the interactions between classes in the system. Class diagram also shows the attributes and operation of a class and the constraints that apply to the way objects are connected.

Classes contain information and behavior that acts on that information. Each class on class diagram is represented by rectangle divided into three sections. The first section shows the class name, second section shows the attributes the class contains and last section contains the operation of the class.

In our PAYROLL SYSTEM, the classes identified are

a. Login
b. Display
c. Add
d. Update
e. Controller
f. Database

Each class has its own attributes and operations.



Login class - The attributes defined is administrator and staff.
The method identified is login.

Display class - The attributes are id.no, name, destination, b.salary, c.salary.

The operation identified is Display.

Add class - The attributes are id.no, name, destination, b.salary, c.salary.

The operation defined is adding new record.

Update class - The attributes are id.no, name, salary,

The operation defined is update.

Controller class -The attributes it has is id.no, name, password, salary.

The operations carried out by this class are added, update and display.

Database class - The attributes are id.no, name, destination, b.salary, c.salary.

The operations defined are store, display, update, and add.

The Solid line between the classes shows the Association relationship between them.

SOFTWARE DEVELOPMENT

Login Form:

```
Private Sub Command1_Click()

If Text1.Text = "rv" And Text2.Text = "mk" Then

   payroll.Show

Else

MsgBox ("Invalid input")

End If

End Sub
```

Payroll Form:

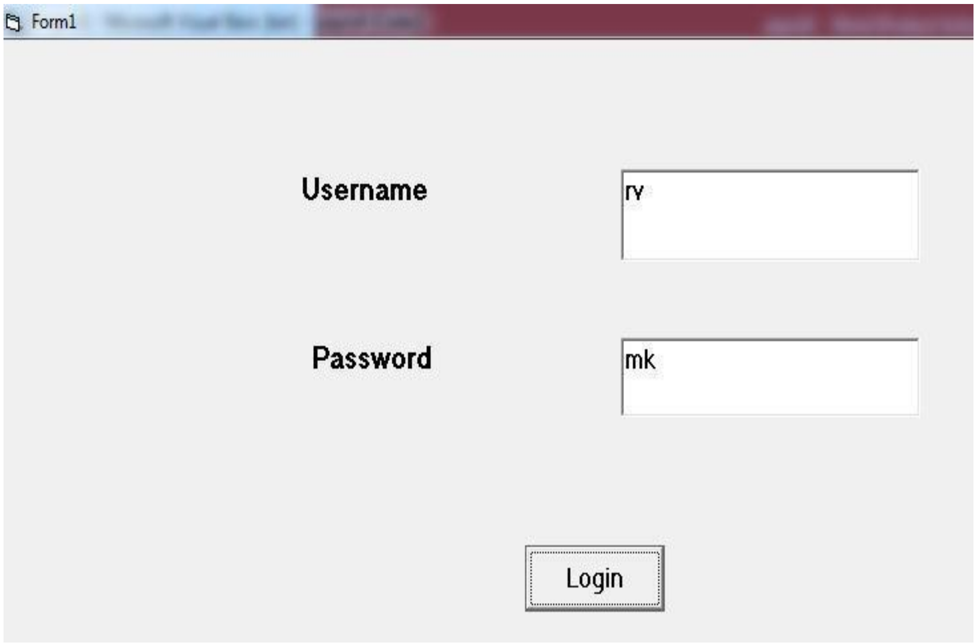
```
Dim id As Integer

Private Sub cmdadd_click()

 If Adodc1.Recordset.RecordCount = 0 Then

 id = 0

 Else

 Adodc1.Recordset.MoveLast
 id = Mid(Adodc1.Recordset("empid"), 2, 3) + 1
 End If
  Adodc1.Recordset.AddNew
  textid = Format(id, "E000")
  textname.SetFocus
End Sub
Private Sub cmdexit_click()
End
End Sub
Private Sub cmdsave_click()
Dim flag As Boolean
flag = False
Select Case combodesig
Case "manager"
If Val(textbasic.Text) > 15000 And Val(textbasic.Text) <= 25000 Then
textda.Text = Val(textbasic) * 0.08
texthra.Text = Val(textbasic) * 0.03
textdeduct.Text = Val(textbasic) * 0.01
textgross.Text = Val(textbasic) + Val(textda) + Val(texthra)
textnet.Text = Val(textgross) - Val(textdeduct)
flag = True
Else
MsgBox "input basic pay between 15k and 25k", vbInformation + vbOKOnly
```

```
End If
Case "engineer"
If Val(textbasic.Text) > 15000 And Val(textbasic.Text) <= 20000 Then
textda.Text = Val(textbasic) * 0.07
texthra.Text = Val(textbasic) * 0.04
textdeduct.Text = Val(textbasic) * 0.01
textgross.Text = Val(textbasic) + Val(textda) + Val(texthra)
textnet.Text = Val(textgross) - Val(textdeduct)
flag = True
Else
MsgBox "input basic pay between 15k and 20k", vbInformation + vbOKOnly
End If
Case "clerk"
If Val(textbasic.Text) > 5000 And Val(textbasic.Text) <= 15000 Then
textda.Text = Val(textbasic) * 0.04
texthra.Text = Val(textbasic) * 0.02
textdeduct.Text = Val(textbasic) * 0.01
textgross.Text = Val(textbasic) + Val(textda) + Val(texthra)
textnet.Text = Val(textgross) - Val(textdeduct)
flag = True
Else
MsgBox "input basic pay between 5k and 15k", vbInformation + vbOKOnly
End If
```

```
Case "messenger"

If Val(textbasic.Text) > 5000 And Val(textbasic.Text) <= 10000 Then

textda.Text = Val(textbasic) * 0.03

texthra.Text = Val(textbasic) * 0.02

textdeduct.Text = Val(textbasic) * 0.01

textgross.Text = Val(textbasic) + Val(textda) + Val(texthra)

textnet.Text = Val(textgross) - Val(textdeduct)

flag = True

Else

MsgBox "input basic pay between 5k and 10k", vbInformation + vbOKOnly

End If

End Select

If flag Then

Adodc1.Recordset.Update

MsgBox "employee details saved", vbOKOnly

End If

End Sub
```

Output:

Form1:

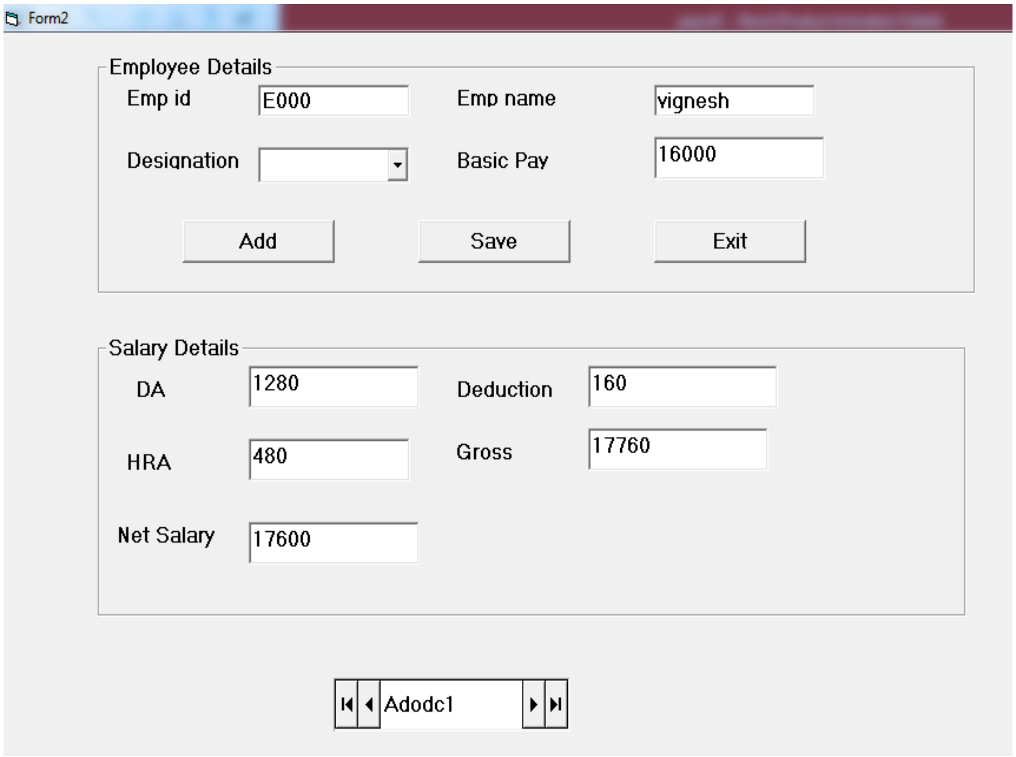


Form1

Username: ry

Password: mk

Form2:



Form2

Employee Details

Emp id: E000

Emp name: vignesh

Designation:

Basic Pay: 16000

Salary Details

DA: 1280

Deduction: 160

HRA: 480

Gross: 17760

Net Salary: 17600

SOFTWARE TESTING:

TEST CASES:

- If an unauthorized user attempts to access the system, the system should not allow them to access.

- If the password entered by the administrator is incorrect, the system should display the bad password message and allows them to reenter the password or to terminate the process.

RESULT:

Thus the **PAYROLL SYSTEM** is developed with all necessary documents and UML diagrams using Software Engineering methodology.

EX.NO:2
DATE :

ONLINE SHOPPING

AIM

To prepare necessary documents and to develop ONLINE SHOPPING with UML diagrams using Software Engineering Methodology.

PROGRAM ANALYSIS AND PROJECT PLANNING

Problem Statement:

This project ONLINE SHOPPING is to develop an application for shopping. Initially a user should choose the category whether he\she is a Customer or Administrator. If the person is an Administrator then they can add the products. If the user is a Customer then he\she can register an account, then choose and buy the products through online.

SOFTWARE REQUIREMENT ANALYSIS

The Modules in the Project:

1. Login.
2. Add Products.
3. User Registration.
4. Choose Products.
5. Calculate Total Amount.
6. Online Payment.

The first module is Login in which the user has to login to the system as a Customer or Administrator.

The next module is Add Products, where if the user is an administrator, then he\she add the new products.

The next module is User Registration the new customer can register to create an account.

The next module is Choose Products, the customer choose the products.

The next module is Calculate Total Amount, here the total amount for the products are calculated.

The last module is the Online Payment the payment is transferred through online.

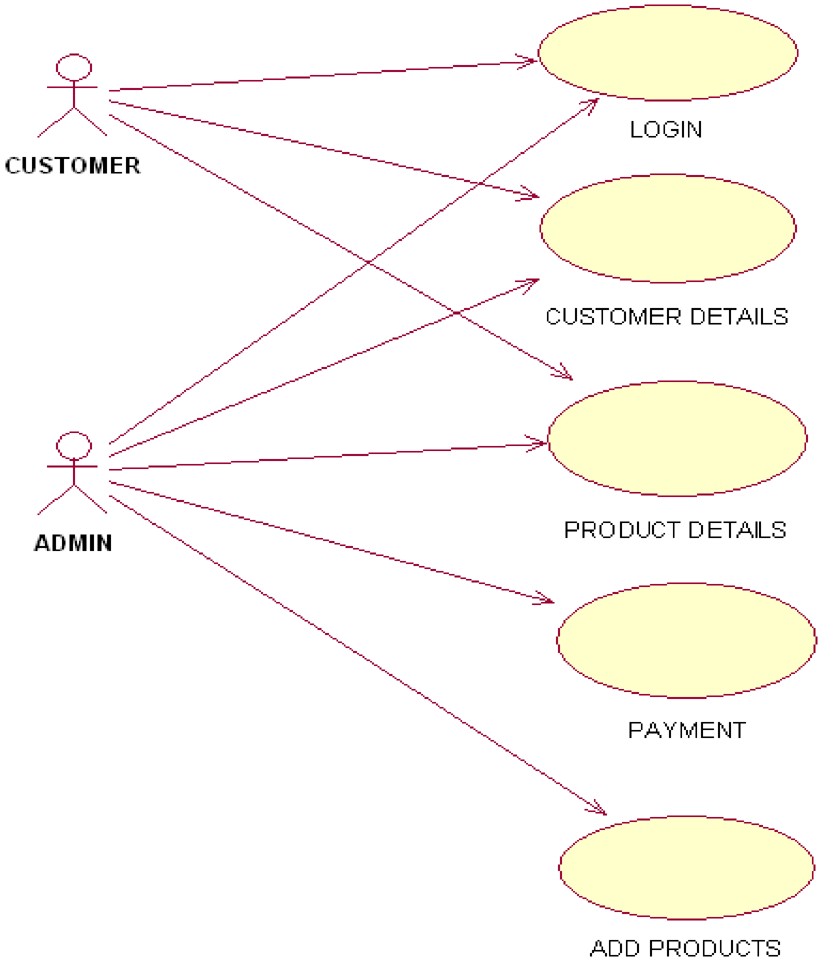
DATA MODELING

Data Dictionary:

Field Name	Description	Data Type	Field size	Default value	Validation
Name	Name of the customer	String	50	NULL	(A-Z) or (a-z)
Pass wd	Password for the customer	String	50	NULL	(A-Z) or (a-z)
dob	Date of Birth	Date/Time	8	NULL	(0-9)
Gender	Gender	String	50	NULL	(A-Z) or (a-z)
Address	Address of the customer	String	70	NULL	(A-Z) or (a-z)
Ph no	Phone no of the customer	Double	8	NULL	(0-9)
Mail id	Customer’s Mail id	String	50	NULL	(A-Z) or (a-z)
Pi d	Product id	Integer	2	NULL	(0-9)
P name	Name of the Product	String	50	NULL	(A-Z) or (a-z)
P type	Type of the Product	String	50	NULL	(A-Z) or (a-z)
C name	Company Name	String	50	NULL	(A-Z) or (a-z)
price	Price of the Product	Currency	8	NULL	(0-9)
quantity	Quantity of the Product	Integer	2	NULL	(0-9)

t payment	Type of Payment	String	50	NULL	(A-Z) or (a-z)
b name	Name of the Bank	String	50	NULL	(A-Z) or (a-z)
acc no	Account number for the Bank	Integer	8	NULL	(0-9)
date	Current Date	Date	8	NULL	(0-9)

USE CASE DIAGRAM:



Use case Diagram:

Use case diagram is a graph of actors, set of use cases enclosed by a system boundary, communication association between the actors and the use cases and a generalization among the use cases.

Actor:

An actor represent a set of roles that user of a use case play when interacting with the use cases. Actor identified here is Administrator and Customer.

Use case:

A use case is a description of a set of sequence of actions that a system performs to yield result of value to an actor.

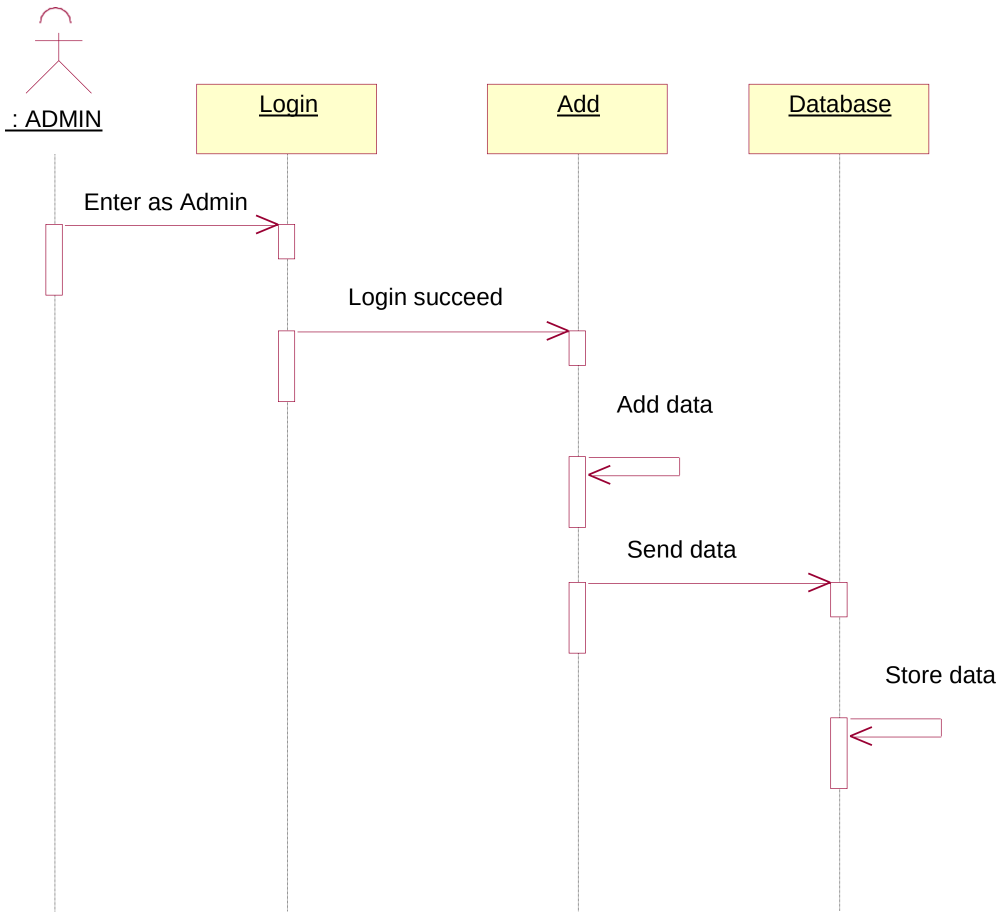
The Use Cases described are,

1. Login
2. Customer Details
3. Product Details
4. Payment
5. Add Products

- The Login use case is to describe that, the user should choose his/her category whether he/she is a administrator or customer.
- The use case Customer Details describes that, the details of the customers are added into the database.
- The use case Product Details describes that, it displays the list of products from the database.
- The Payment use case describes that, the payment options for online shopping.
- The Add Products use case describes that the Administrator can add the new products.

SEQUENCE DIAGRAM

User: ADMIN



Sequence Diagram:

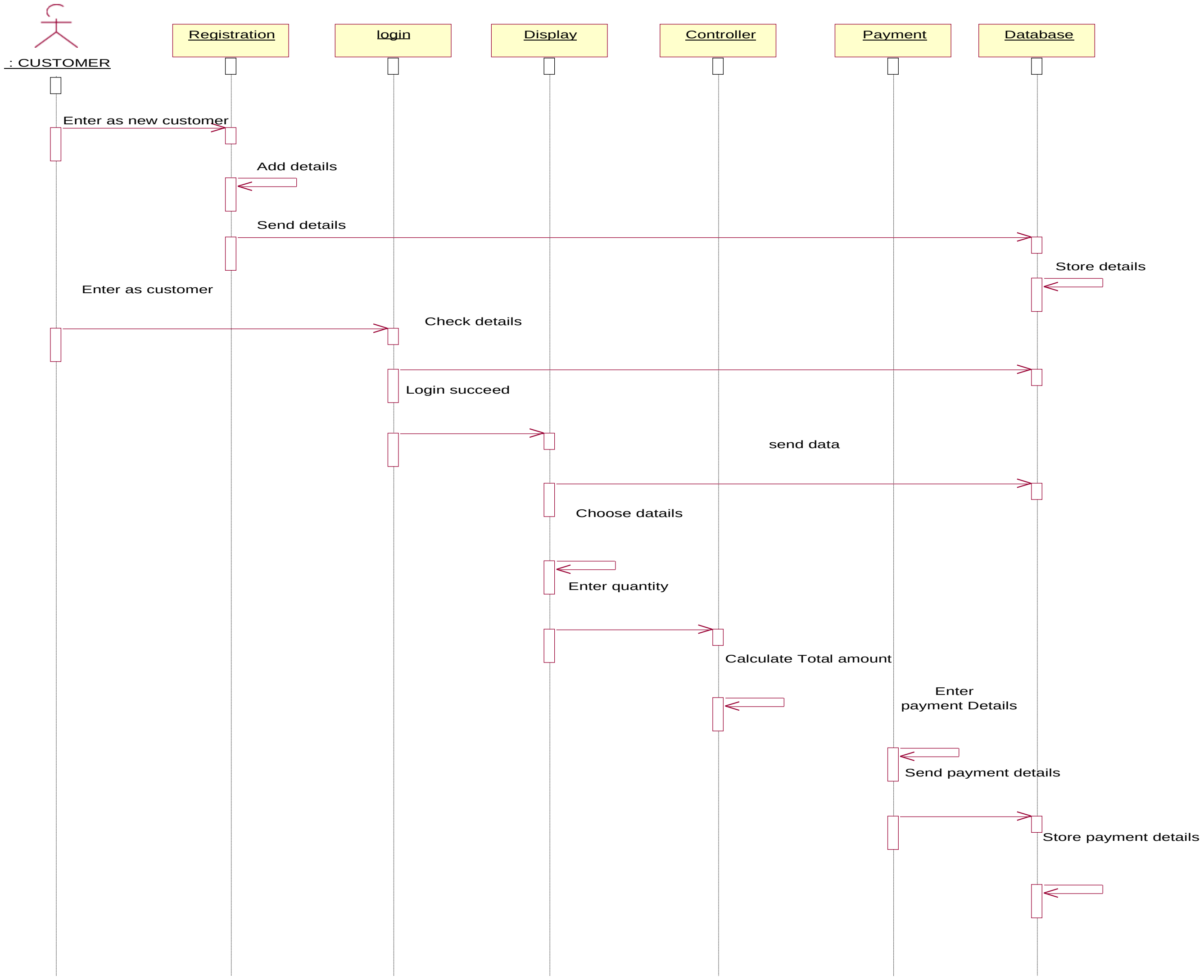
Sequence diagrams are easy and intuitive way of describing the behavior of a system by viewing the interaction between the system and its environment. A sequence diagram shows an interaction arranged in a time sequence.

The objects used in this sequence diagram are,

1. Login
2. Add
3. Display
4. Controller
5. Payment
6. Database

SEQUENCE DIAGRAM

User: CUSTOMER



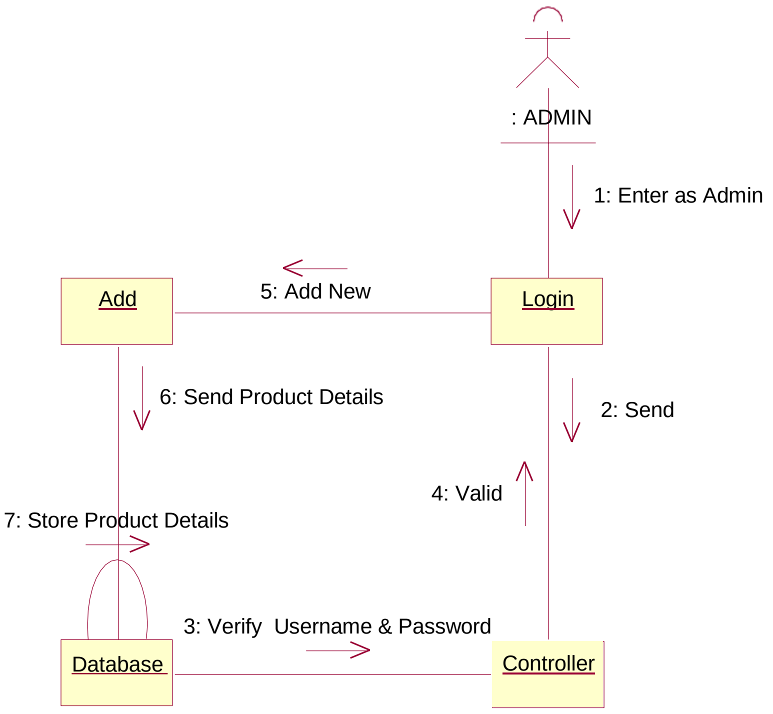
Basic Flow:

The administrator or customer enters their name and password, and the password gets checked by the system. After confirmation of the password the system allows them to access.

