**SOFTWARE ENGINEERING**

**Tutorial Sessions**

**V Semester ‘B’ Section**

**1. CASE tools**

CASE tools known as Computer-aided software engineering tools is a kind of component-based development which allows its users to rapidly develop information systems. The main goal of case technology is the automation of the entire information systems development life cycle process using a set of integrated software tools, such as modeling, methodology and automatic code generation

Some typical CASE tools are:

* **Unified Modeling Language**
* Data modeling tools, and
* Source code generation tools

**2. Unified Modeling Language**

he Unified Modeling Language (UML) is used to specify, visualize, modify, construct and document the artifacts of an object-oriented software-intensive system under development. UML offers a standard way to visualize a system's architectural blueprints, including elements such as:

* activities
* actors
* business processes
* database schemas
* (logical) components
* programming language statements
* Reusable software components.[[2]](http://en.wikipedia.org/wiki/Unified_Modeling_Language#cite_note-OMG00-1)

UML combines techniques from data modeling (entity relationship diagrams), business modeling (work flows), object modeling, and component modeling. It can be used with all processes, throughout the software development life cycle, and across different implementation technologies

**2.1 UML class diagram**

The UML class diagram is also known as object modeling. It is a static analysis diagram. These diagrams show the static structure of the model

The purpose of a class diagram is to depict the classes within a model. In an object oriented application, classes have attributes (member variables), operations (member functions) and relationships with other classes.

**A class diagram is represented as:**

<<Class name>>

<<Attribute 1>>

<<Attribute n>>

<<Operation ()>>

**2.2 Use Case Diagrams**

An important part of the Unified Modeling Language (UML) is the facilities for drawing use case diagrams. Use cases are used during the analysis phase of a project to identify and partition system functionality. They separate the system into *actors* and *use cases*.

Actors represent roles that can are played by users of the system. Those users can be humans, other computers, pieces of hardware, or even other software systems. The only criterion is that they must be external to the part of the system being partitioned into use cases. They must supply stimuli to that part of the system, and the must receive outputs from it.

Use cases describe the behavior of the system when one of these actors sends one particular stimulus. This behavior is described textually. It describes the nature of the stimulus that triggers the use case; the inputs from and outputs to other actors, and the behaviors that convert the inputs to the outputs. The text of the use case also usually describes everything that can go wrong during the course of the specified behavior, and what remedial action the system will take.

**2.3 Sequence diagram:**

A sequence diagram is a graphical view of scenario that shows object interaction in a time based sequence what happens first what happens next. Sequence diagrams are closely related to collaboration diagram.

The main difference between sequence and collaboration diagram is that sequence diagram show time based interaction while collaboration diagram shows objects associated with each other

**2.4 Collaboration diagram**:

Collaboration diagram and sequence diagrams are alternate representations of an interaction. A collaboration diagram is an interaction diagram that shows the order of messages that implement an operation or a transaction. Collaboration diagram is an interaction diagram that shows the order of messages that implement an operation or a transaction. Collaboration diagram shows object s, their links and their messages. They can also contain simple class instances and class utility instances.

During, analysis indicates the semantics of the primary and secondary interactions. Design, shows the semantics of mechanisms in the logical design of system.

**List of Experiments:**

1. ATM system
2. Student marks analysing system
3. Employee payroll system
4. Airline ticket reservation system
5. Course Registration System

**ATM SYSTEM**

**Aim:**

To create a system to perform Bank ATM transaction

**Problem statement:**

This system is build for the bank client and the manager.

* The bank client must be able to deposit and withdraw amount from his/her accounts using the ATM machine. Each transaction must be recorded and the client must be able to review all transactions performed in his/her account. Recorded transactions must include the date, time, transaction type, amount and account balance after the transaction.
* The bank manager must be able to view the ATM machine status that is the total balance of the ATM machine, today’s withdrawal, today’s balance and the limitations of the machine.
* The bank client is provided by login verification. If it is valid he/she will access their account otherwise an appropriate message is displayed to the client.

**Software requirements:**

Microsoft visual basic 6.0 is used as front-end for our project and ms-access is used as back-end to store the data.