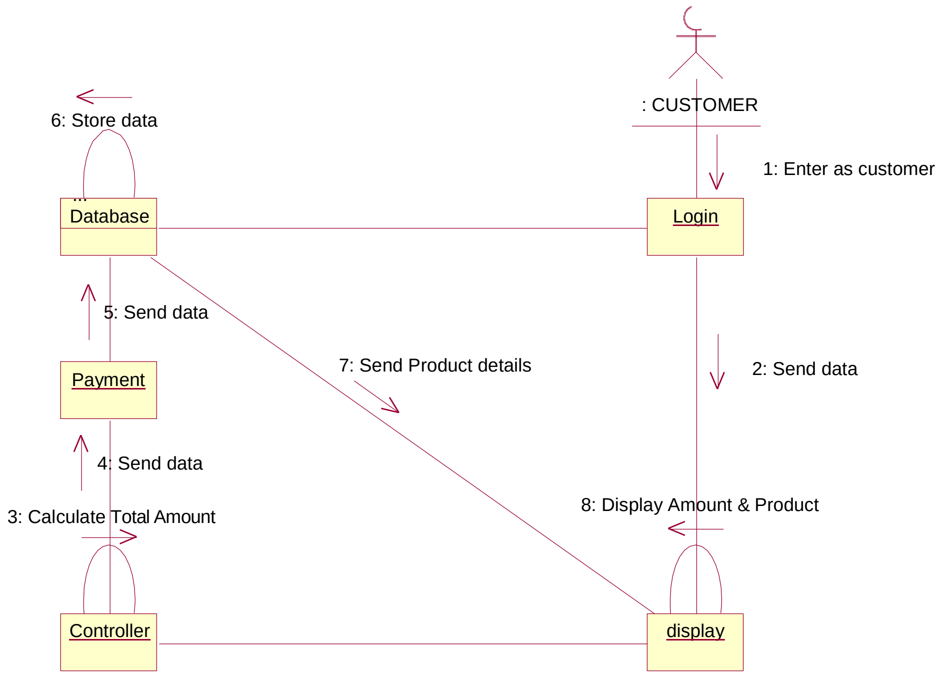
Alternate Flow:

- The actor Administrator and Customer are the persons who interacts with the system.
- The object Login makes the administrator/ customer to enter.
- The object Add will add the new products into the database.
- The object Display will display the product details from the database.
- The object Payment will allows to choose the mode of payment.
- The object Database will store all the product details & customer details.

COLLABORATION DIAGRAM :
USER: ADMIN



COLLABORATION DIAGRAM:
User: CUSTOMER



Collaboration Diagram:

A collaboration diagram represents a collaboration, which is a set of objects related in a particular context, and interaction, which is a set of messages exchanged among the objects within the collaboration to achieve a desired outcome.

Collaboration diagram shows exactly the same information as the sequence diagram. However, collaboration diagram shows this information in a different way and with different purpose.

In this collaboration diagram, the objects are represented as rectangle, the actors are stick figures. Whereas the sequence diagram illustrates the object and actor interaction overtime, the collaboration diagram shows the object and actor interaction without reference to time.

In our ONLINE SHOPPING each object interacts with each other or collaborates with each other; it gets represented by the solid line drawn between them.

ACTIVITY DIAGRAM:

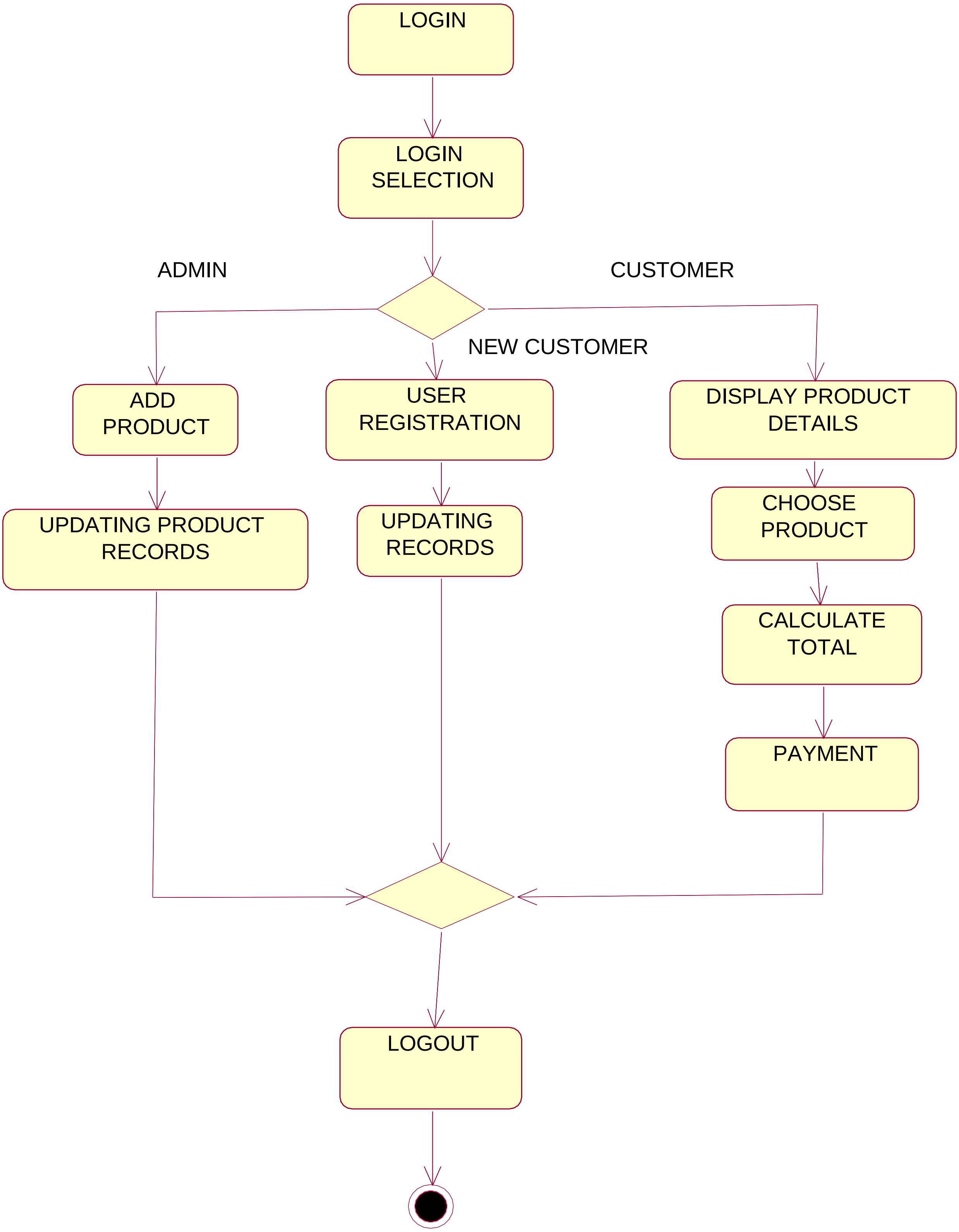
The activity diagram describes the sequencing of activities with support for both conditional and parallel behavior.

The Activity diagram is used to describe the various activities taking place in an application. Here in our ONLINE SHOPPING, we have various activities starting from login.

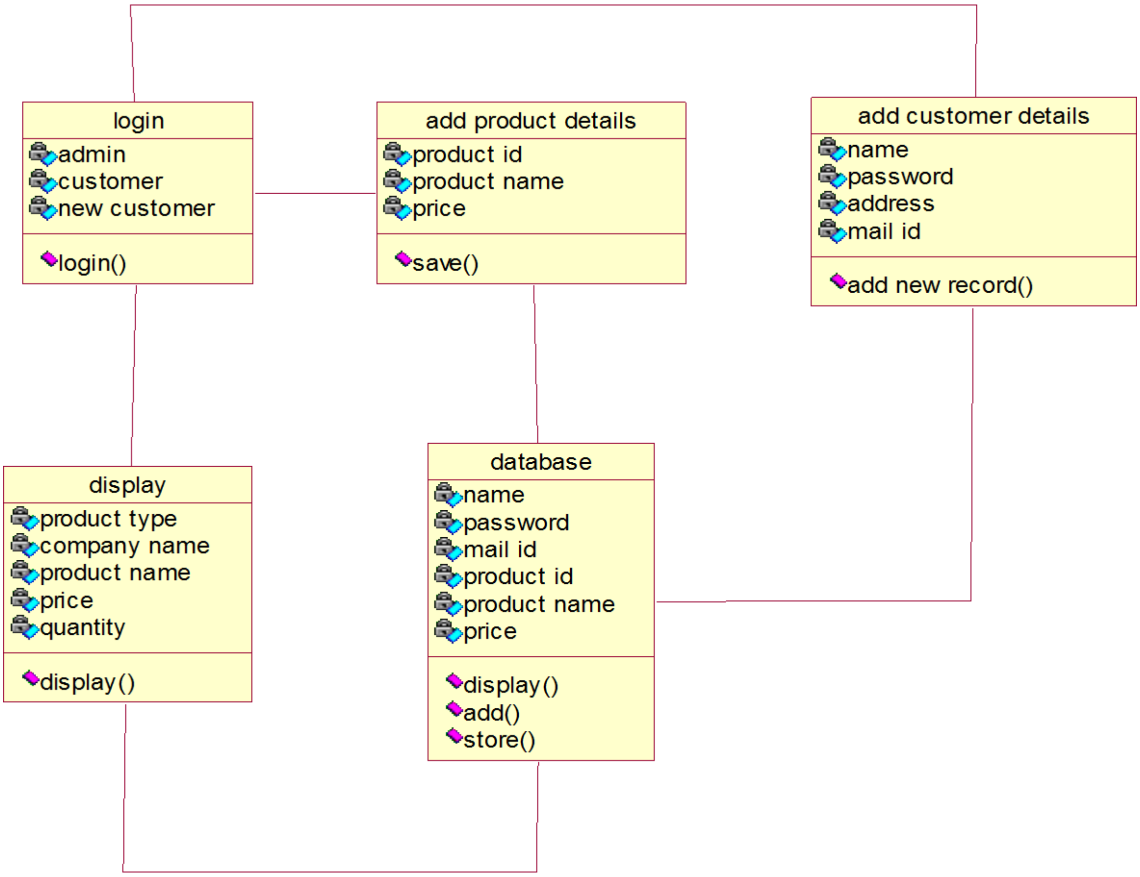
After login, the user selection activity gets performed, where the user can be an administrator or customer.

If the user is a administrator, then they have to enter their name and password and only when those details are valid they can access the system. They can add the new products which gets stored in the database.

If the user is a customer, then they login into the system using valid username and password, after login they can choose the products and also buy the products.


LOGIN
LOGIN SELECTION
ADMIN
CUSTOMER
NEW CUSTOMER
ADD PRODUCT
USER REGISTRATION
DISPLAY PRODUCT DETAILS
UPDATING PRODUCT RECORDS
UPDATING RECORDS
CHOOSE PRODUCT
CALCULATE TOTAL
PAYMENT
LOGOUT

CLASS DIAGRAM:



Class Diagram:

Class diagrams show the interactions between classes in the system. Class diagram also shows the attributes and operation of a class and the constraints that apply to the way objects are connected.

Classes contain information and behavior that acts on that information. Each class on class diagram is represented by rectangle divided into three sections. The first section shows the class name, second section shows the attributes the class contains and last section contains the operation of the class.

In our ONLINE SHOPPING, the classes identified are

1. Login
2. Add Product Details
3. Add Customer Details
4. Display
5. Database

Each class has its own attributes and operations.

Login class - The attributes defined are administrator, customer. and new customer .The method identified is login.

Add Product Details class - The attributes are product id, product name, Price.The operation defined is saving new record.

Add Customer Details class - The attributes are name, password, address, mail .The operation defined is adding new record.

Display class - The attributes are product name, company name, price, quantity.The operation identified is Display.

Database class - The attributes are name, password, mail id, product name, price.The operations defined are display, add and store.

The Solid line between the classes shows the Association relationship between them.

SOFTWARE DEVELOPMENT:

Login Form:

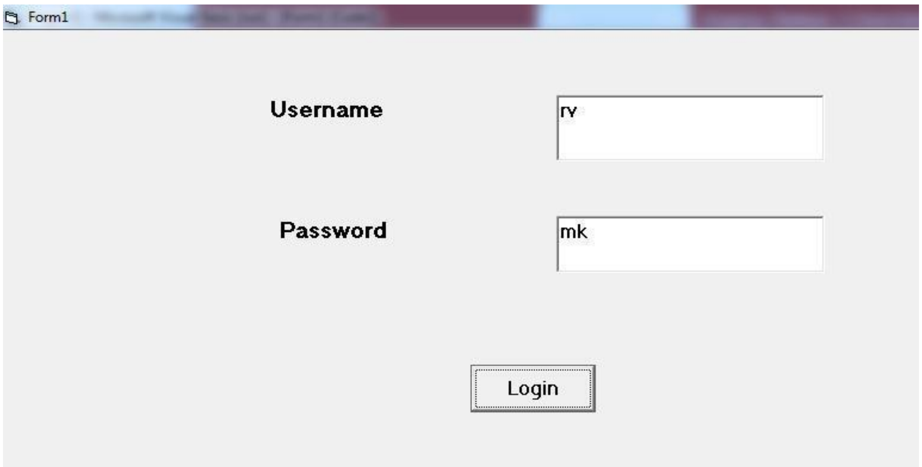
```
Private Sub
Command1_Click()
If Text1.Text = "rv" And Text2.Text = "mk" Then
 shopping.Show
Else
MsgBox ("Invalid input")
End If
End Sub
```

Shopping Form:

```
Private Sub buybtn_Click()
shoppingado.Recordset.Fields("name") = txtname.Text
shoppingado.Recordset.Fields("date of birth") = txtdob.Text
shoppingado.Recordset.Fields("gender") = txtgender.Text
shoppingado.Recordset.Fields("Address") = txtadd.Text
shoppingado.Recordset.Fields("Contact no") = txtphone.Text
shoppingado.Recordset.Fields("mail") = txtmail.Text
shoppingado.Recordset.Fields("product name") =
txtproduct.Text
shoppingado.Recordset.Fields("price") =txtprice.Text
shoppingado.Recordset.Fields("quantity") = txtquantity.Text
shoppingado.Recordset.Fields("total amount") =
txtamount.Text
shoppingado.Recordset.Fields("payment") = txtpayment.Text
shoppingado.Recordset.Fields("code word") = txtcode.Text
shoppingado.Recordset.Update
MsgBox "User Shopping Successful"
End Sub

Private Sub
Command2_Click()
End
End Sub

Private Sub Form_Load()
shoppingado.Recordset.AddNew
End Sub
```

OUTPUT:

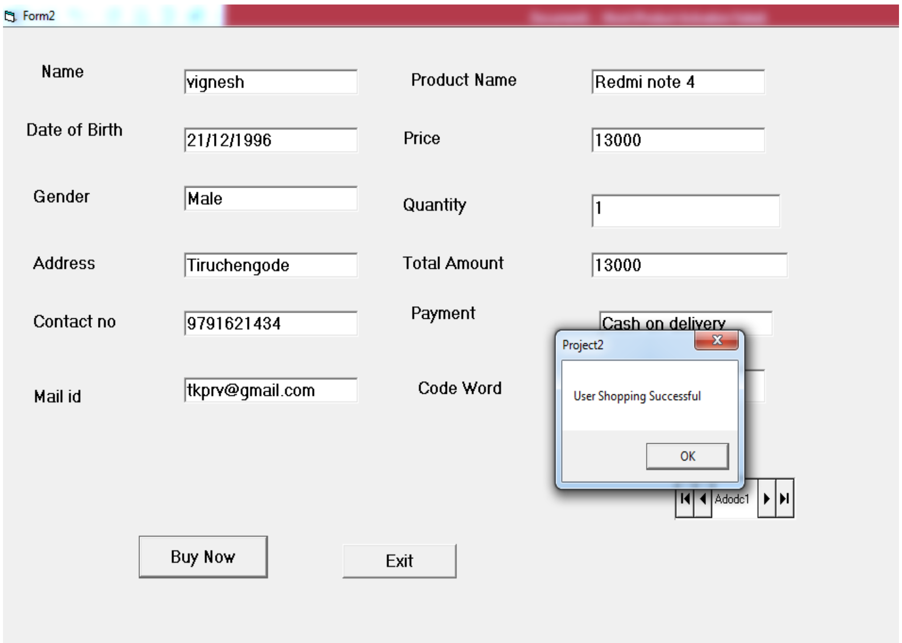
Form1:



Username rv

Password mk

Form2:



Name vignesh

Date of Birth 21/12/1996

Gender Male

Address Tiruchengode

Contact no 9791621434

Mail id tkprv@gmail.com

Product Name Redmi note 4

Price 13000

Quantity 1

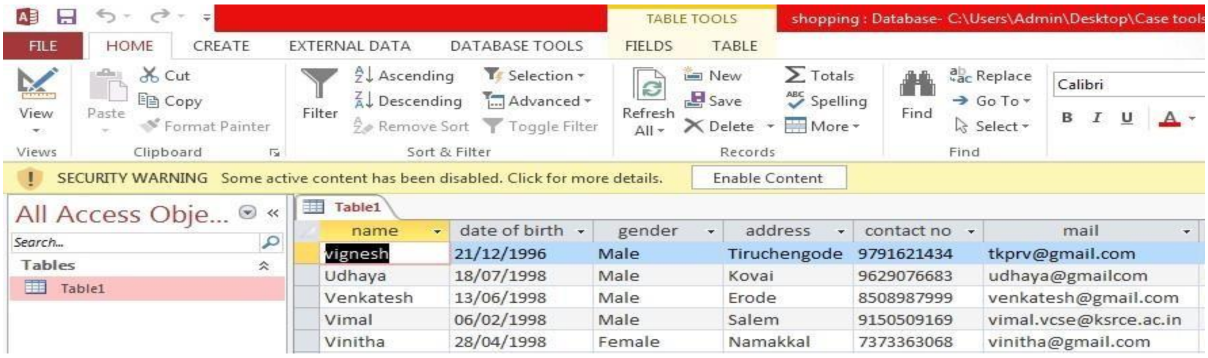
Total Amount 13000

Payment Cash on delivery

Code Word

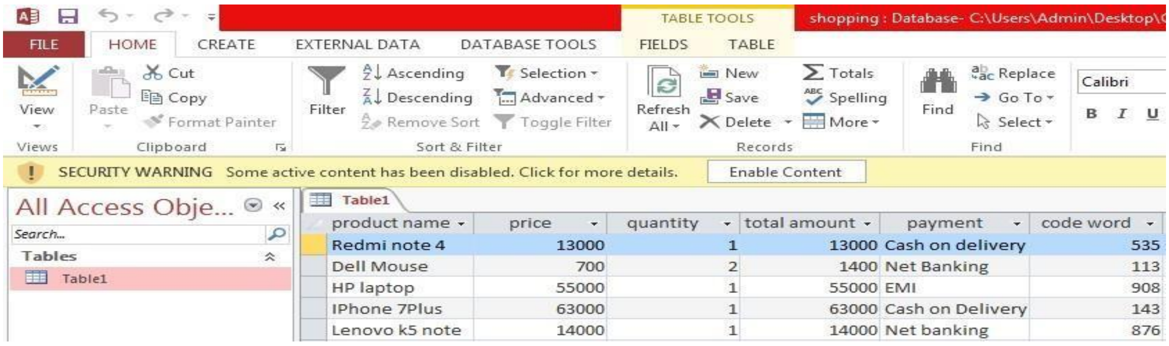
User Shopping Successful

Database Form:



SECURITY WARNING Some active content has been disabled. Click for more details.

name	date of birth	gender	address	contact no	mail
vignesh	21/12/1996	Male	Tiruchengode	9791621434	tkprv@gmail.com
Udhaya	18/07/1998	Male	Kovai	9629076683	udhaya@gmailcom
Venkatesh	13/06/1998	Male	Erode	8508987999	venkatesh@gmail.com
Vimal	06/02/1998	Male	Salem	9150509169	vimal.vcse@ksrce.ac.in
Vinitha	28/04/1998	Female	Namakkal	7373363068	vinitha@gmail.com



SECURITY WARNING Some active content has been disabled. Click for more details.

product name	price	quantity	total amount	payment	code word
Redmi note 4	13000	1	13000	Cash on delivery	535
Dell Mouse	700	2	1400	Net Banking	113
HP laptop	55000	1	55000	EMI	908
IPhone 7Plus	63000	1	63000	Cash on Delivery	143
Lenovo k5 note	14000	1	14000	Net banking	876

RESULT:

Thus the ONLINE SHOPPING is developed with all necessary documents and UML diagrams using Software Engineering methodology.

EX.NO:3
DATE

BANKING SYSTEMS

AIM.:

To develop a software for banking system by using software engineering methodology.

ABSTRACT:

All the banks are centralized with computers nowadays. All the accounts maintained in the bank and also the transaction effected, including banking transaction, are to be processed by the computers in the bank. An BANKING accepts relevant cash card, interacts with user, communicates with central system to carry out the transaction such as amount deposit,withdraw,balance enquiry and prints receipts. The system to be designed and implement must include appropriate record keeping and security provisions. The system must handle concurrent access to the same amount.

SYSTEM SPECIFICATION:

Software Requirements:

Operating system	: Windows xP
Front end	: Visual Basic 6.0
Back end	: Microsoft Access

Hardware Requirements:

Processor	: Intel Pentium @ 3.06GHz
RAM	: 512MB DDR
Hard Disk	: 80GB SATA
Monitor	: 15’’TFT
Keyboard	: Multimedia Keyboard
Mouse	: USB Optical type

PROGRAM ANALYSIS AND PROJECT PLANNING

PROBLEM STATEMENT:

This application BANKING System helps the bank client to access his/her account. The client has to first enter his/her name and account id number. Once when the details are confirmed the client can access the account, if the details are wrong then the client has to reenter the details. Next the system should ask for the type of transaction to perform, where the client has to choose the type of transaction they wish to carry on. And based on the transaction chosen the actions have to be carried out. And based on their transaction their account has to be updated.

SOFTWARE REQUIREMENT ANALYSIS

Modules.:

1. Card verification.
2. Selecting transaction.
3. Deposit.
4. Withdraw.

Card verification:

This module is an Approval process in which the client has to enter his/her name and account ID number, these details will be referred with the database and the service will be provided if the details are correct.

Selecting Transaction:

This module selects transaction in which the system displays the types of transaction (Deposit/Withdraw/Balance enquiry), the client has to choose any one, and the transaction will be carried on accordingly.

Deposit:

This module is Deposit saving module where the client has to enter the amount to be deposited. The account will be updated automatically and saved.

Withdraw:

This is a module where the client has to enter the amount to withdraw. After verifying that the funds are sufficient, the transaction is performed.

DATA MODELING

Data Dictionary:

Field Name	Description	Data Type	Field size	Default value	Validation
Name	Name of the client	String	20	NULL	(A-Z) or (a-z)
D.O.B	Date of birth	Date and time	10	NULL	(0-9)
Gender	Specify the gender	string	20	NULL	(A-Z) or (a-z)
Acc.no	Account Number of the Client	Integer	10	NULL	(0-9)
Balance	Amount In the client's Account	Integer	20	NULL	(0-9)

USE CASE DIAGRAM:

Use case diagram is a graph that describes the association between the actors and the use cases and a generalization among the use cases.

Actor:

An actor represent a set of roles that user of a use case play when interacting with the use cases. Actor identified here is the user and banker.

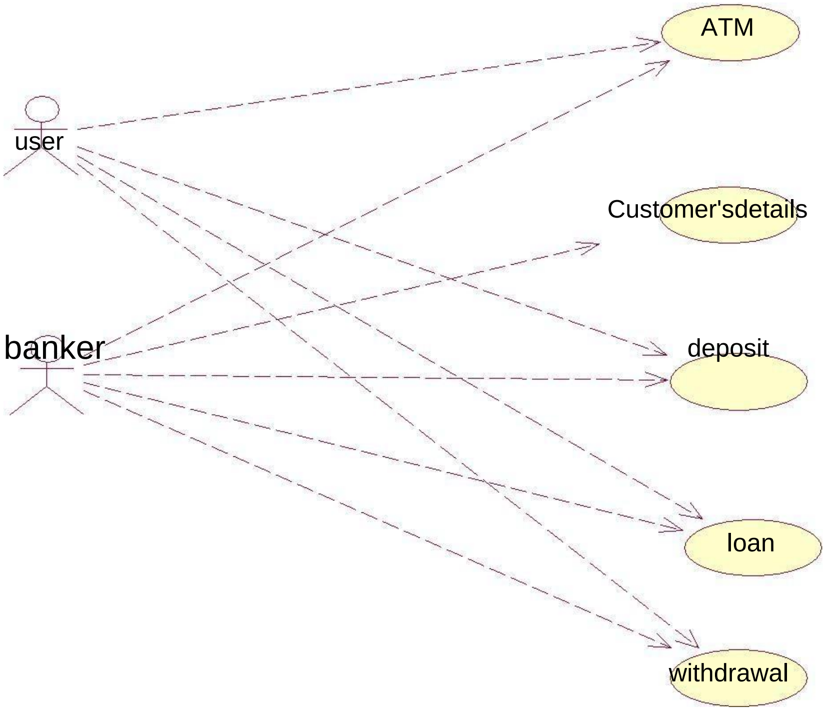
Use case:

A use case is a description of a set of sequence of actions that a system performs to yield result of value to an actor.

The Use Cases here it describes,

- Customers insert a card and verify it. Then he selects the type of transaction.
- BANKING performs the required transaction.

After transaction completes customer takes the card.



UML class diagram is the main static analysis diagram. This diagram shows the static structure of the model. This is a collection of static modeling elements, such as classes and their relationships, connected as a graph to each other and to their contents.

A class is drawn as a rectangle with three fields. The top field contains class name, the middle field contains attributes and the bottom field contains list of operations.

In our BANKING application, the classes are

1. New account
2. Deposit
3. Withdraw
4. Balance check
5. Transactions

Each class has its own attributes and properties

New account class:

The attributes involved are name,acc.no,dob,..
The operations involved are created and cancel.

Deposit class:

The attributes involved are account number and balance.
The operations involved are add and exit.

Withdraw class:

The attributes involved are account number and balance.
The operations involved are save and cancel.

Balance checking class:

This involves attributes like name, account number and balance amount.
The operation involved is getbalance.

Transactions:

This class involves attributes like name, account number, last transaction, date and amount in that account.
The operation involved is getdetails.

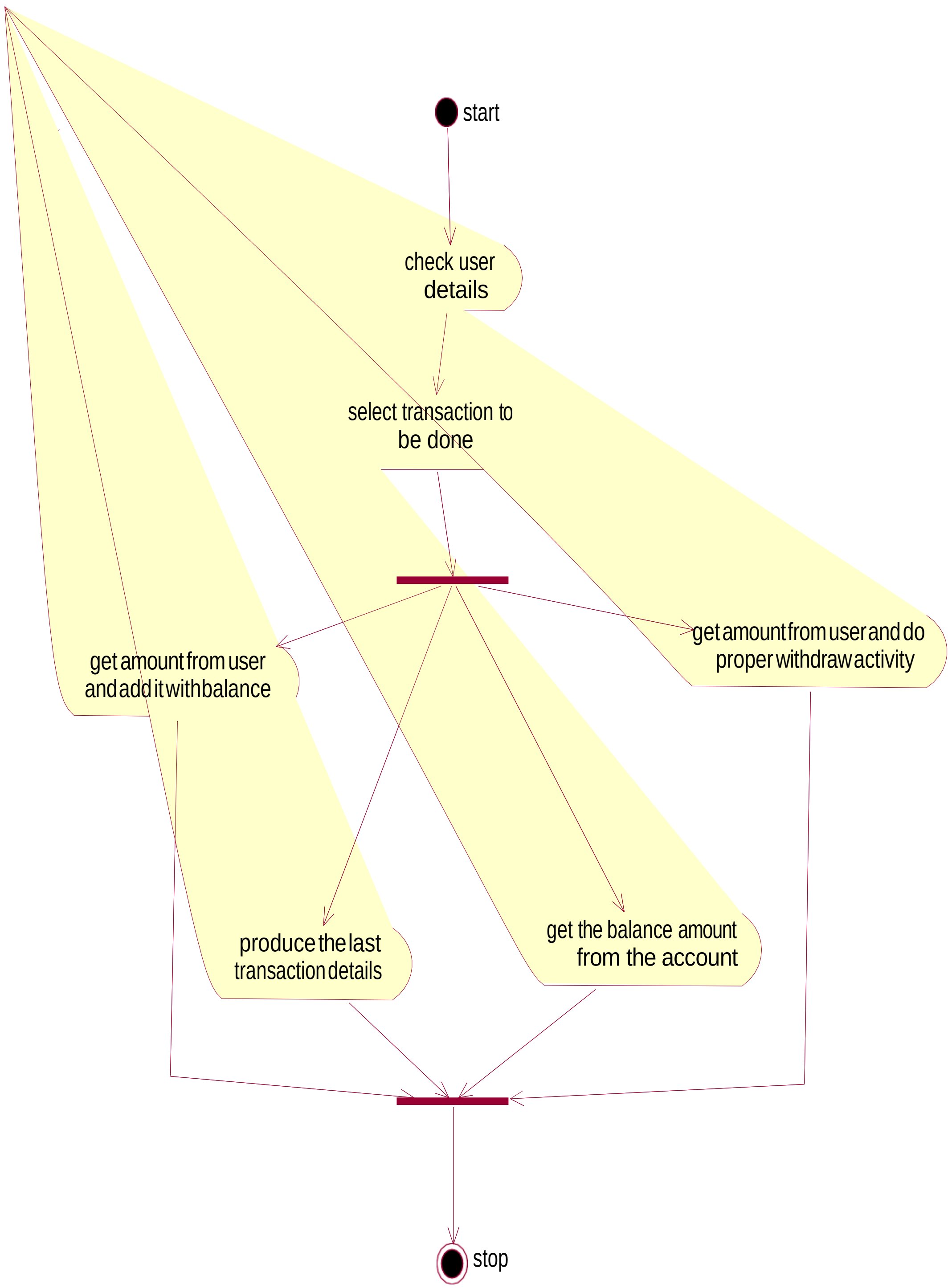
ACTIVITY DIAGRAM:

In this, the states are activities representing the performance of operations and the transactions are triggered by the completion of the operations.This diagram provides the view of flows and what is going on inside a usecase or among several classes.

The Activity diagram is used to describe the various activities taking place in an application. Here in our BANKING SYSTEM, we have various activities starting from login.

After login the client has to select the transactions to be done. If it is deposit, then add the given amount with the balance amount and update the database.

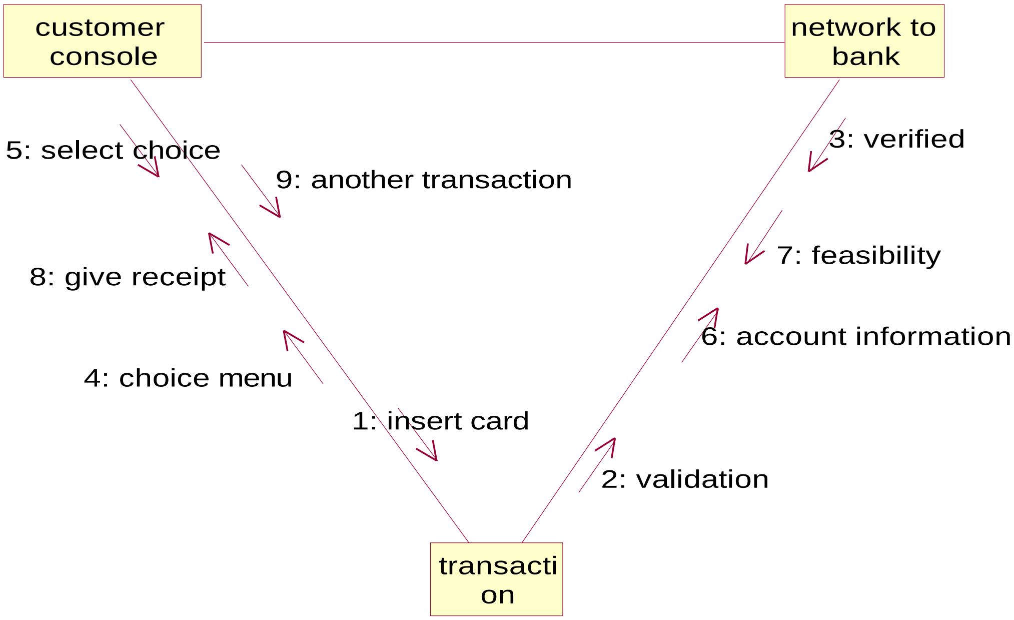
If it is withdraw, then decrement the given amount with the balance amount and update the database.


start
check user details
select transaction to be done
get amount from user and add it with balance
get amount from user and do proper withdraw activity
produce the last transaction details
get the balance amount from the account
stop

If it is balance checking, then get the balance amount from the database.

If it is ministatement generation, then obtain all the details from the database.

COLLABORATION DIAGRAM(ATM):



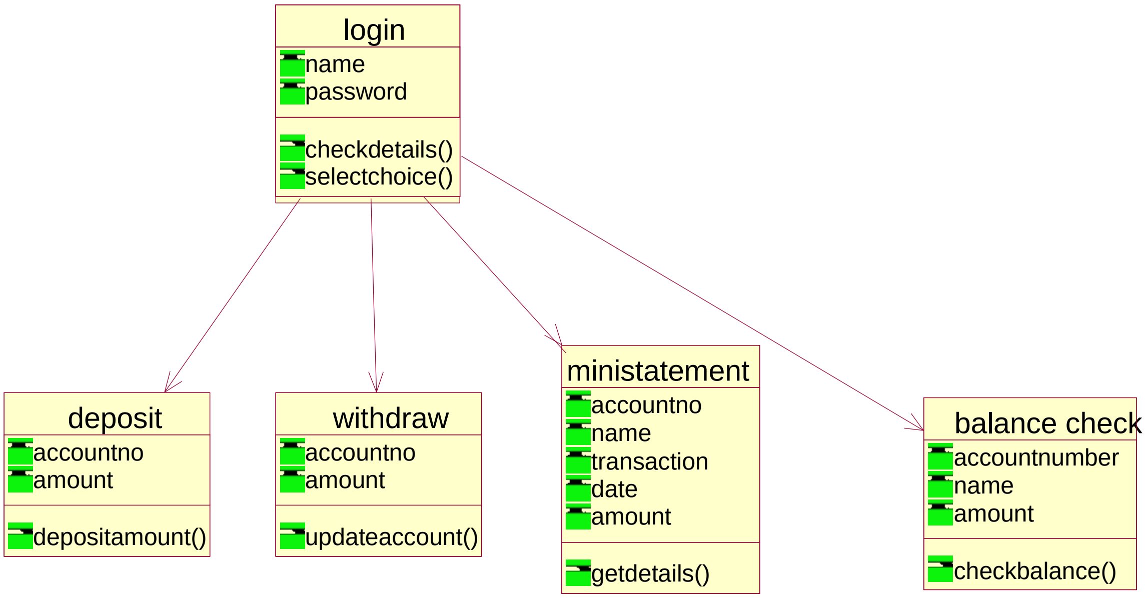
A collaboration diagram represents a collaboration, which is a set of objects related in a particular context, and interaction, which is a set of messages exchanged among the objects within he collaboration to achieve a desired outcome.

llaboration diagrams show exactly same information as the sequence diagram. However, collaboration diagram show this information in a different way and with different purpose.

In this collaboration diagram, the objects are represented as rectangle, the actors are stick figures. Whereas the sequence diagram illustrates the object and actor interaction overtime, the collaboration diagram shows the object and actor interaction without reference to time.

In our BANKING SYSTEM, each object interacts with each other or collaborates with each other; it gets represented by the solid line drawn between them.

CLASS DIAGRAM:



UML class diagram is the main static analysis diagram. This diagram shows the static structure of the model. This is a collection of static modeling elements, such as classes and their relationships, connected as a graph to each other and to their contents.

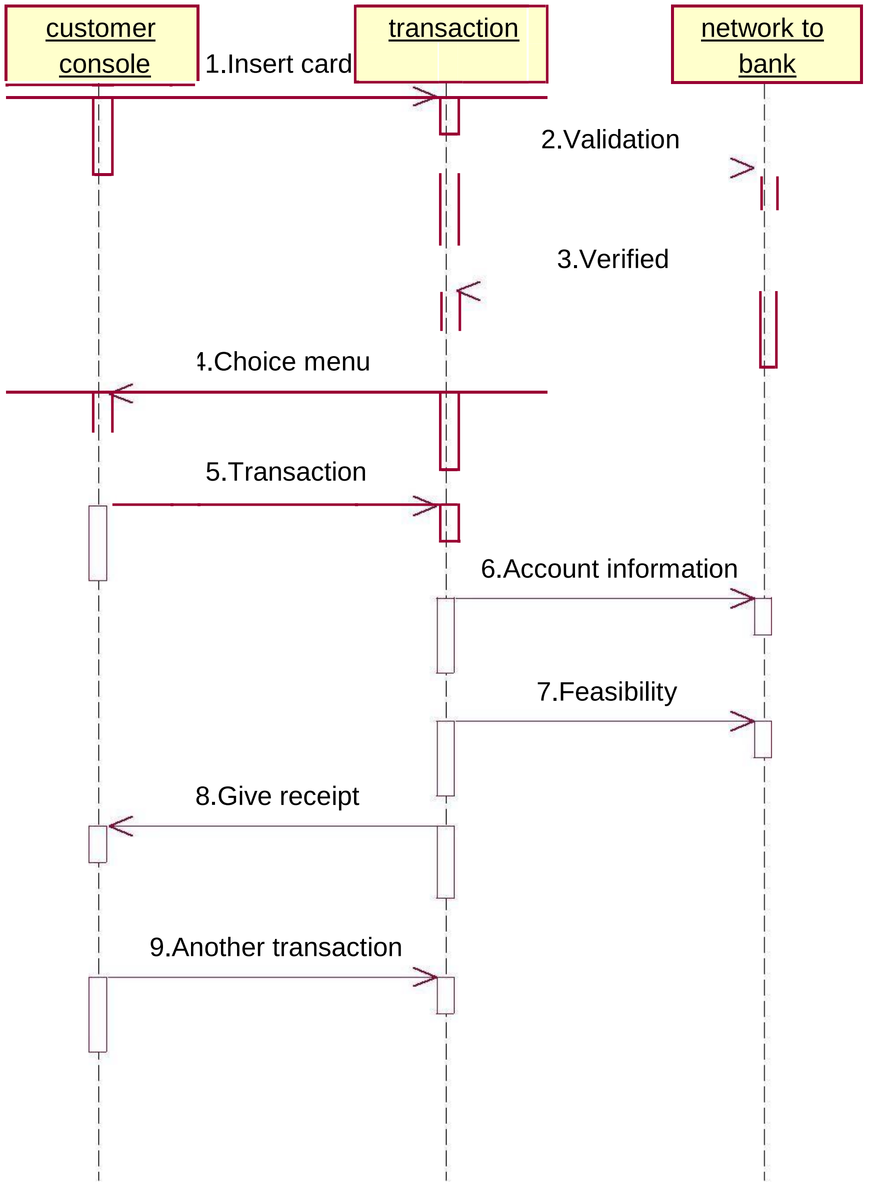
A class is drawn as a rectangle with three fields. The top field contains class name, the middle field contains attributes and the bottom field contains list of operations.

In our BANKING application, the classes are

1. New account
2. Deposit
3. Withdraw
4. Balance check
5. Transactions

Each class has its own attributes and properties

SEQUENCE DIAGRAM(ATM):



Sequence diagrams are easy and intuitive way of describing the behavior of a system by viewing the interaction between the system and its environment. A sequence diagram shows an interaction arranged in a time sequence.

The objects used in this sequence diagram are,
1.customer console
2. transaction
3. network to bank

The object client enters the login information. The object login access the details from the database and enters into the application. The object control obtains the amount for the operation either it is deposit or withdraw and update them according to the balance amount.

CODING:

Login Form:

```
Private Sub Command1_Click()
If Text1.Text = "rv" And Text2.Text = "mk" Then
   banking.Show
Else
MsgBox ("Invalid input")
End If
End Sub
```

Banking Form:

```
Private Sub Command1_Click()
newaccount.Show
End Sub
```

Account Form:

```
Private Sub crebtn_Click()
newaccountado.Recordset.Fields("Name") = txtname.Text
newaccountado.Recordset.Fields("Date of Birth") = txtdob.Text
newaccountado.Recordset.Fields("Gender") = txtgender.Text
newaccountado.Recordset.Fields("Address") = txtadd.Text
newaccountado.Recordset.Fields("Contact no") = txtphone.Text
newaccountado.Recordset.Update
MsgBox "User newaccount Successful"
withdraw.Show
End Sub

Private Sub Form_Load()
newaccountado.Recordset.AddNew
End Sub
```

Withdraw Form:

```
Private Sub Form_Load()
withdrawado.Recordset.AddNew
End Sub

Private Sub okbtn_Click()
withdrawado.Recordset.Fields("Accountno") = txtaccno.Text
withdrawado.Recordset.Fields("Withdraw amount") = txtwithdraw.Text
withdrawado.Recordset.Fields("Current Balance") = txtcurbal.Text
withdrawado.Recordset.Fields("Minimum Balance") = txtminbal.Text
withdrawado.Recordset.Fields("New Balance") = txtnewbal.Text
withdrawado.Recordset.Update
MsgBox "User withdraw Successful"
Deposit.Show
End Sub
```

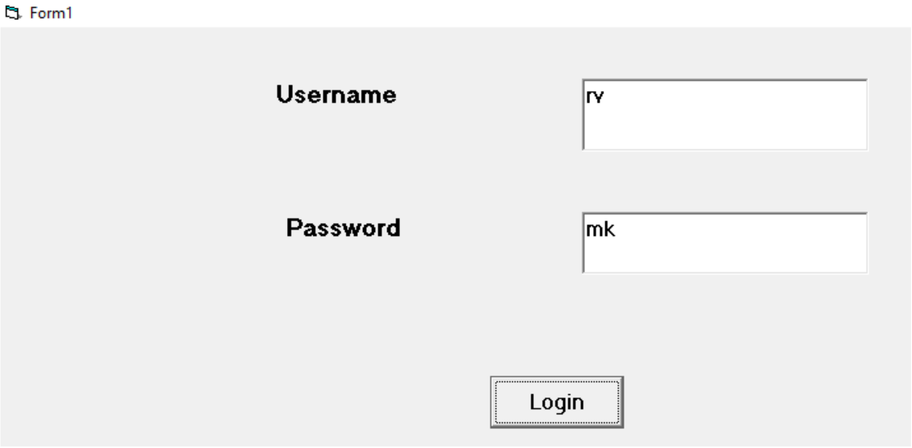
Deposit Form:

```
Private Sub addbtn_Click()
depositado.Recordset.Fields("Account no") = txtaccno.Text
depositado.Recordset.Fields("Current Balance") = txtcurbal.Text
depositado.Recordset.Fields("Deposit amount") = txtdepamt.Text
depositado.Recordset.Fields("New Balance") = txtnewbal.Text
depositado.Recordset.Update
MsgBox "User deposit Successful"
Transaction.Show
End Sub

Private Sub Form_Load()
depositado.Recordset.AddNew
End Sub
```

Transaction Form:

```
Private Sub Exitbtn_Click()
End
End Sub

Private Sub Form_Load()
transactionado.Recordset.AddNew
End Sub

Private Sub transactionbtn_Click()
transactionado.Recordset.Fields("Account no") = txtaccno.Text
transactionado.Recordset.Fields("Receiver Account no") = txtrecaccno.Text
transactionado.Recordset.Fields("Current Balance") = txtcurbal.Text
transactionado.Recordset.Fields("Minimum Balance") = txtminbal.Text
transactionado.Recordset.Fields("transaction Amount") = txttransamt.Text
transactionado.Recordset.Update
MsgBox "User transaction Successful"
End Sub
```

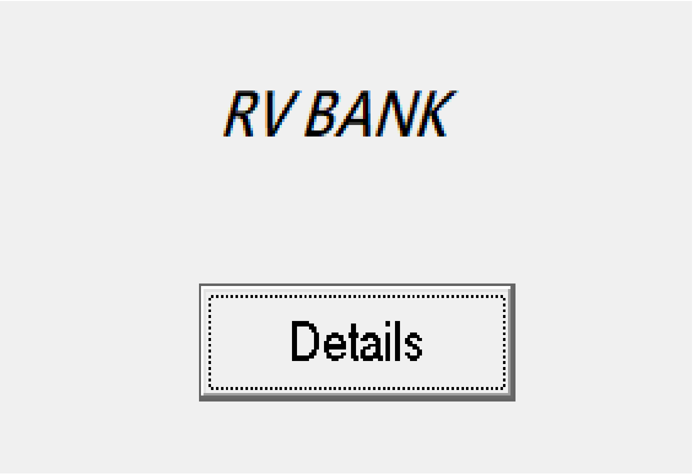
Output:
Form1:



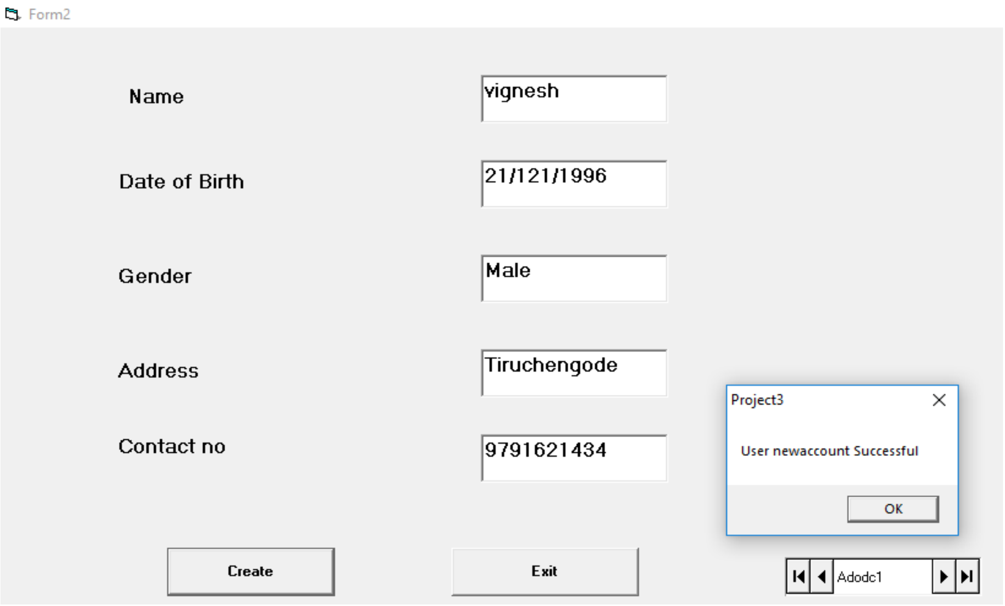
Username rv

Password mk

Form2:



Form3:



Name vignesh

Date of Birth 21/121/1996

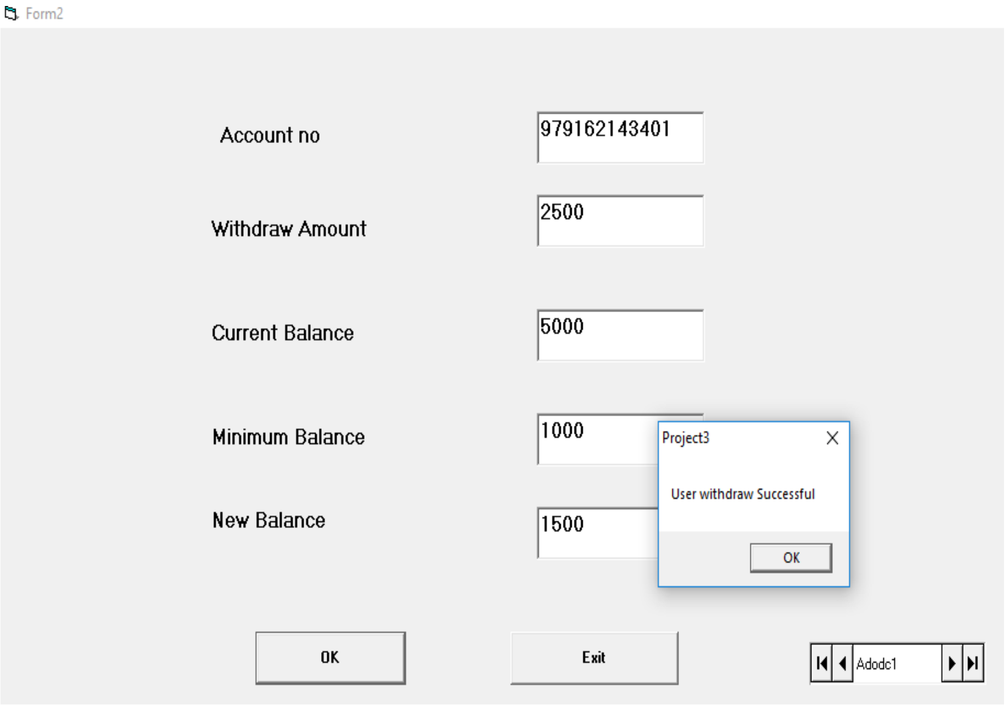
Gender Male

Address Tiruchengode

Contact no 9791621434

User newaccount Successful

Form4:



Account no 979162143401

Withdraw Amount 2500

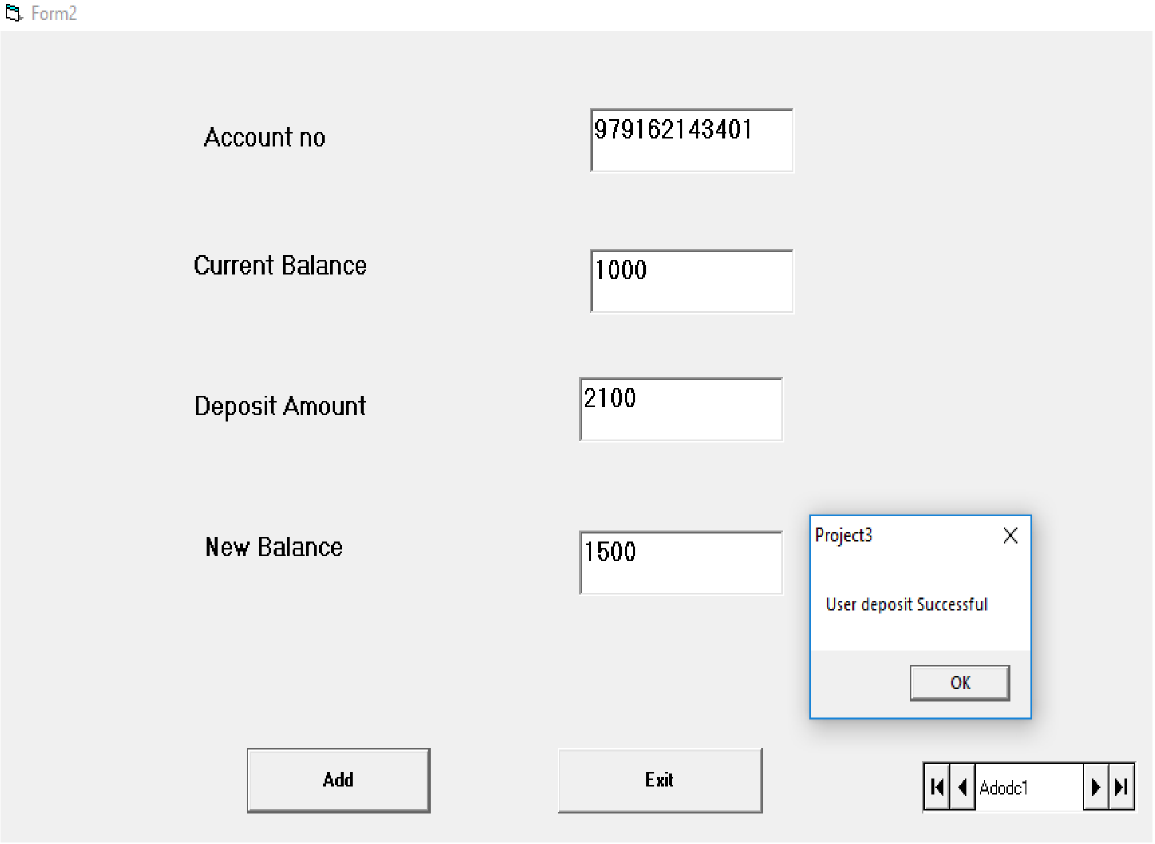
Current Balance 5000

Minimum Balance 1000

New Balance 1500

User withdraw Successful

Form5:



Form2

Account no 979162143401

Current Balance 1000

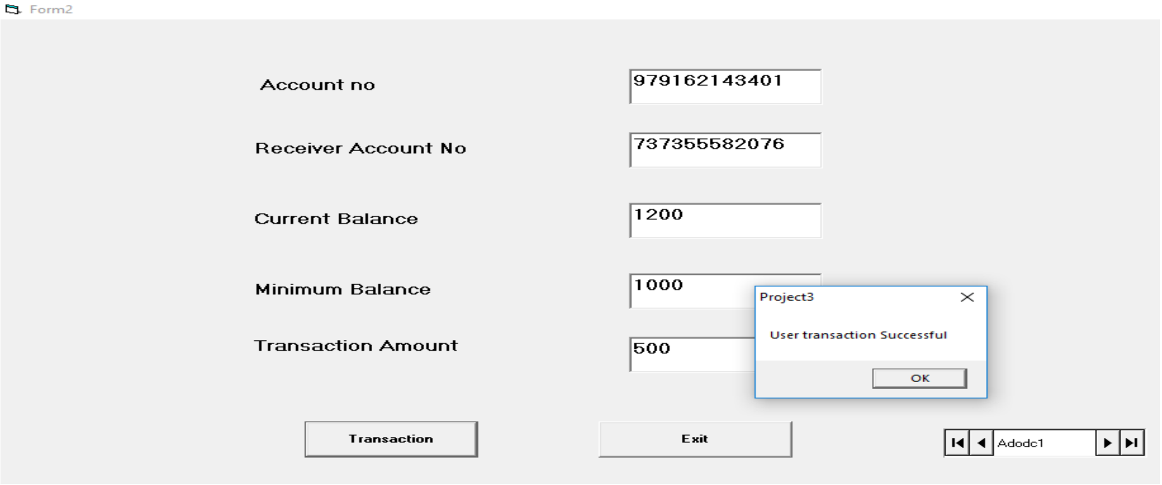
Deposit Amount 2100

New Balance 1500

Project3

User deposit Successful

Form6:



Form2

Account no 979162143401

Receiver Account No 737355582076

Current Balance 1200

Minimum Balance 1000

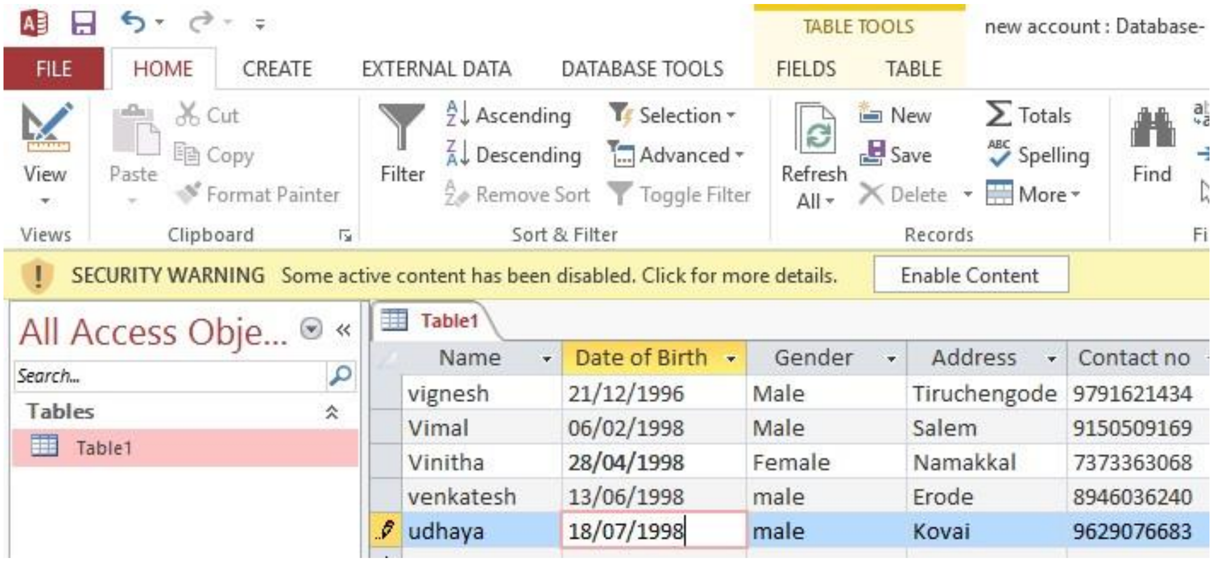
Transaction Amount 500

Project3

User transaction Successful

Database:

New Account Form



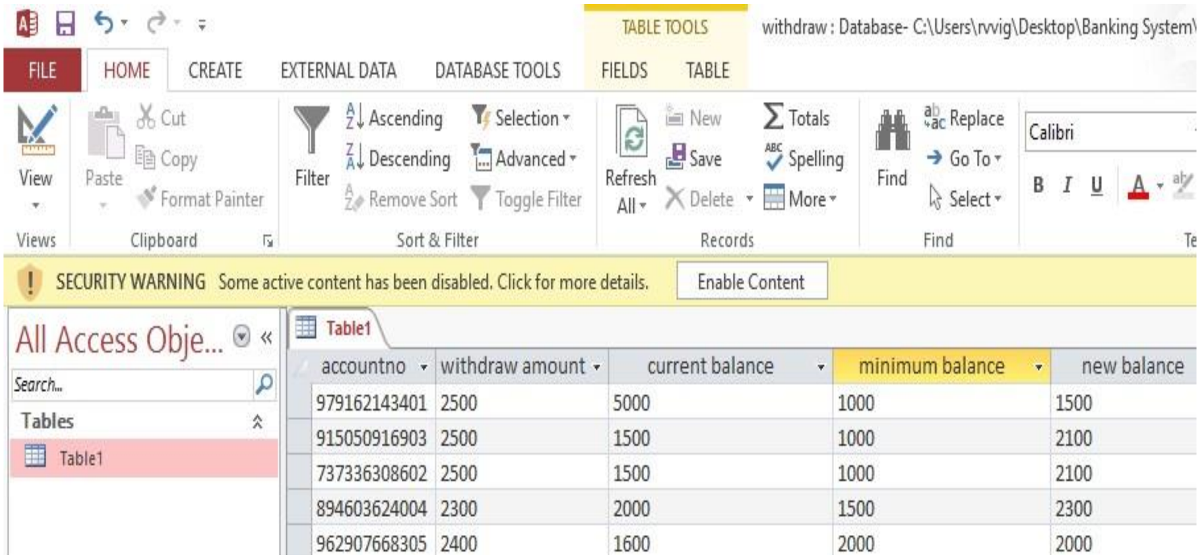
new account : Database-

SECURITY WARNING Some active content has been disabled. Click for more details. Enable Content

Table1

Name	Date of Birth	Gender	Address	Contact no
vignesh	21/12/1996	Male	Tiruchengode	9791621434
Vimal	06/02/1998	Male	Salem	9150509169
Vinitha	28/04/1998	Female	Namakkal	7373363068
venkatesh	13/06/1998	male	Erode	8946036240
udhaya	18/07/1998	male	Kovai	9629076683

WithDraw Form



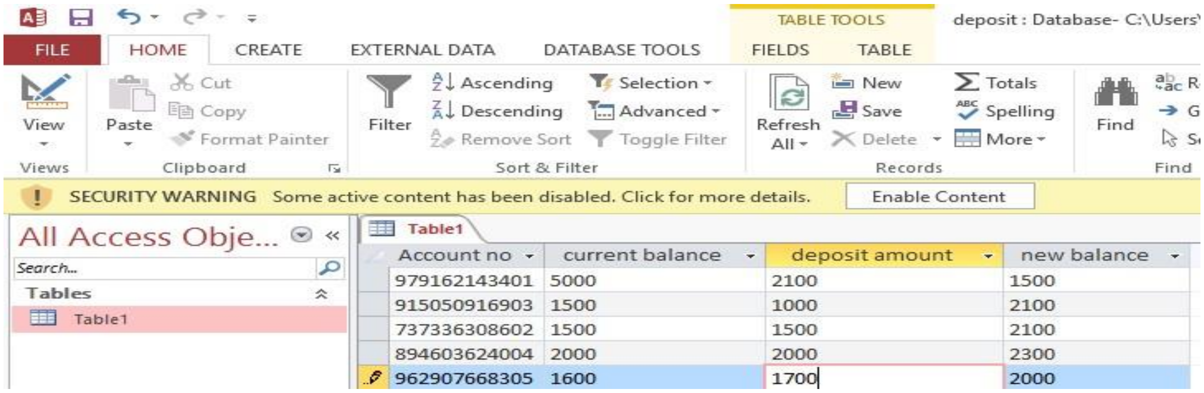
withdraw : Database- C:\Users\rvvig\Desktop\Banking System\

SECURITY WARNING Some active content has been disabled. Click for more details. Enable Content

Table1

accountno	withdraw amount	current balance	minimum balance	new balance
979162143401	2500	5000	1000	1500
915050916903	2500	1500	1000	2100
737336308602	2500	1500	1000	2100
894603624004	2300	2000	1500	2300
962907668305	2400	1600	2000	2000

Deposit Form

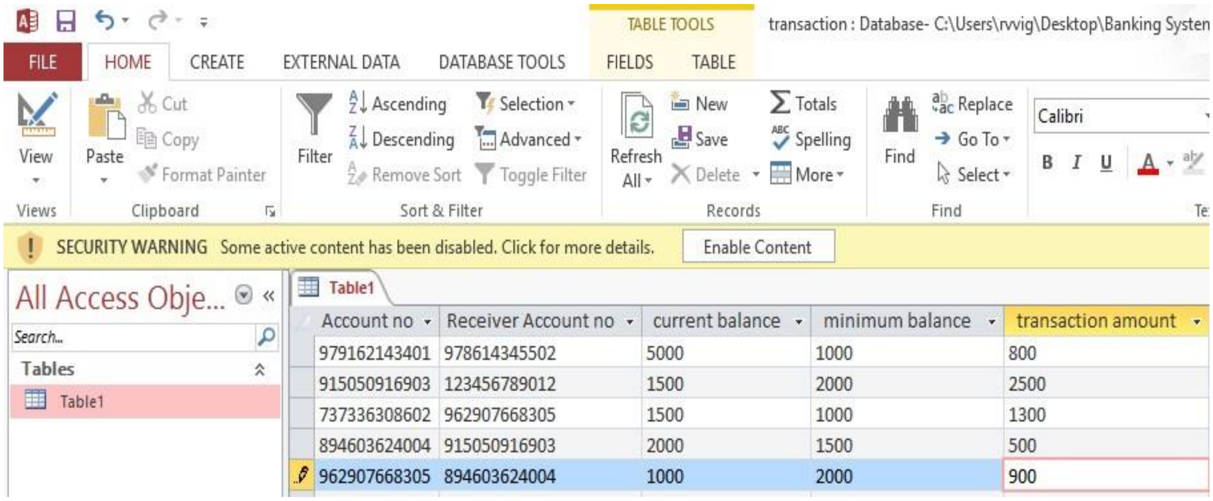


SECURITY WARNING Some active content has been disabled. Click for more details.

Table1

Account no	current balance	deposit amount	new balance
979162143401	5000	2100	1500
915050916903	1500	1000	2100
737336308602	1500	1500	2100
894603624004	2000	2000	2300
962907668305	1600	1700	2000

Transaction Form



SECURITY WARNING Some active content has been disabled. Click for more details.

Table1

Account no	Receiver Account no	current balance	minimum balance	transaction amount
979162143401	978614345502	5000	1000	800
915050916903	123456789012	1500	2000	2500
737336308602	962907668305	1500	1000	1300
894603624004	915050916903	2000	1500	500
962907668305	894603624004	1000	2000	900

<u>Result:</u>

Thus the **BANKING SYSTEM** is developed with all necessary documents and UML diagrams using Software Engineering methodology.

EX.NO:4
DATE :

TEXT EDITOR

AIM

To prepare necessary documents and to develop the “Text Editor” with usecase diagrams using software engineering methodology.

PROBLEM STATEMENT

The project Text Editor is to develop an application for allows the user to type the text with different font styles, type, size and vibrant colors. It contains facility to save the typed text as document and to open the existing document for reading and writing.

SOFTWARE REQUIREMENT ANALYSIS

1. Editor Form
2. Font Style Frame
3. Save Frame
4. Open Frame

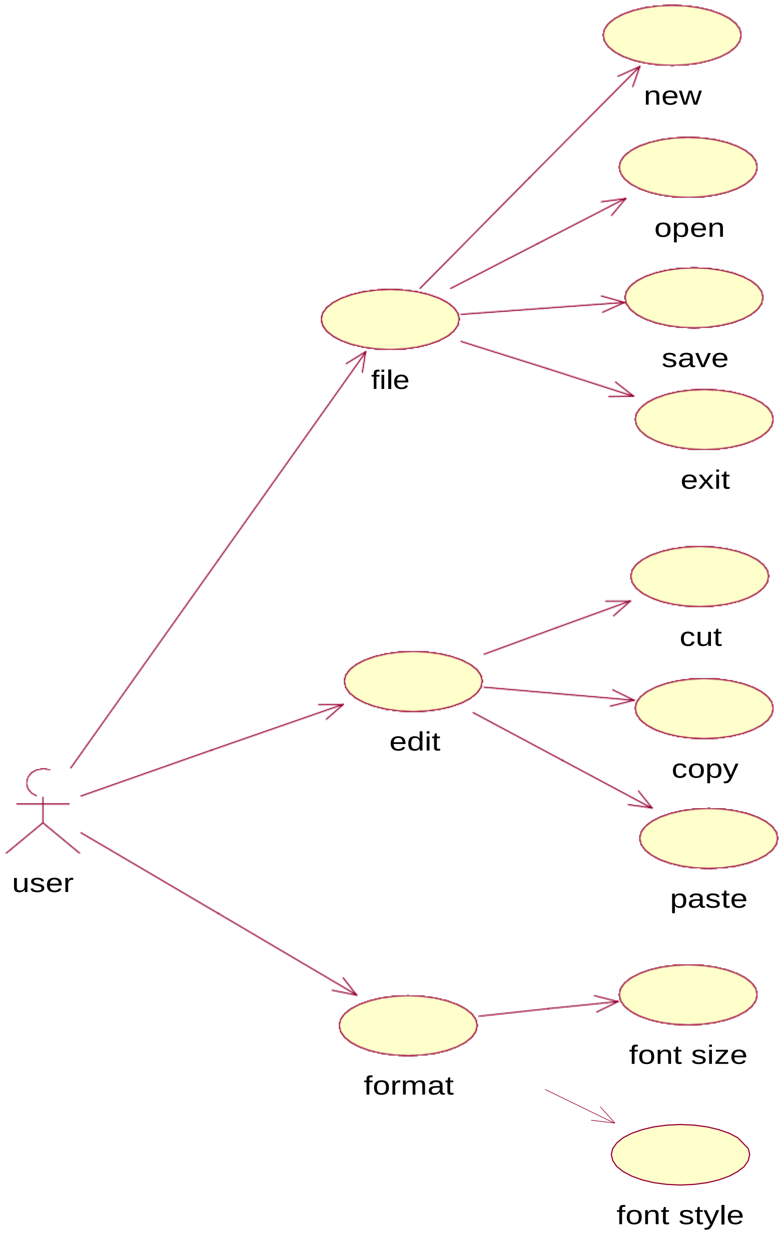
The Editor form contains the components that allow the user to type the text and the menu bar that provides various options.

The Font style, Save and Open frames are the members of the Editor form.

The Font Style frame allows the user to enrich their document with different font color, style, type and size of the text.

The Save frame allows the user to save their typed text as document and the Open frame allows the user to open the existing document for reading and writing.

USE CASE DIAGRAM



Use case Diagram:

Use case diagram is a graph of actors, set of use cases enclosed by a system boundary, communication (participation) association between the actors and the use cases and a generalization among the use cases.