**USE-CASE diagram:**

The ATM transaction use cases in our system are:

1. Login
2. Withdraw
3. Mini statement
4. ATM machine status
5. Deposit

**Actors involved:**

1. User
2. Bank manager

**USE-CASE name: Login**

The user enters a user name and password. If it is valid, the user’s account becomes available. If it is invalid, an appropriate message is displayed to the user

**USE-CASE name: Withdraw**

The user tries to withdraw an amount from his or her checking account. The amount is less than or equal to the checking account’s balance, the transaction is performed and the available information is displayed. The system creates a record of the transaction and the display confirmation message is displayed to the client.

**USE-CASE name: Mini statement**

The bank user requests a history of transactions for a checking account. The system displays the transaction history for the checking account. The transaction history consists of amount, date, transaction type and balance of the particular account.

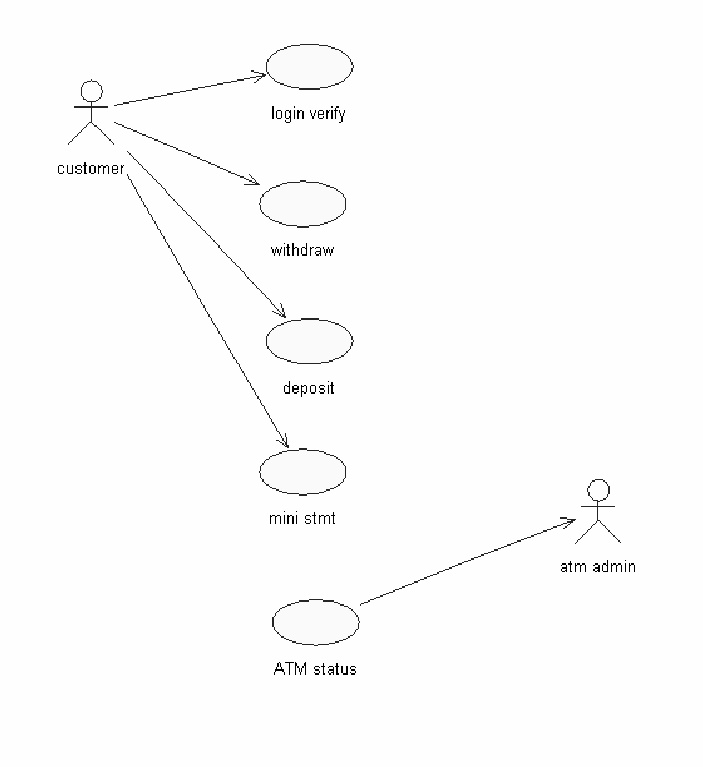
**USE-CASE name: ATM machine status**

The bank manager enters a username and password. If it is valid, the bank manager accesses the machine status. If it is invalid, an appropriate message is displayed to the user

**USE-CASE name: Deposit**

The bank user requests the system to deposit money to an account. The user accesses the account for which a deposit is going to be made and enters the amount. The system creates a record of the transaction and an appropriate confirmation message (display confirmation) is displayed to the client. The transaction must include the date, type, amount and account balance after the transaction

**Use-case diagram for ATM system**



**Class diagram**

The class diagram, also referred to as object modeling is the main staticanalysis diagram. The main task of object modeling is to graphically show what eachobject will do in the problem domain. The problem domain describes the structure andthe relationships among objects.The

ATM system class diagram consists of four classes:

* 1. User class
  2. ATM machine status
  3. Account
  4. Transaction

**1) User class:**

It consists of four attributes and two operations. The attributes are user name, password, address and DOB. The operations of this class are read (), display () and write ().

**2) ATM machine status:**

The attributes of this class are ATM balance, today’s withdrawal, today’s balance, and limitations. The operations are login verification (), ATM status () and display confirmation ().