Use case:

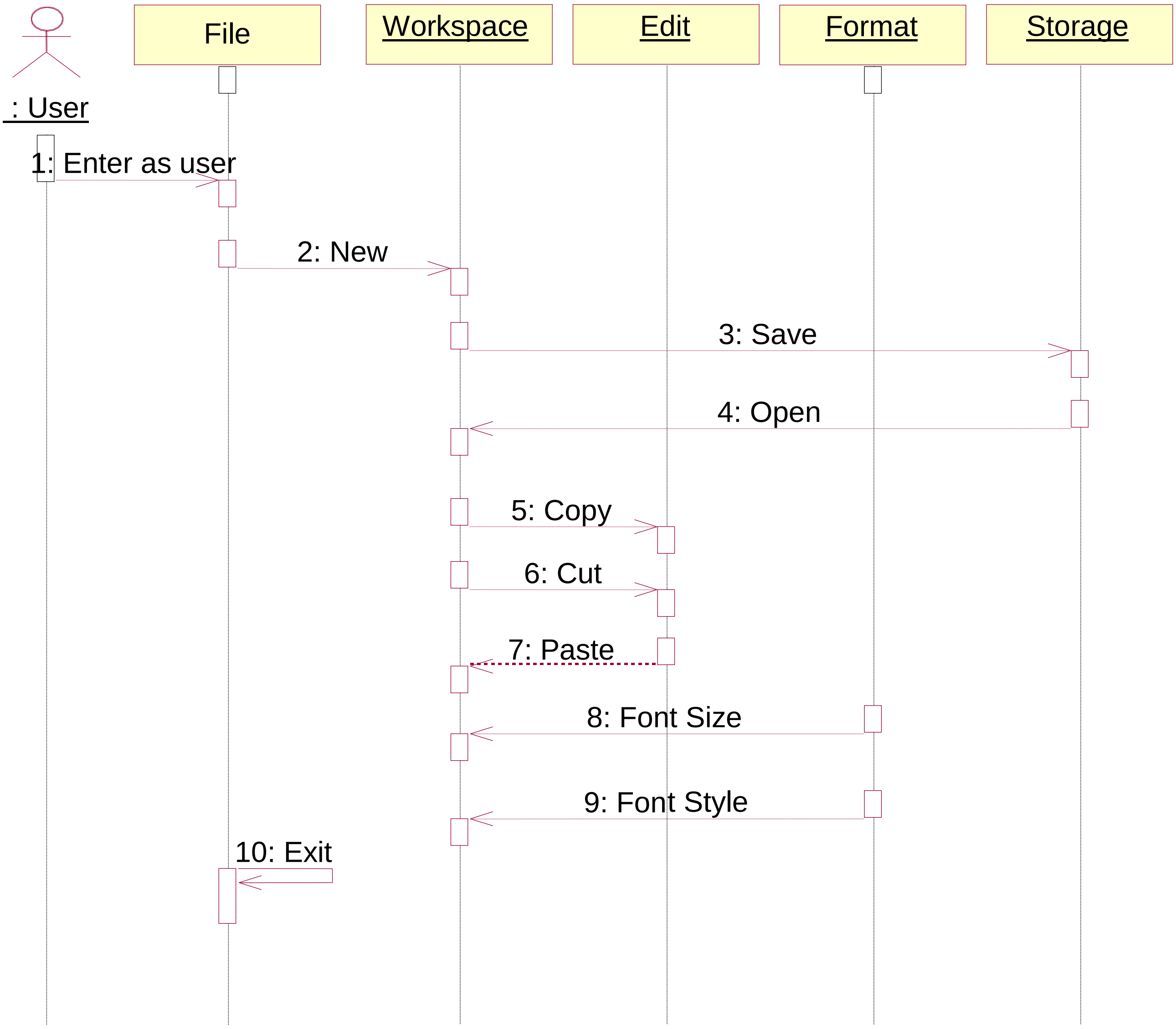
A use case is a description of a set of sequence of actions that a system performs to yield result of value to an actor.

The Use Cases described are,

1. New File
2. Type Text
3. Format Text
4. Change Style
5. Open, Save File

The File use case is to describe that, the user should Open and Save the files or create a new File.

SEQUENCE DIAGRAM:



Sequence diagrams are easy and intuitive way of describing the behavior of a system by viewing the interaction between the system and its environment.A sequence diagram shows an interaction arranged in a time sequence.

The objects used in this sequence diagram are,

1. File
2. Workspace
3. Edit
4. Format
5. Storage

BASIC FLOW:

The user opens the application and application allows the user to type the text.

ALTERNATE FLOW:

The object File makes the user to select new file or to open the existing file and to save the file.

The object Workspace allows the users to type text, from the workspace the users access all other object.

The object Edit used to cut, copy and paste the selected text.

The object Format allows the user to change the font style, color and its type.

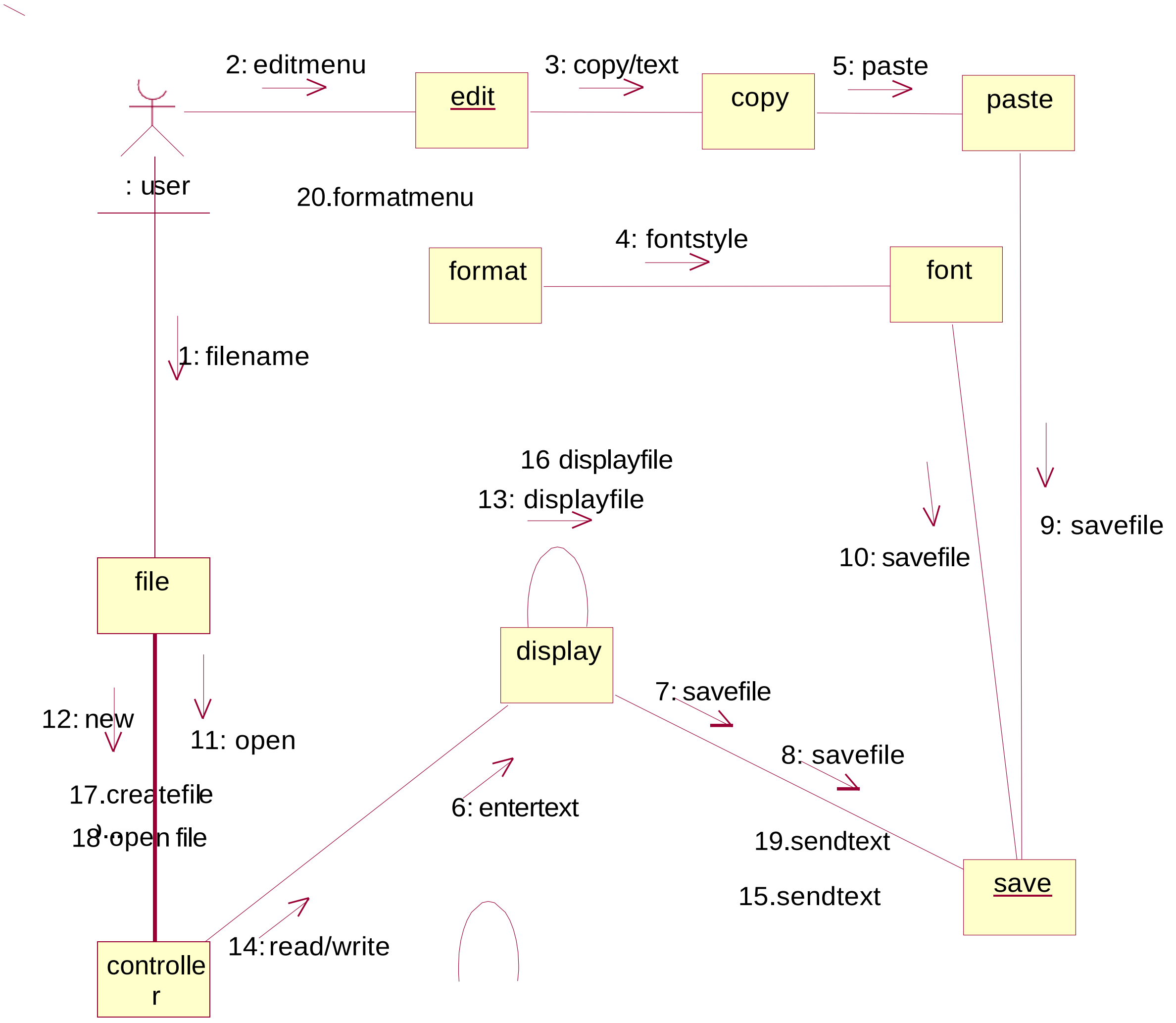
The object Storage is to store the typed text in the disk as file.

COLLOBORATION DIAGRAM

A collaboration diagram represents a collaboration, which is a set of objects related in a particular context, and interaction, which is a set of messages exchanged among the objects within the collaboration to achieve a desired outcome.

Collaboration diagram shows exactly the same information as the sequence diagram. However, collaboration diagram shows this information in a different way and with different purpose.

In this collaboration diagram, the objects are represented as rectangle, the actors are stick figures. Whereas the sequence diagram illustrates the object and actor interaction overtime, the collaboration diagram shows the object and actor interaction without reference to time.



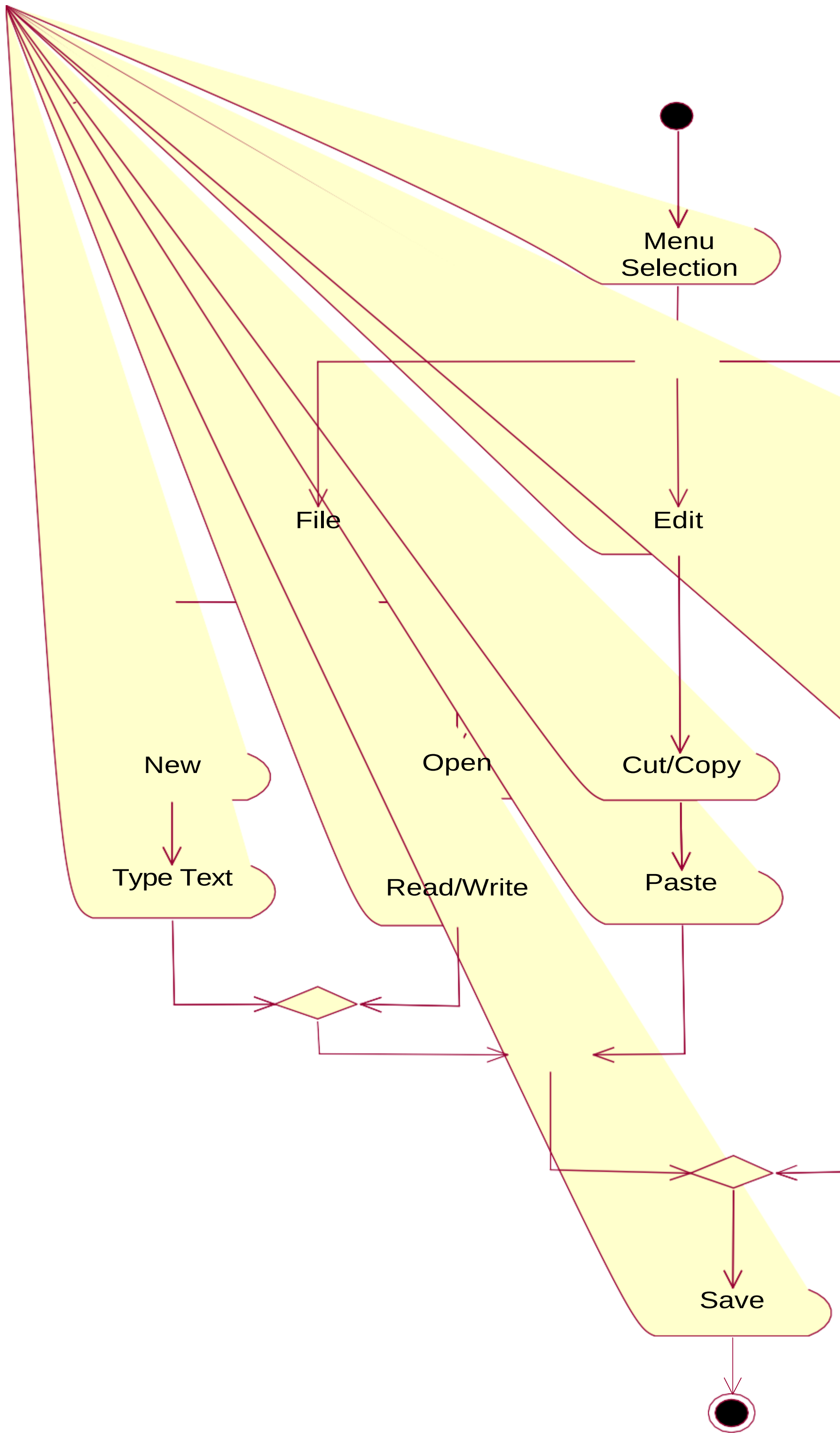
In our TEXT EDITOR each object interacts with each other or collaborates with each other; it gets represented by the solid line drawn between them.

ACTIVITY DIAGRAM

The activity diagram describes the sequencing of activities with support for both conditional and parallel behavior.

The Activity diagram is used to describe the various activities taking place in an application. Here in our TEXT EDITOR, we have various activities starting from File.

After user opens the application he/she choose file menu to create new file or to open the existing file then user read/write. User chooses the Edit menu to cut, copy and paste the selected text.

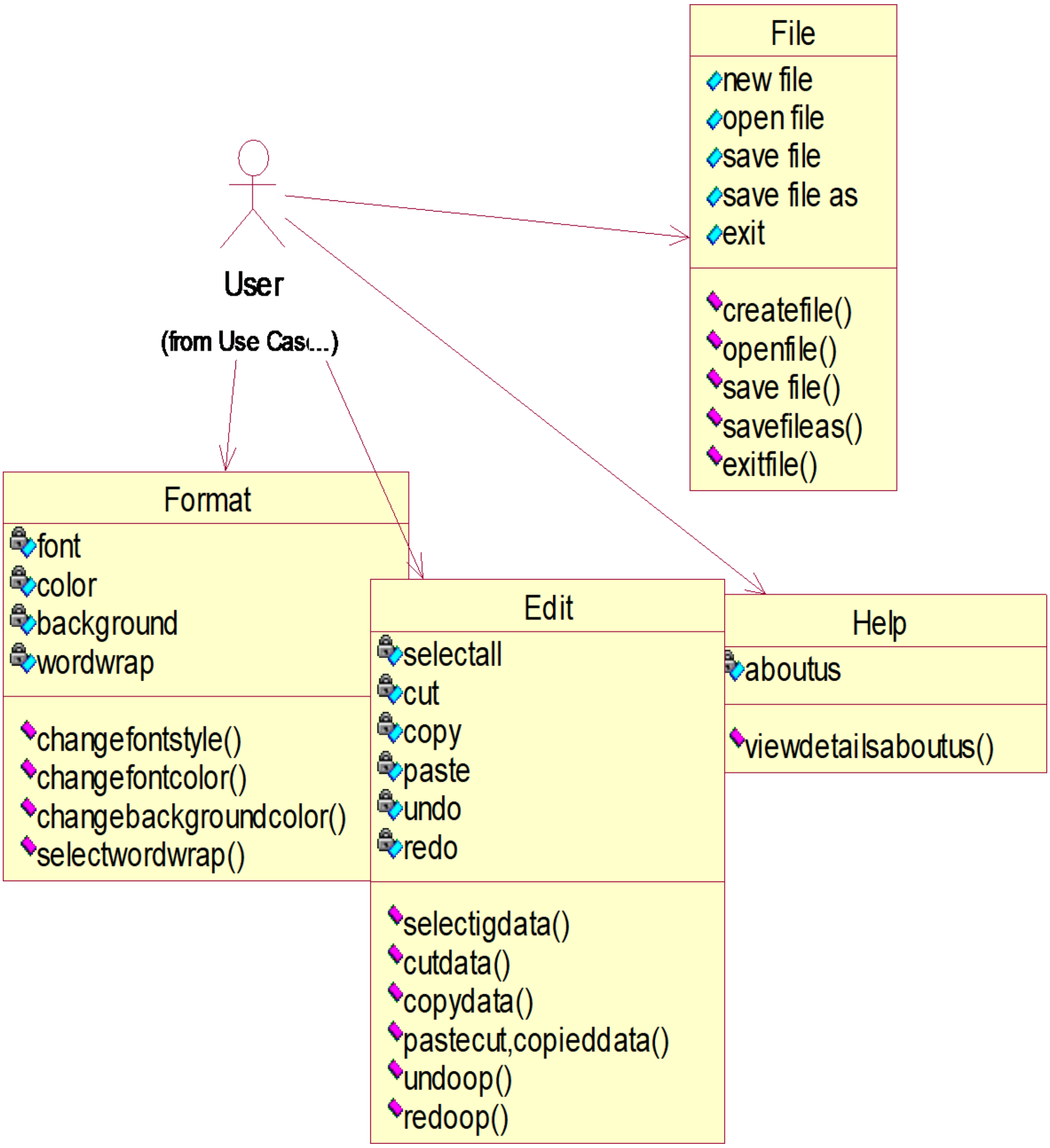


By choosing Format the user can change the font style, color and its type.
The object Storage is to store the typed text in the disk as file.

CLASS DIAGRAM:

Class diagrams show the interactions between classes in the system. Class diagram also shows the attributes and operation of a class and the constraints that apply to the way objects are connected.

Classes contain information and behavior that acts on that information. Each class on class diagram is represented by rectangle divided into three sections. The first section shows the class name, second section shows the attributes the class contains and last section contains the operation of the class.



SOFTWARE DEVELOPMENT

Fore color:

```
Private Sub mnuTxtColor_Click()
cd.ShowColor
FrmChild.TextBox.ForeColor = cd.Color
End Sub
```

Background Color:

```
Private Sub mnuBackColor_Click()
cd.ShowColor
FrmChild.TextBox.BackColor = cd.Color
End Sub
```

Fonts:

```
Private Sub mnuFont_Click()
cd.Flags = cd1CFBoth
cd.ShowFont
FrmChild.TextBox.Font = cd.FontName
FrmChild.TextBox.FontItalic = cd.FontItalic
FrmChild.TextBox.FontBold = cd.FontBold
FrmChild.TextBox.FontSize = cd.FontSize
End Sub
```

Exit:

```
Private Sub mnuExit_Click()
End
End Sub
```

SOFTWARE TESTING:

TEST CASES

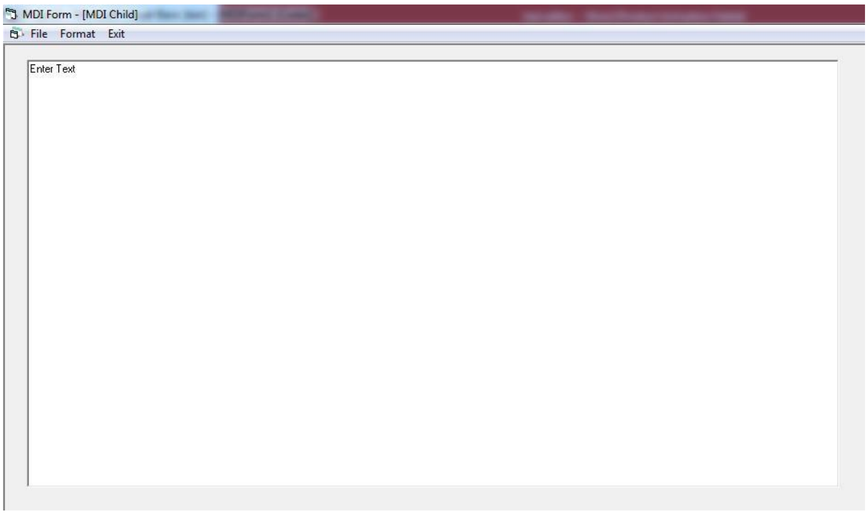
The sample text is typed and its font style, font color and font type are changed.

The new file is created and the typed text is stored in disk as file. The existing file is opened and it allows to read or write the file content.

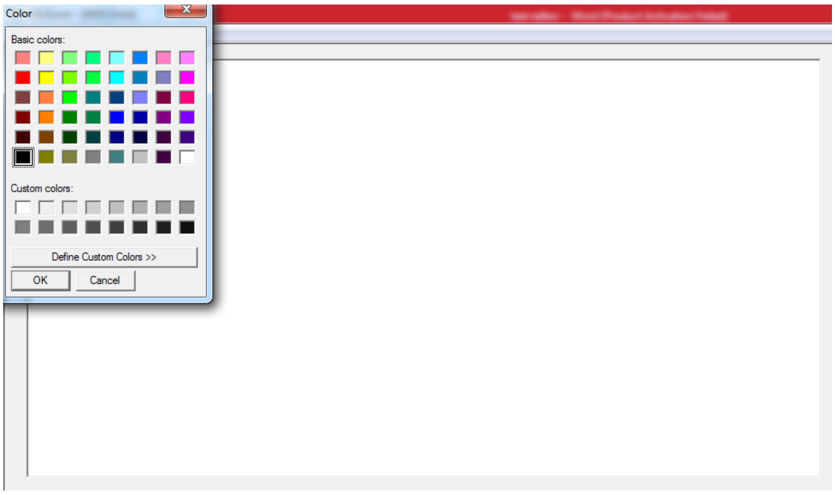
The developed application gets tested with the sample inputs.

OUTPUT:

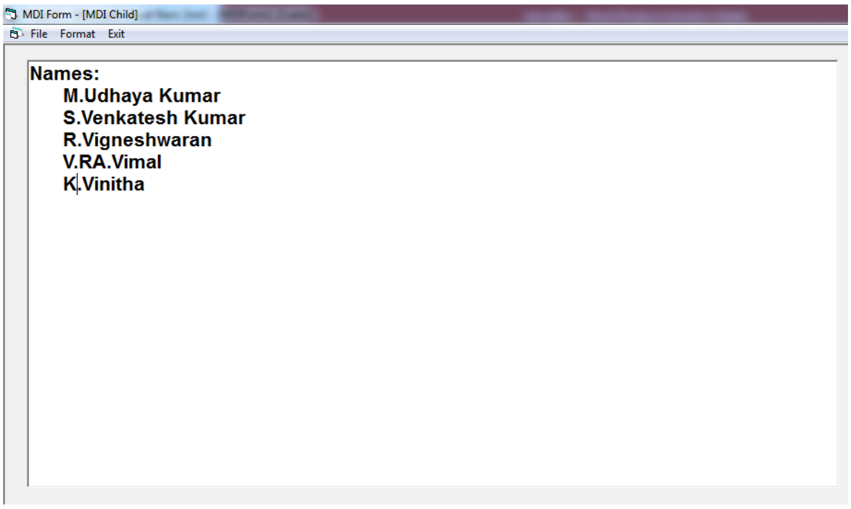
Form1:



Form2:



Form3:



Names:
M.Udhaya Kumar
S.Venkatesh Kumar
R.Vigneshwaran
V.RA.Vimal
K.Vinitha

RESULT:

Thus the Text Editor is developed with all necessary documents and usecase diagrams using software engineering methodology.

EX. NO:5
DATE :

ONLINE VOTING SYSTEM

AIM:

To prepare necessary documents and to develop the project **Course Registration** S**ystem** with UML diagrams using Software Engineering Methodology.

PROGARM ANALYSIS AND PROJECT PLANNING

This project **COURSE REGISTRATION SYSTEM** is to develop an application to make the student registration in computer centers. First the students select their needed course from the available courses. Then submit their details with their qualification. If there is wanted course available we allocate a seat and reduce one from that. These functions are done in student login. In administration login the authorized person can do the operations of the delete, update and search.

SOFTWARE REQUIREMENT ANALYSIS

The Modules in the Project:

1. Login student 1.1
   course detail 1.2
   Student details
2. Login Administration
   2.1 view
   2.2 update
   2.3 exit

The student login provides the options are course details.

In this login a student can choose the given course and fill up the student details for registration such as

1. Name
2. Gender
3. Place
4. Date of birth
5. Qualification

In the administration login ,the management uses it for the purpose of making an admission etc. They are

1. Amount
2. Payment details
3. Student details
4. Name of the bank

Some function in the administration can be given below:

1. View -helps in viewing the student details.
2. Update -helps in updating the latest function.admission.
3. Exit -helps in the normal exit

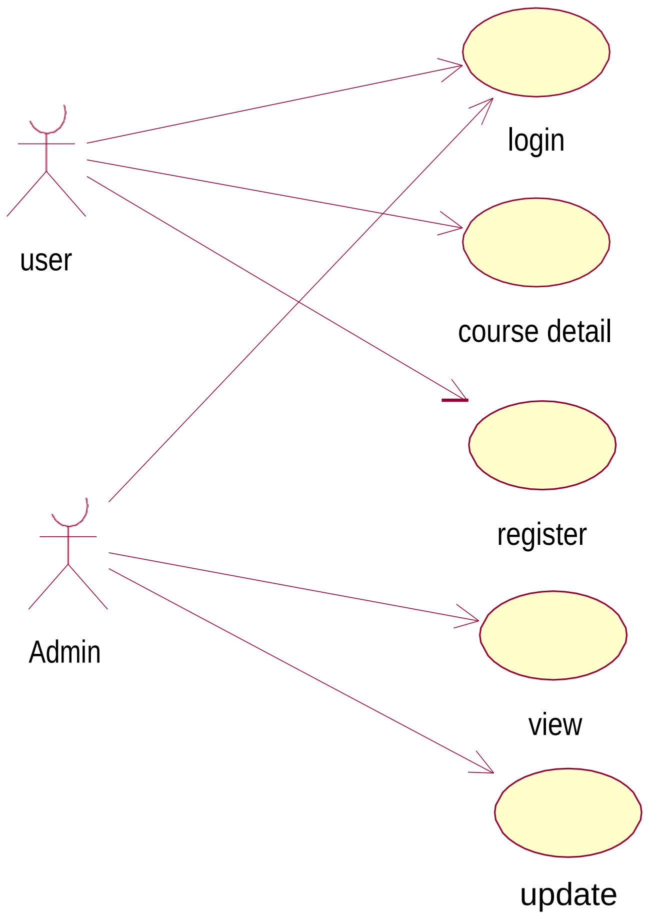
SYSTEM SPECIFICATION SOFTWARE REQUIRMENTS:
DATA DICTIONARY:

Field Name	Description	Data Type	Field Size	Default Value	Validation
Name	Name of the Student	string	20	Null	(A-Z)or(a-z)
Gender	Gender of the student	string	10	Null	(A-Z)or(a-z)
Place	Address of the student	string	20	Null	(A-Z)or -
Dob	Date of birth of a student	integer	03	Null	-
qualification	Course studied by the student	string	50	Null	-

HARDWARE REQUIRMENTS:

- Operating system:windows xp
- Front end:Microsoft Visual Studio 2005
- Back end:MS-Acess
- Processor:Intel pentium IV
- RAM:512MB
- Harddisk:80GB
- Monitor:17”LCD
- Keyboard:multimedia keyboard
- Mouse: USB optical mouse

USE CASE DIAGRAM:



Use Case Diagram:

Use case diagram is a graph of actors, set of use cases enclosed by a system boundary, communication (participation) association between the actors and the cases and a generalization among the use cases.

Actors:

An actor represents a set of roles that user of a use case play when interacting with the use cases. Actor identified here is Staff, Student and Administrator.

Use case:

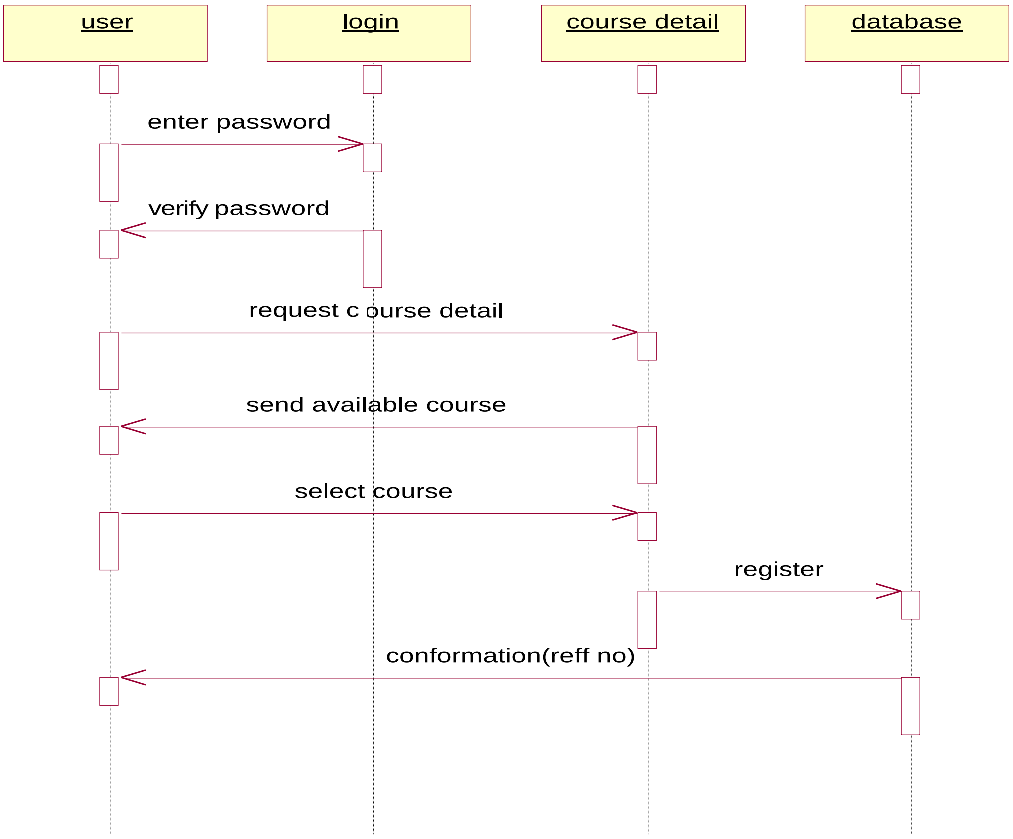
A use case is a description of a set of sequence of action that a system performs to yield result of value to an actor.

The Use Cases described are

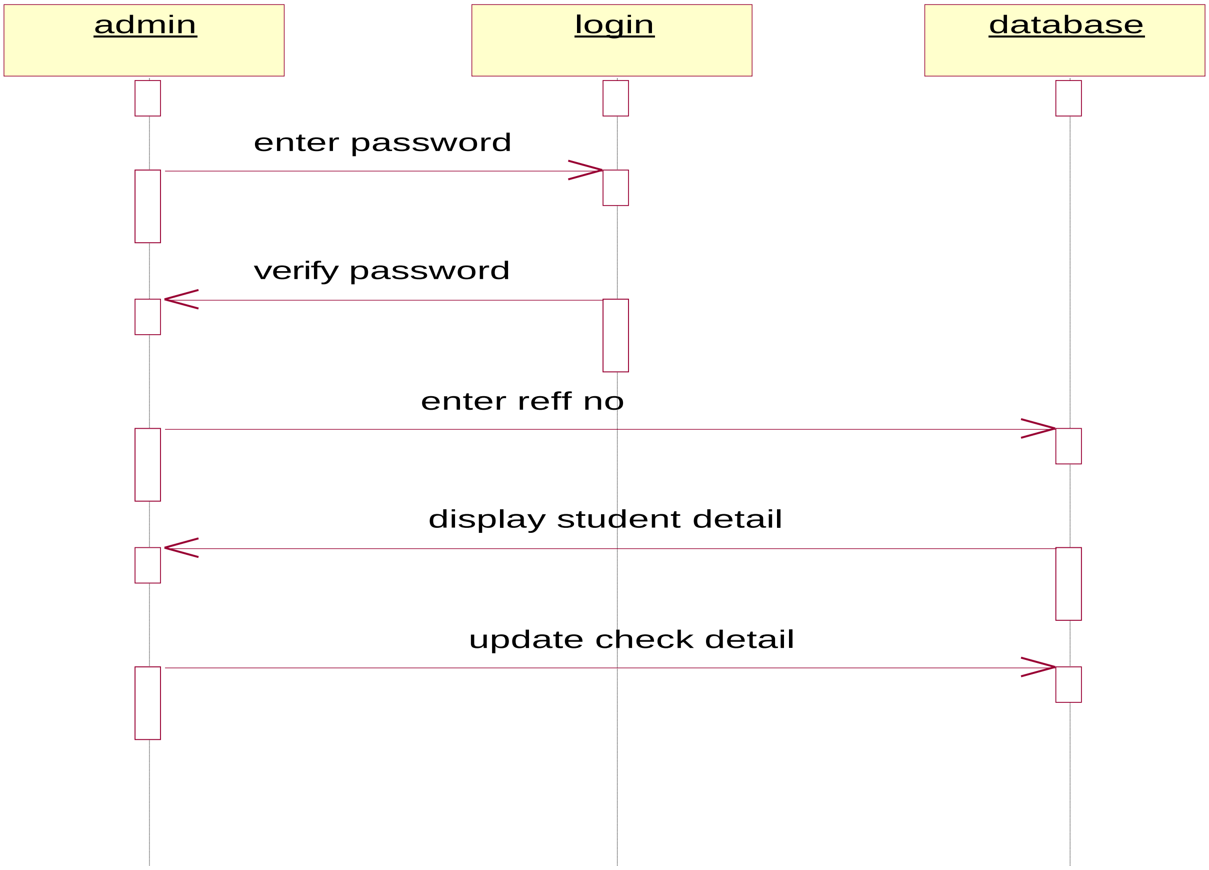
1. Login
2. Course Details
3. Register
4. View
5. Update

1. The Login use case is to describe that, the user should choose his/her category whether he/she is user or administrator.
2. The use case display the course details and the offered with their instruction and also about the payment.
3. The use case collects the information about the user such as name, address, contact no, E=mail id, etc, and stored the details in the data base
4. The View Details use case describes that the administrator views the details about the user and update the payment details in the registration form.
5. The update details are confirmation details of user requisition to the course registration, and he is allowed to attain the course.

SEQUENCE DIAGRAM FOR USER:



SEQUENCE DIAGRAM FOR ADMIN:



Sequence diagrams are easy and intuitive way of describing the behavior of a system by viewing the interaction between the system and its environment. A sequence diagram shows an interaction arranged in a time sequence. Here we use two sequence diagrams one is for administrator and another is for user.

The objects used in this sequence diagram of user are,

1. Student
2. Login
3. Course Detail
4. Register

The student is the course selector. They first select their course and fill the registration form and then submit. After submitting the administrator modifies and updates the registration.

The objects used in this sequence diagram of administrator are,

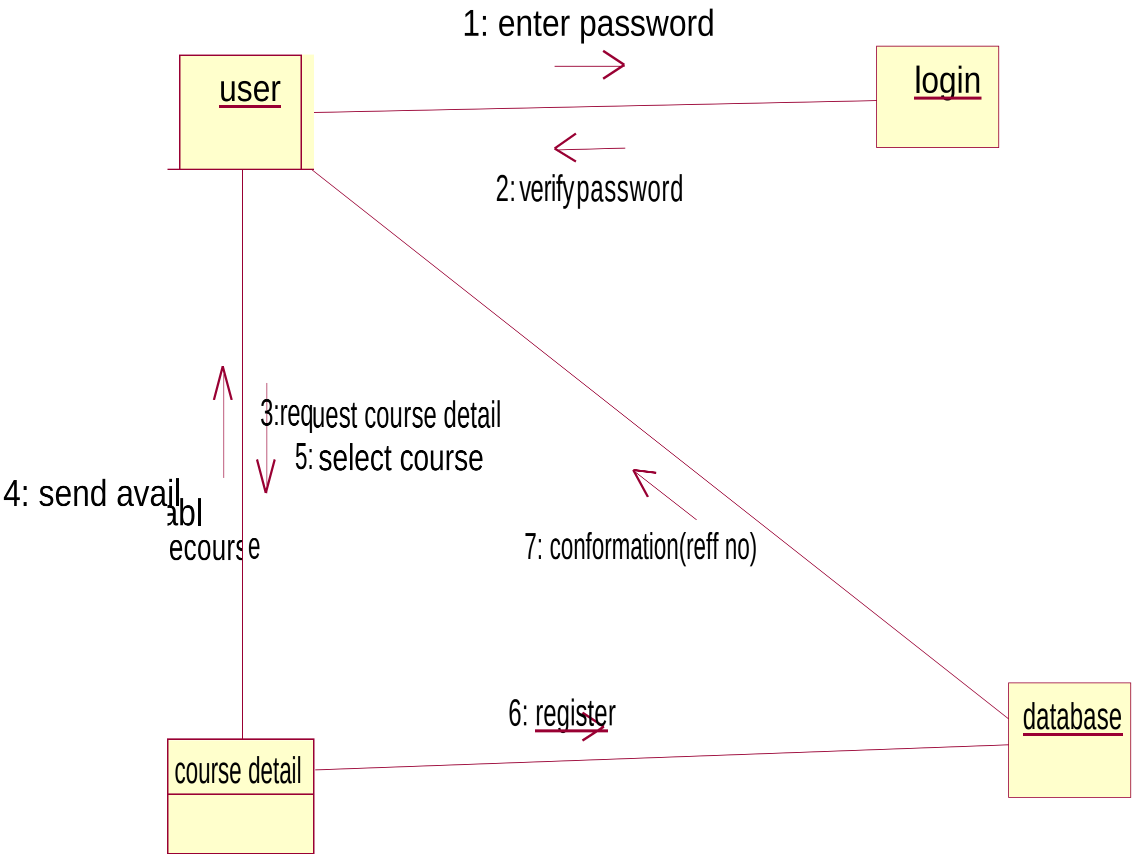
a. Admin
b. Login
c. View
d. Update

BASIC FLOW:

The user has to login and fill up the registration form for their course need and submit it. For each registration a reference number is given. The administrator views the details with that reference number and he updates the details with additional information.

COLLABORATION DAIGRAM:

COLLABRATION DIAGRAM FOR USER



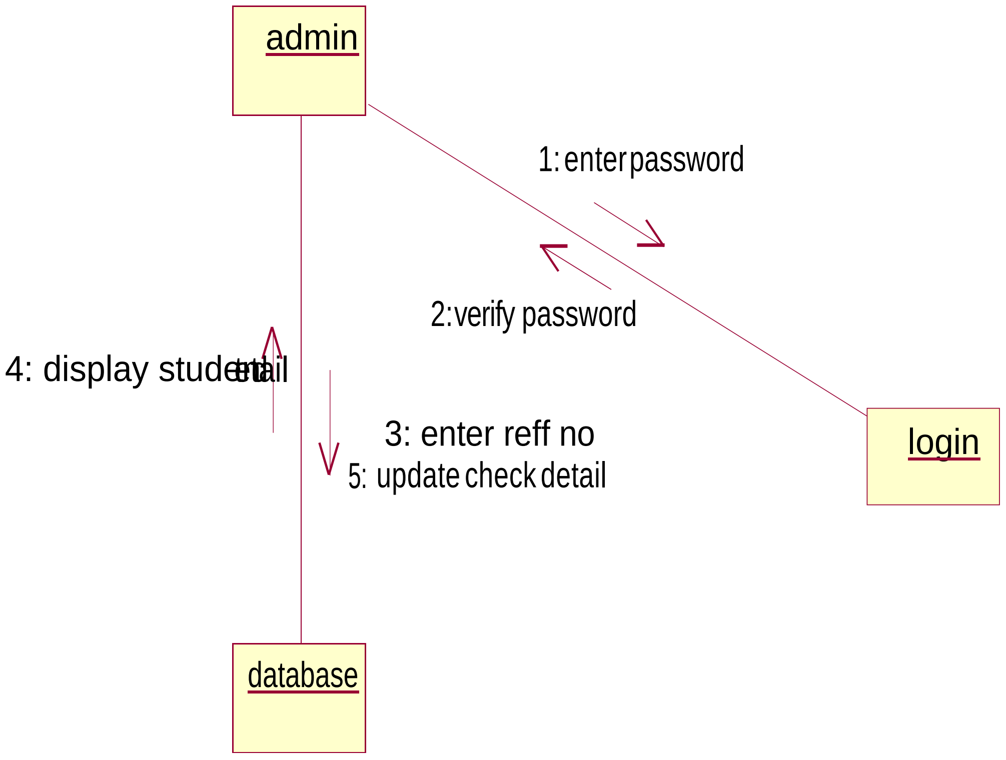
A collaboration diagram represents a collaboration, which is a set of objects related in a particular context, and interaction, which is a set of messages exchanged among the objects within the collaboration to achieve a desired outcome.

Collaboration diagram shows exactly the same information as the sequence diagram. However, collaboration diagram shows this information in a different way and with different purpose.

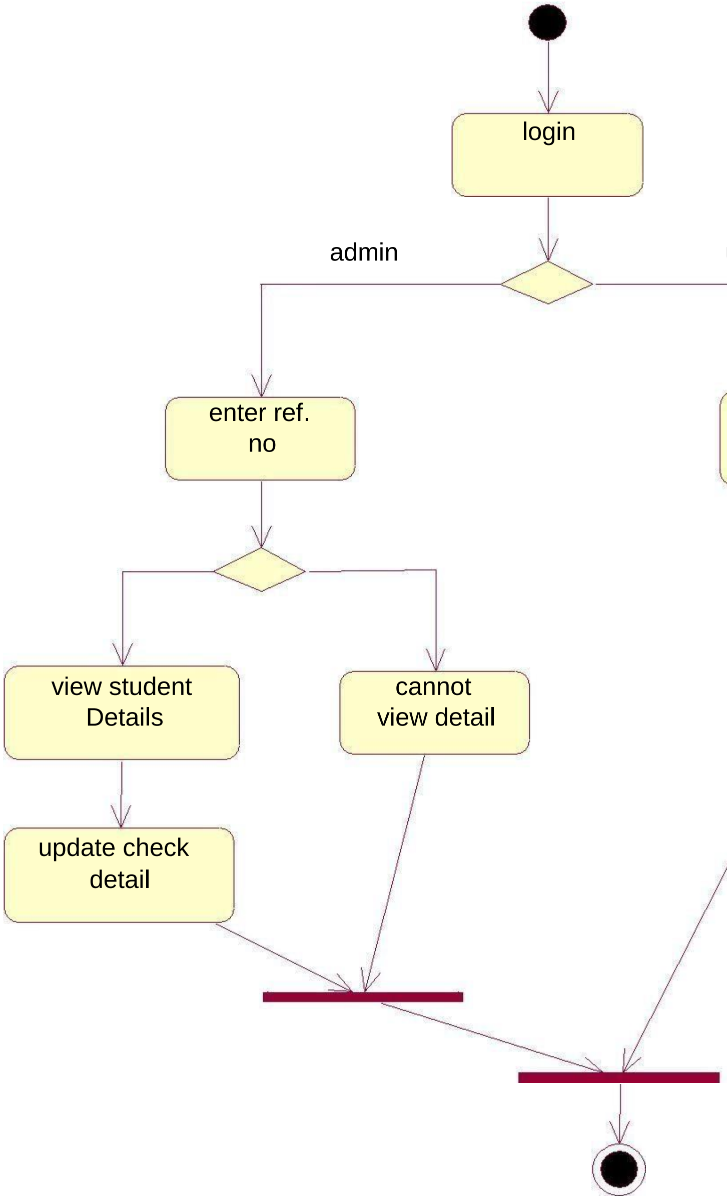
In this collaboration diagram, the objects are represented as rectangle, the actors are stick figures, whereas the sequence diagram illustrates the object and actor interaction overtime, the collaboration diagram shows the object and actor interaction without reference to time.

In our course registration, each object interacts with each other or collaborates with each other; it gets represented by solid line drawn between them.

COLLABRATION DIAGRAM FOR ADMINSTRATOR



ACTIVITY DIAGRAM:

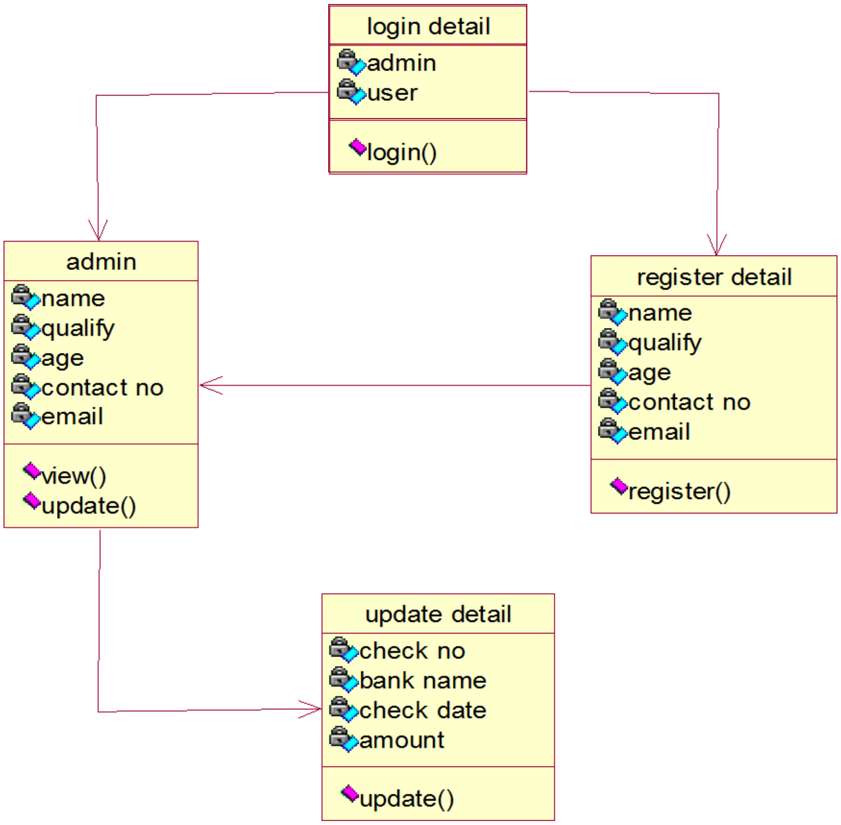


The activity diagram describes the sequence of activities with support for both conditional and parallel behavior.The activity diagram is used to describe the various activities taking place in an application.Here in our COURSE REGISTERATION SYSTEM,we have various activities strating from login.

After login,the user selection activity gets performed,where the user can be a user or admin.

If the user is a student,then they have to enter their password and those details are valid they can access the system.They can register the any course and get the reference number.

CLASS DIAGRAM:



The class diagram involves various classes used in project and their attributes.It also explains various class members details.

The four class in this project are,

1. Login detail
2. Admin detail
3. Register detail
4. Update detail

Each class has its own attributes and operations.

Login class-the attributes defined is user and admin

Admin class-It contains the attributes of the student and own details.

Register class-It contains the attributes of the student

Update class-the attributes of students can be added to the admin class

SOFTWAREDEVELOPMENT

Login Form:

```
Private Sub Command1_Click()
If Text1.Text = "rv" And Text2.Text
= "mk" Then
   coursedetail.Show
Else
MsgBox ("Invalid input")
End If
End Sub
```

Course Detail:

```
Private Sub Command1_Click()
If Option1.Value = True Then
instructions.Show
ElseIf Option2.Value = True Then
  instructions.Show
ElseIf Option3.Value = True Then
  instructions.Show
ElseIf Option4.Value = True Then
  instructions.Show
End If
End Sub
```

Instruction Form:

```
Private Sub Command1_Click()
If Button = Click_here Then
registration.Show
End If
End Sub

Private Sub Command2_Click()
If Button = Back Then
coursedetail.Show
End If
End Sub
```

Registration Form:

```
Private Sub Form_Load()
registerado.Recordset.AddNew
End Sub

Private Sub regbtn_Click()
registerado.Recordset.Fields("Rolln
o") = txtroll.Text
registerado.Recordset.Fields("Name
") = txtname.Text
registerado.Recordset.Fields("Class
") = txtclass.Text
registerado.Recordset.Fields("Addr
ess") = txtadd.Text
registerado.Recordset.Fields("mail"
) = txtmail.Text
registerado.Recordset.Fields("Conta
ct") = txtphone.Text
registerado.Recordset.Update
MsgBox "User Registration
Successful"
report.Show
End Sub
```

Exit Form:

```
Private Sub Command1_Click()
End
End Sub
```

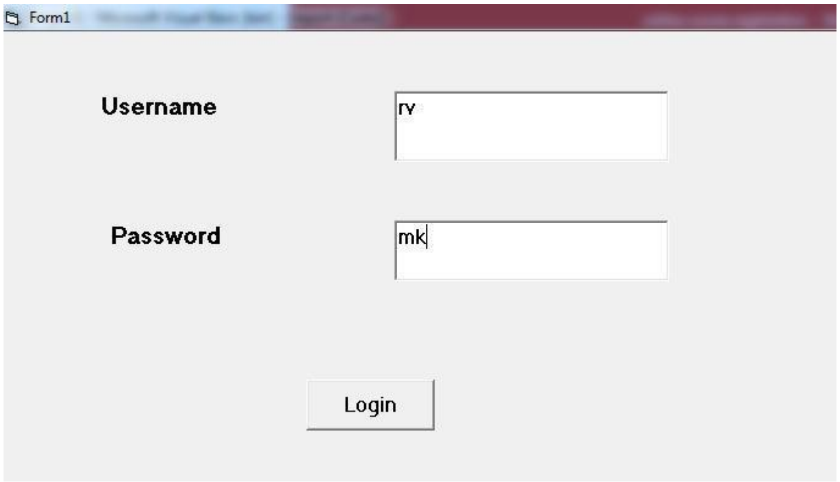
SOFTWARE TESTING:

TEST CASES:

- If an unauthorized user attempts to access the system, the system should not allow them to access.
- If the password entered by the user is incorrect, the system should display the bad password message and allows them to reenter the password or to terminate the process.

OUTPUT :

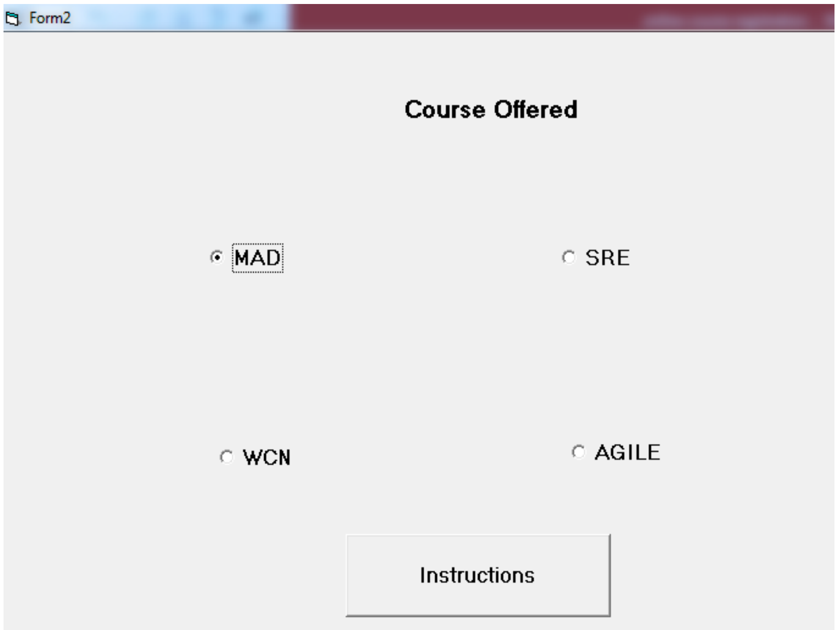
Form1:



Username rv

Password mk

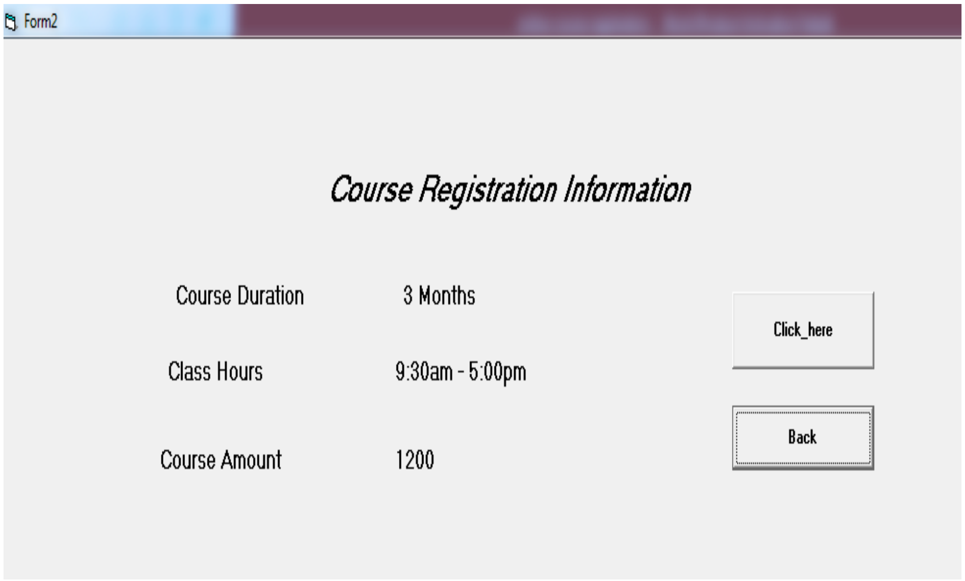
Form2:



● MAD ○ SRE

○ WCN ○ AGILE

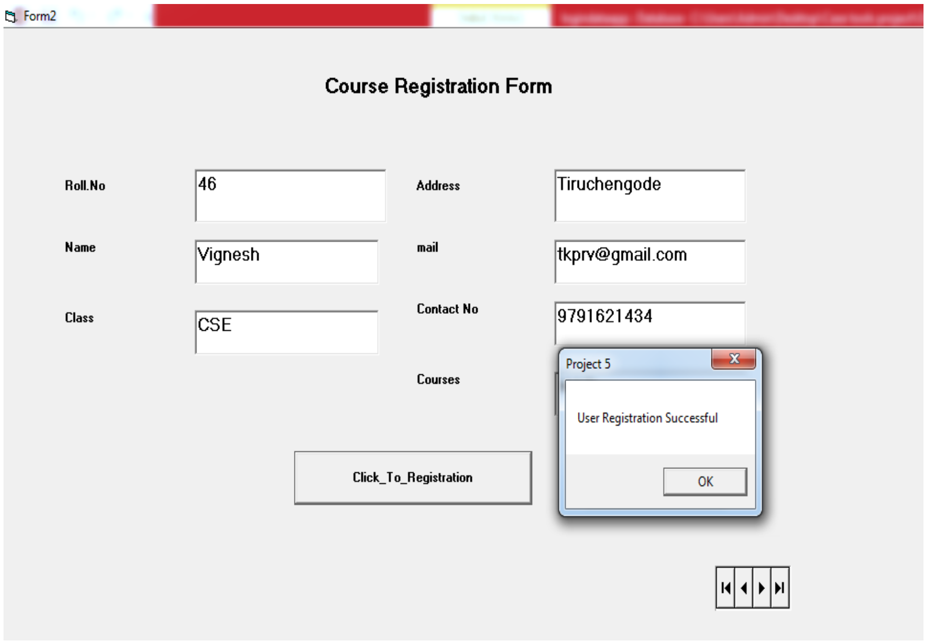
Form3:



Course Registration Information

Course Duration	3 Months
Class Hours	9:30am - 5:00pm
Course Amount	1200

Form4:



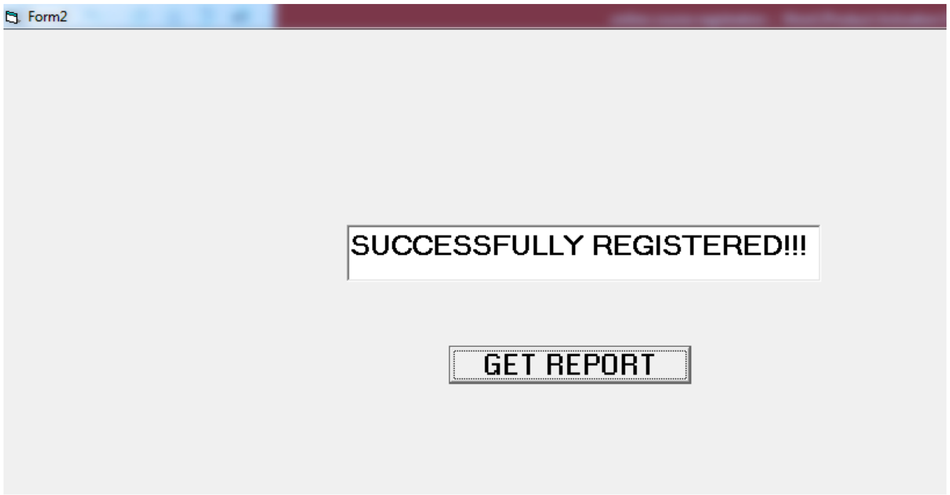
Course Registration Form

Roll.No	46	Address	Tiruchengode
Name	Vignesh	mail	tkprv@gmail.com
Class	CSE	Contact No	9791621434
		Courses	

Project 5

User Registration Successful

Form5:



SUCCESSFULLY REGISTERED!!!

Database:

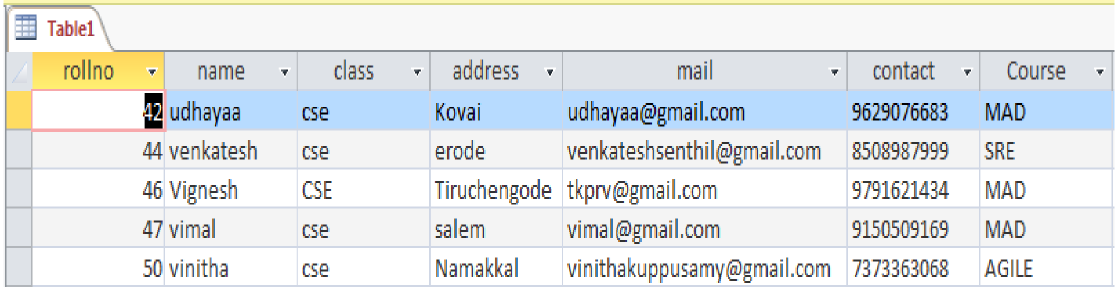


Table1

rollno	name	class	address	mail	contact	Course
42	udhayaa	cse	Kovai	udhayaa@gmail.com	9629076683	MAD
44	venkatesh	cse	erode	venkateshsenthil@gmail.com	8508987999	SRE
46	Vignesh	CSE	Tiruchengode	tkprv@gmail.com	9791621434	MAD
47	vimal	cse	salem	vimal@gmail.com	9150509169	MAD
50	vinitha	cse	Namakkal	vinithakuppusamy@gmail.com	7373363068	AGILE

RESULT:

Thus the **Online Voting** as **Course Registration System** is developed with all documents and UML diagrams using Rational Rose software Engineering methodology.

EX.NO:6
DATE :

LIBRARY MANAGEMENT SYSTEM

Aim:

To prepare necessary documents and to develop the LIBRARY MANAGEMENT SYSTEM with UML diagrams using Software Engineering Methodology.

PROGRAM ANALYSIS AND PROJECT PLANNING

Problem Statement:

This project LIBRARY MANAGEMENT SYSTEM is to develop an application to analyze the books details. Initially the user should login. After login to lend the book the user should click book lending command button. If the user wants to return the book then must choose the book returning command button. If the user wants to view the details of book lending and returning history then must click the report command button.

SOFTWARE REQUIREMENT ANALYSIS

The Modules in the Project:

1. Login.
2. Password Checking
3. Book lending
4. Book returning
5. Report

The first module is Login in which the user has to login to the system.

The next module is Password Checking, where the checking of name and password occurs if correct then only the user can lend or return the book.

The next module is Book lending. This module is useful to lend the book from library.

The next module is Book returning. This module is useful to return the book to the library.

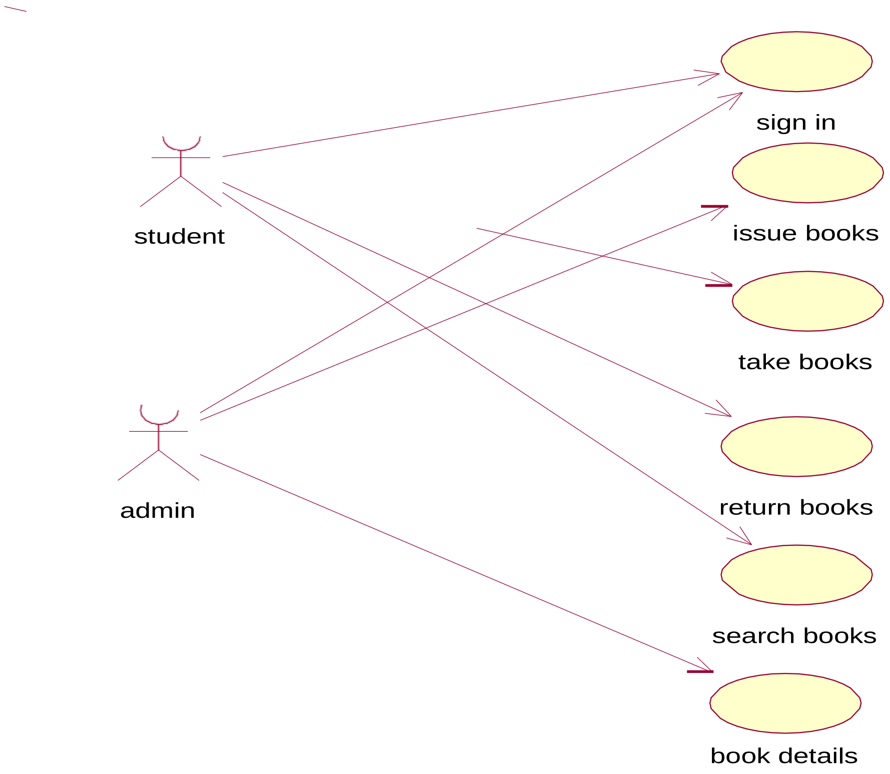
The last module is Report. This module is to view the history.

DATA MODELING

Data Dictionary:

Field Name	Description	Data Type	Field size	Default Value	Validation
ID.NO	Identification Number	Integer	10	NULL	(0-9)
Name	Name of the user	String	20	NULL	(A-Z) or (a-z)
Date	Date	Integer	10	NULL	(MM/DD/YYYY)
Author	Name of the author	String	10	NULL	(A-Z) or (a-z)
Book	Name of the book	String	10	NULL	(A-Z) or (a-z)

USE CASE DIAGRAM:



Use case Diagram:

Use case diagram is a graph of actors, set of use cases enclosed by a system boundary, communication (participation) association between the actors and the use cases and a generalization among the use cases.

Actor:

An actor represent a set of roles that user of a use case play when interacting with the use cases. Actor identified here is user.

Use case:

A use case is a description of a set of sequence of actions that a system performs to yield result of value to an actor.

The Use Cases described are,

1. Login
2. Book lending
3.Book returning
4. Report

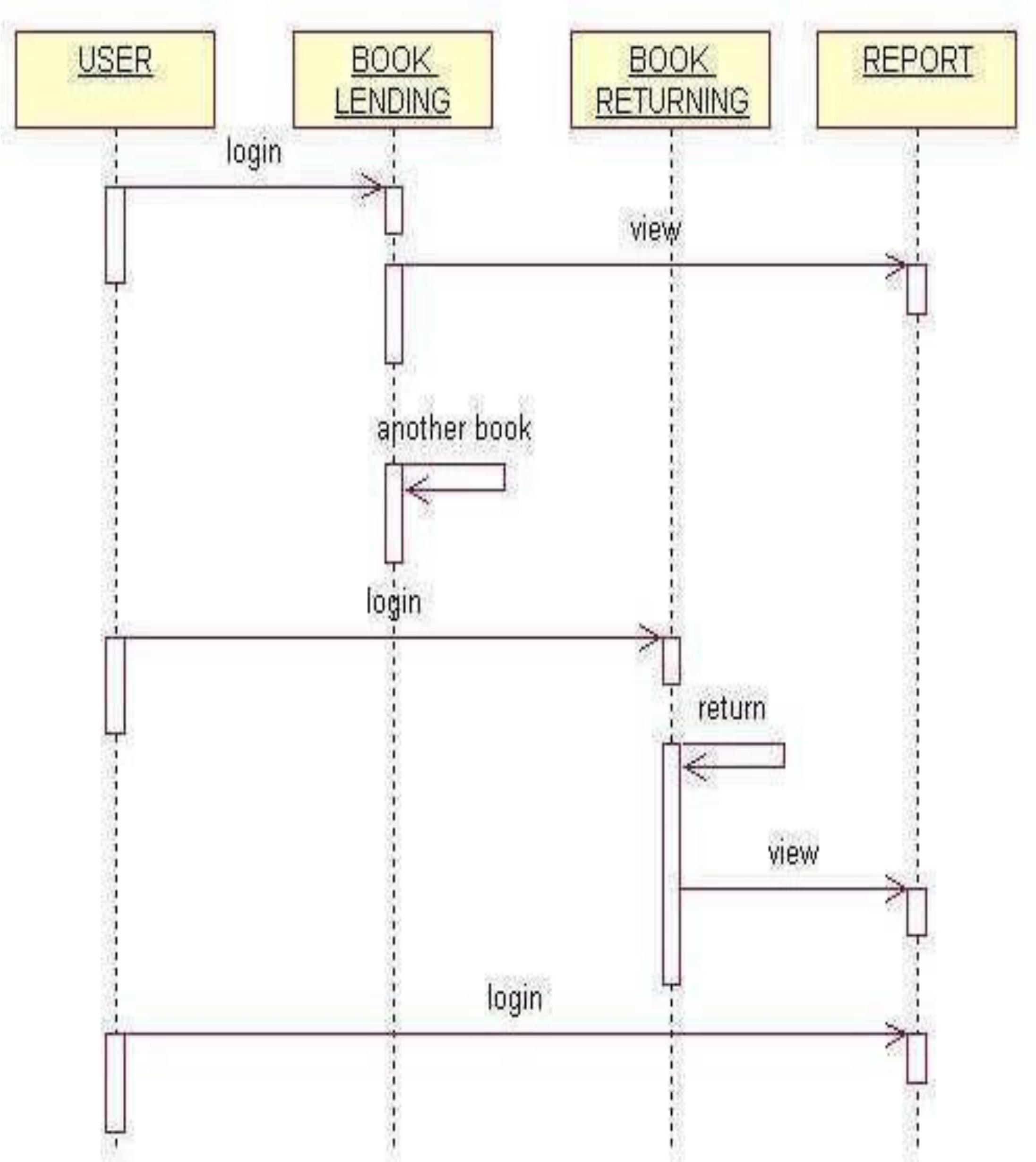
The Login use case is to describe that, the user should choose his/her category whether he/she is a administrator or staff.

SEQUENCE DIAGRAM

Sequence diagrams are easy and intuitive way of describing the behavior of a system by viewing the interaction between the system and its environment. A sequence diagram shows an interaction arranged in a time sequence.

The objects used in this sequence diagram are,

a. Login
b. Display
c. Add
d. Update
e. Controller
f. Database



BASIC FLOW:

The administrator enters their name and password, and the password gets checked by the system. After confirmation of the password the system allows them to access.

ALTERNATE FLOW:

If the password entered by the administrator is invalid then they has to reenter or quit from the process.

The actor Administrator and Staff are the persons who interacts with the system.

The object Login makes the administrator/staff to enter.

The object Database will store all the salary of the staff. This will also store the details of new records added by the administrator. It will also update the values changed by the administrator.

The object Display will display the salary of the staffs from the database.

The object Add will add the new records in the database.

The object Update will update the details in the database.

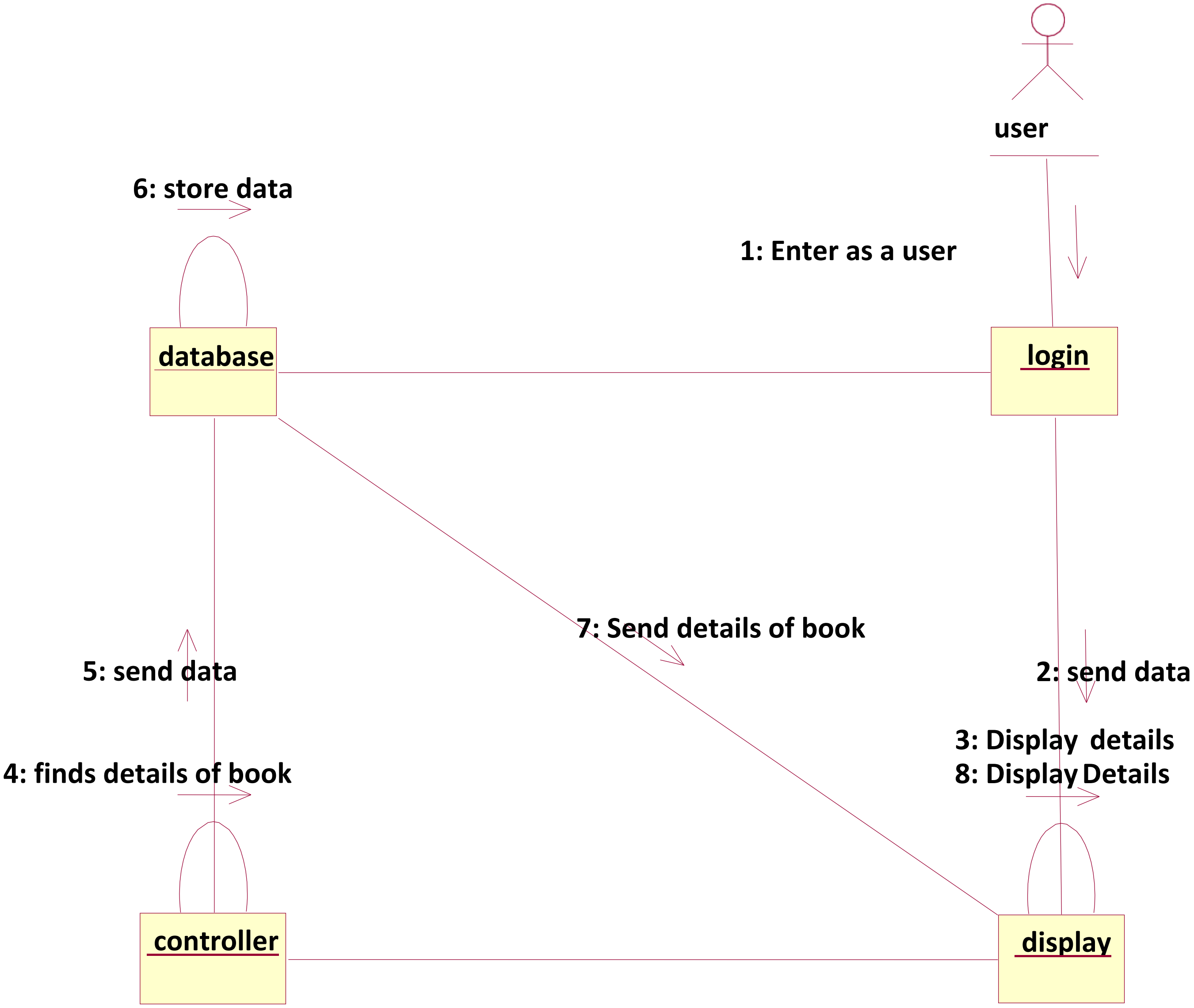
COLLOBORATION DIAGRAM:

A collaboration diagram represents a collaboration, which is a set of objects related in a particular context, and interaction, which is a set of messages exchanged among the objects within the collaboration to achieve a desired outcome.

Collaboration diagram shows exactly the same information as the sequence diagram. However, collaboration diagram shows this information in a different way and with different purpose.

In this collaboration diagram, the objects are represented as rectangle, the actors are stick figures. Whereas the sequence diagram illustrates the object and actor interaction overtime, the collaboration diagram shows the object and actor interaction without reference to time.

In our LIBRARY MANAGEMENT SYSTEM each object interacts with each other or collaborates with each other; it gets represented by the solid line drawn between them.



ACTIVITY DIAGRAM:

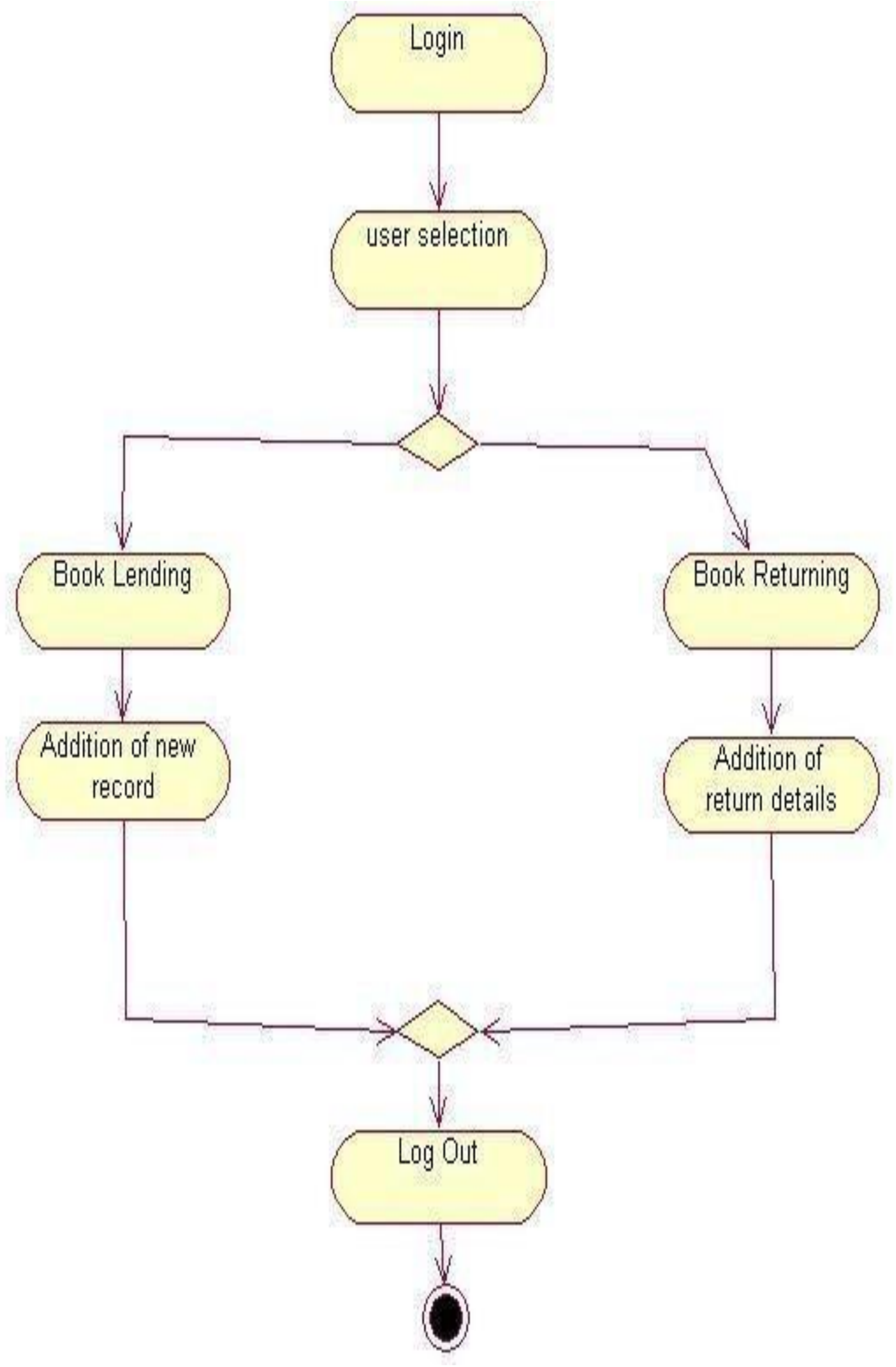
The activity diagram describes the sequencing of activities with support for both conditional and parallel behavior.

The Activity diagram is used to describe the various activities taking place in an application.

After login, the user selection activity gets performed, where the user can be an administrator or staff. If the user is a administrator, then they have to enter their name and password and only when those details are valid they can access the system.

They can calculate the current salary obtained by the staffs, they can add new records, and they can update the values of the records which gets stored in the database.

If the user is a staff then they can view their salary detail and they can calculate their current salary.

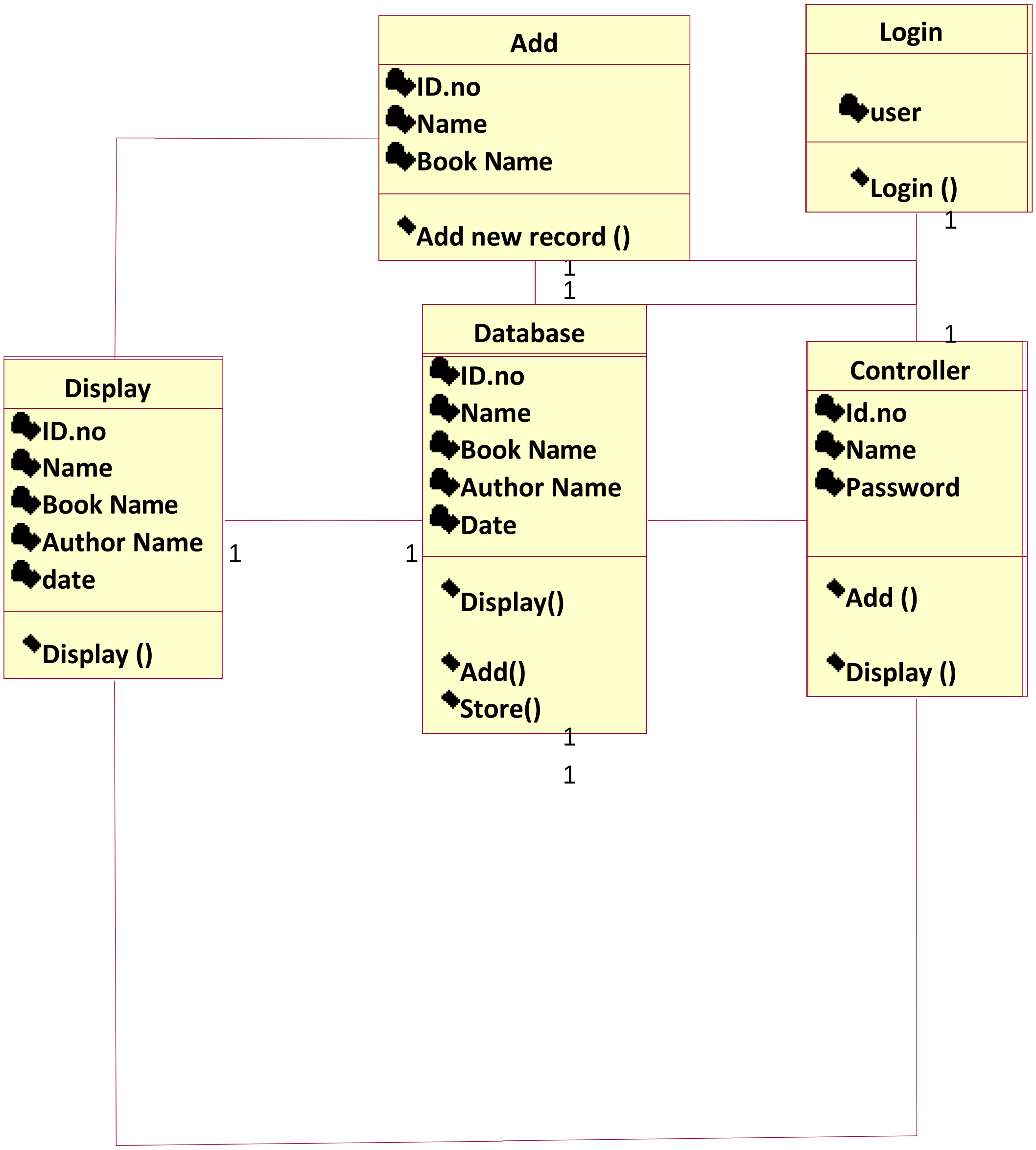


CLASS DAIGRAM:

Class diagrams show the interactions between classes in the system. Class diagram also shows the attributes and operation of a class and the constraints that apply to the way objects are connected.

Classes contain information and behavior that acts on that information.

Each class on class diagram is represented by rectangle divided into three sections.



The first section shows the class name, second section shows the attributes the class contains and last section contains the operation of the class.

In our LIBRARY MANAGEMENT SYSTEM, the classes identified are

1. Login
2. Display
3. Add
4. Controller
5. Database

Each class has its own attributes and operations.

Login class - The attributes defined is user.
The method identified is login.

Display class - The attributes are name, id, book name, author name, date.
The operation identified is Display.

Add class - The attributes are name, id, book name, author name, date.
The operation defined is adding new record.

Controller class - The attributes it has is id.no, name, and password.
The operations carried out by this class are added, display.

The Solid line between the classes shows the Association relationship between them.

SOFTWARE DEVELOPMENT

Login Form:

```
Private Sub Command1_Click()
If Text1.Text = "library" And Text2.Text = "cse" Then
  books.Show
Else
MsgBox ("Invalid input")
End If
End Sub
```

Lending Books:

```
Private Sub Command1_Click()
lendingbooks.Show
End Sub
Private Sub Form_Load()
lendingbooksado.Recordset.AddNew
End Sub
Private Sub Conbtn_Click()
lendingbooksado.Recordset.Fields("Regno") = txtreg.Text
lendingbooksado.Recordset.Fields("Name") = txtname.Text
lendingbooksado.Recordset.Fields("Book name") = txtbook.Text
lendingbooksado.Recordset.Fields("Author name") = txtauthor.Text
lendingbooksado.Recordset.Fields("Date of lending books") =
txtlendingbooks.Text
lendingbooksado.Recordset.Update
MsgBox "User lendingbooks Successful"
returnbooks.Show
End Sub
```

Return Books:

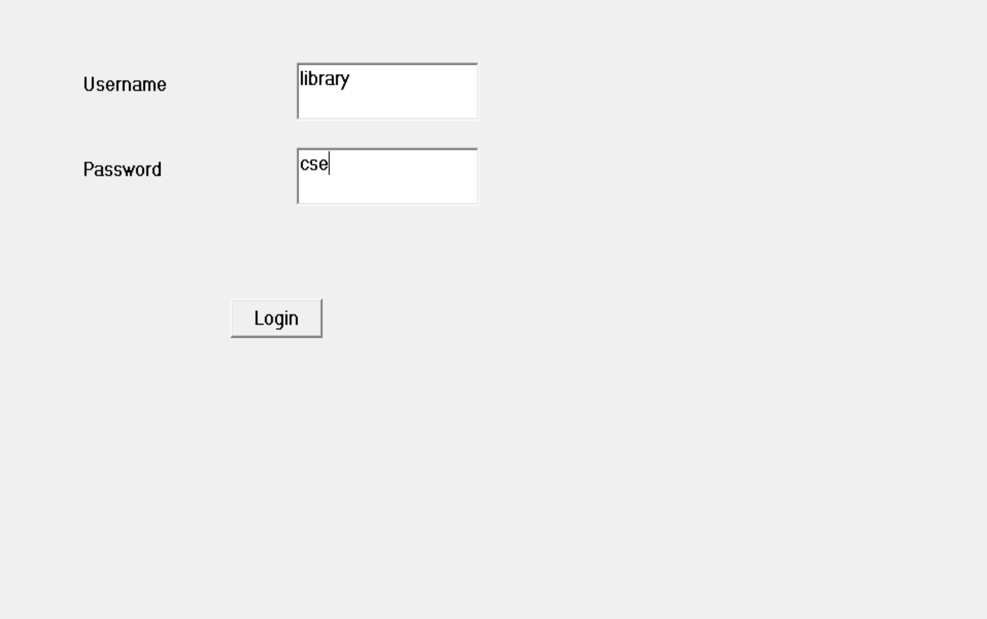
```
Private Sub Form_Load()
returnbooksado.Recordset.AddNew
End Sub
Private Sub Confbtn_Click()
returnbooksado.Recordset.Fields("Regno") = txtreg.Text
returnbooksado.Recordset.Fields("Name") = txtname.Text
returnbooksado.Recordset.Fields("Book name") = txtbook.Text
returnbooksado.Recordset.Fields("Author name") = txtauthor.Text
returnbooksado.Recordset.Fields("Date of Return Books") = txtreturnbooks.Text
returnbooksado.Recordset.Update
MsgBox "User returnbooks Successful"
End Sub
```

Exit:

```
Private Sub exitbtn_Click()
End
End Sub
```

OUTPUT:

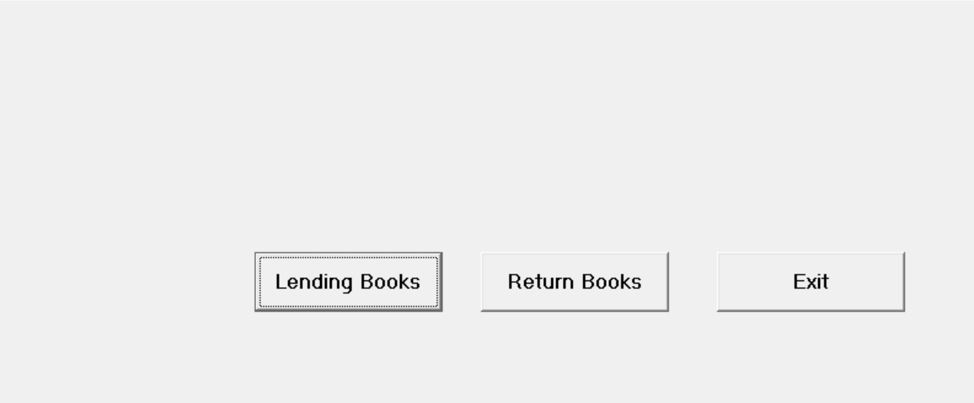
Form1:



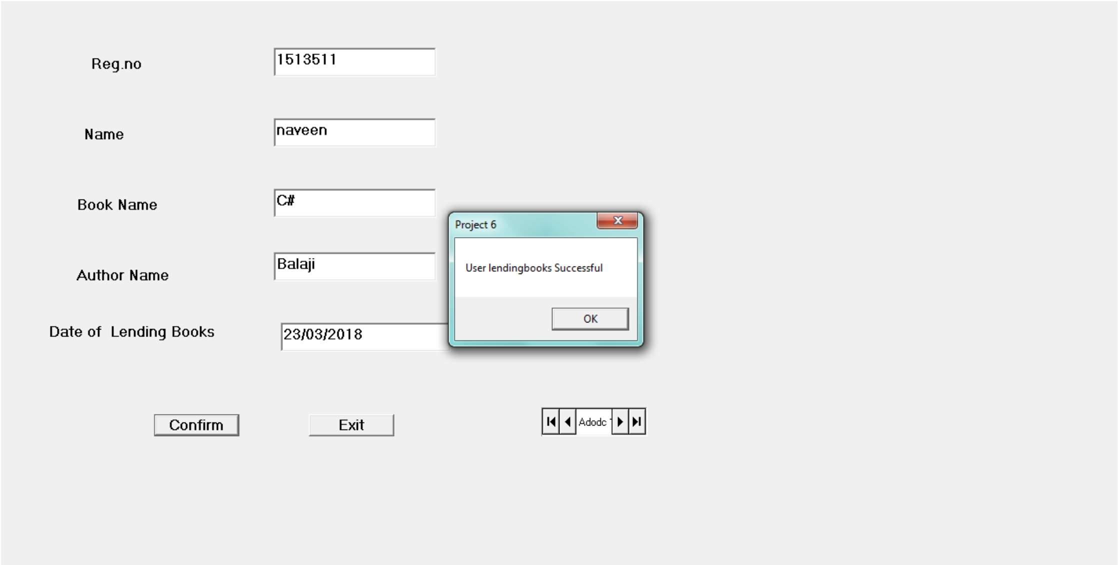
Username library

Password cse

Form2:



Form3:



Reg.no 1513511

Name naveen

Book Name C#

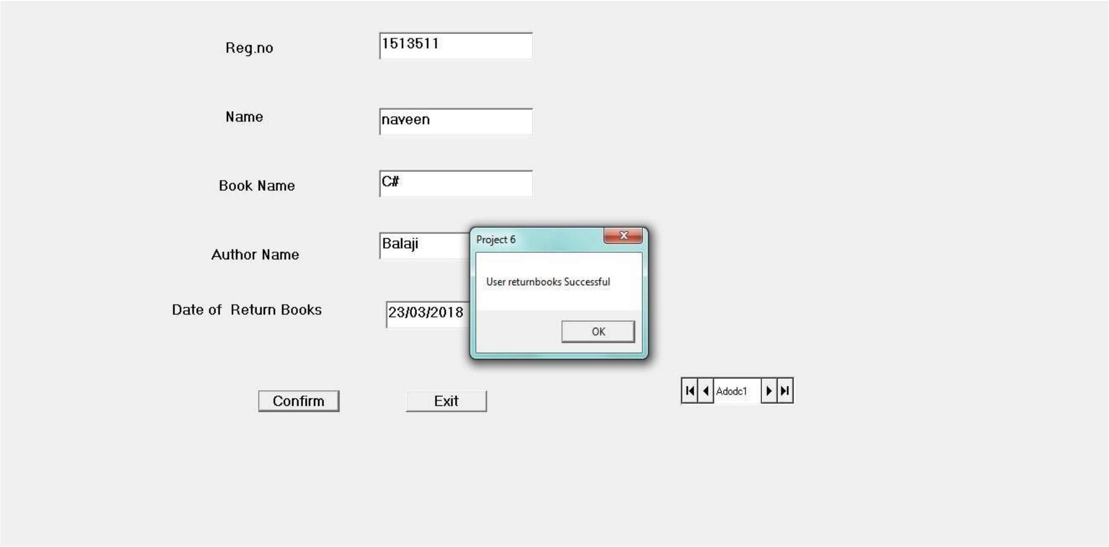
Author Name Balaji

Date of Lending Books 23/03/2018

Project 6

User lendingbooks Successful

Form4:



Reg.no 1513511

Name naveen

Book Name C#

Author Name Balaji

Date of Return Books 23/03/2018

Project 6

User returnbooks Successful

Database:

Lending Books:

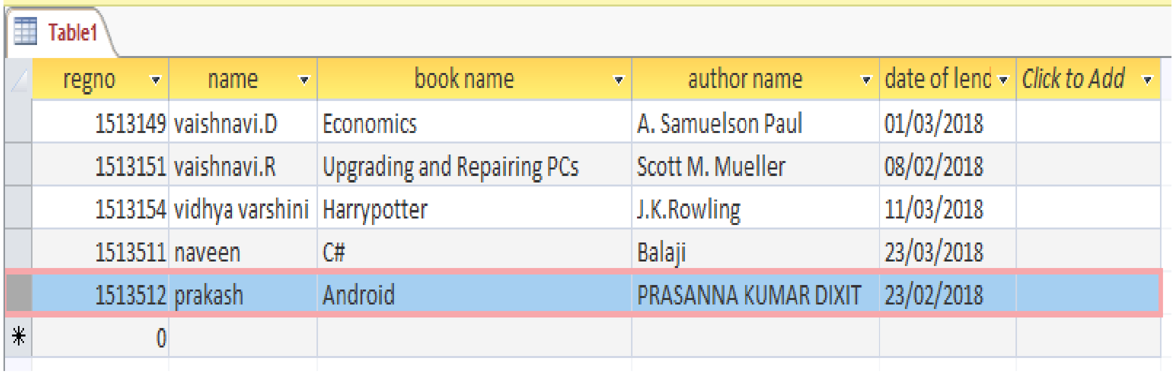


Table1

	regno	name	book name	author name	date of lenc	Click to Add
	1513149	vaishnavi.D	Economics	A. Samuelson Paul	01/03/2018	
	1513151	vaishnavi.R	Upgrading and Repairing PCs	Scott M. Mueller	08/02/2018	
	1513154	vidhya varshini	Harrypotter	J.K.Rowling	11/03/2018	
	1513511	naveen	C#	Balaji	23/03/2018	
	1513512	prakash	Android	PRASANNA KUMAR DIXIT	23/02/2018	
*	0					

Return Books:

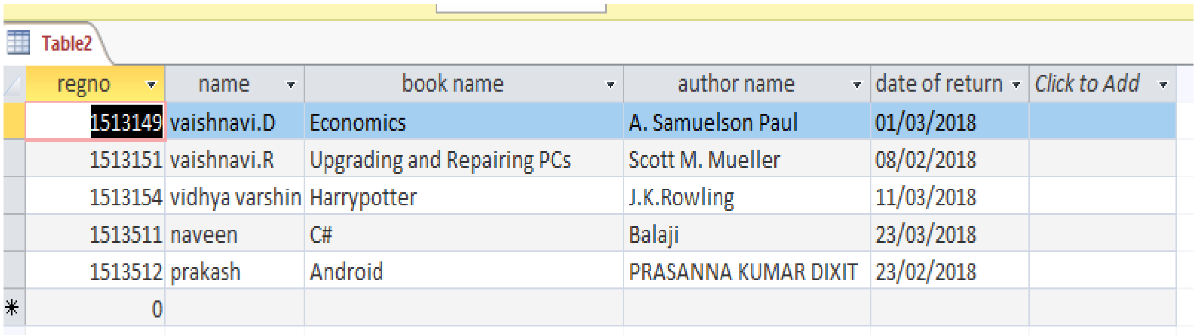


Table2

	regno	name	book name	author name	date of return	Click to Add
	1513149	vaishnavi.D	Economics	A. Samuelson Paul	01/03/2018	
	1513151	vaishnavi.R	Upgrading and Repairing PCs	Scott M. Mueller	08/02/2018	
	1513154	vidhya varshin	Harrypotter	J.K.Rowling	11/03/2018	
	1513511	naveen	C#	Balaji	23/03/2018	
	1513512	prakash	Android	PRASANNA KUMAR DIXIT	23/02/2018	
*	0					

RESULT:

Thus the **LIBRARY MANAGEMENT SYSTEM** is developed with all necessary documents and UML diagrams using Software Engineering methodology.