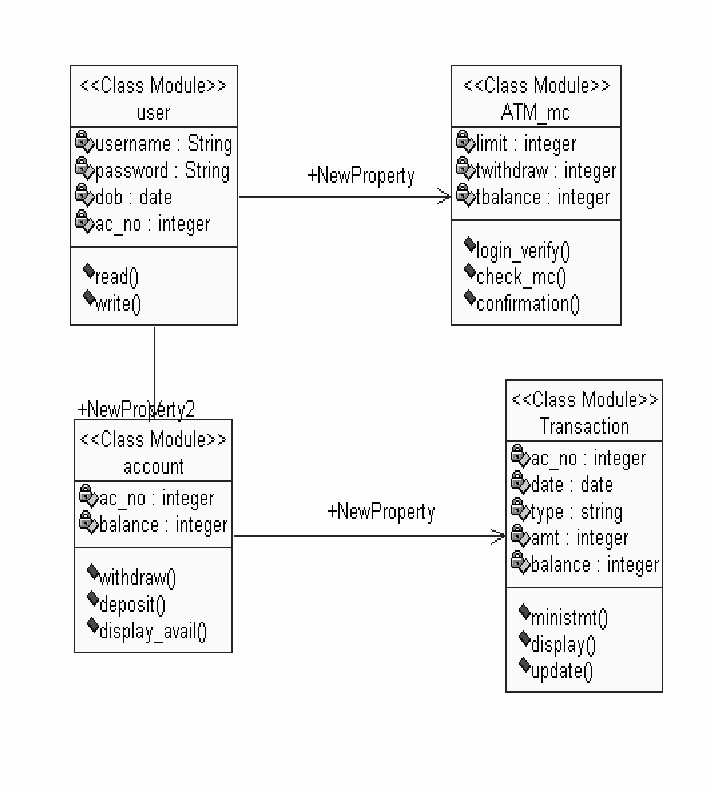
**3) Account:**

The attributes are account no. and balance and the operations are withdraw (), deposit () and display availability ().

**4) Transaction:**

The attributes of this class are account no, transaction type, data, amount, balance and the operations are mini statement () and create transaction ()

**Class diagram for ATM system**



**Sequence diagram:**

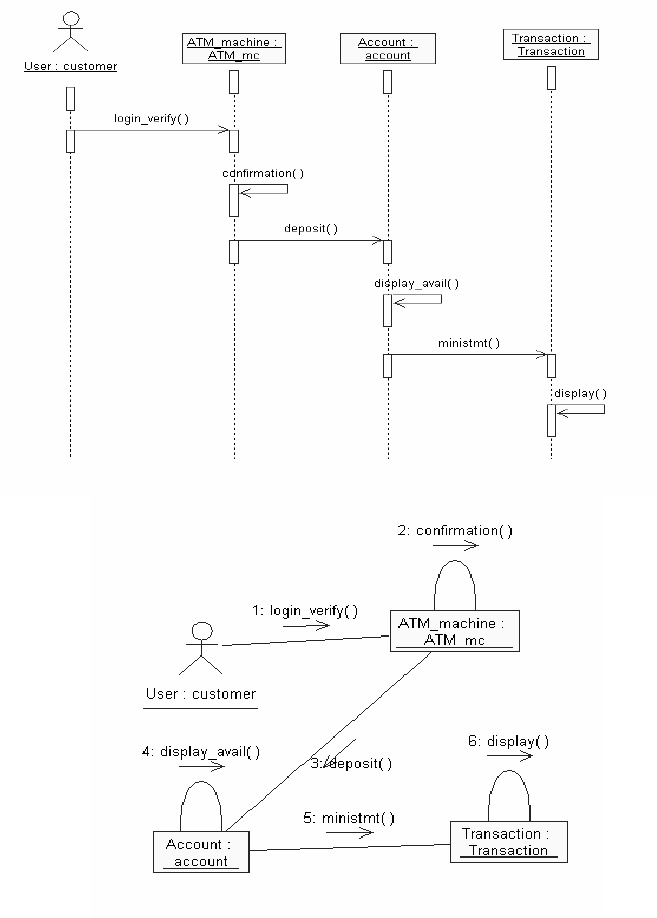
A sequence diagram represents the sequence and interactions of a given USE-CASE or scenario. Sequence diagrams can capture most of the information about the system. Most object to object interactions and operations are considered events and events include signals, inputs, decisions, interrupts, transitions and actions to or from users or external devices.

An event also is considered to be any action by an object that sends in formation. The event line represents a message sent from one object to another, in which the “form” object is requesting an operation be performed by the “to” object. The “to” object performs the operation using a method that the class contains.

It is also represented by the order in which things occur and how the objects in ;the system send message to one another.

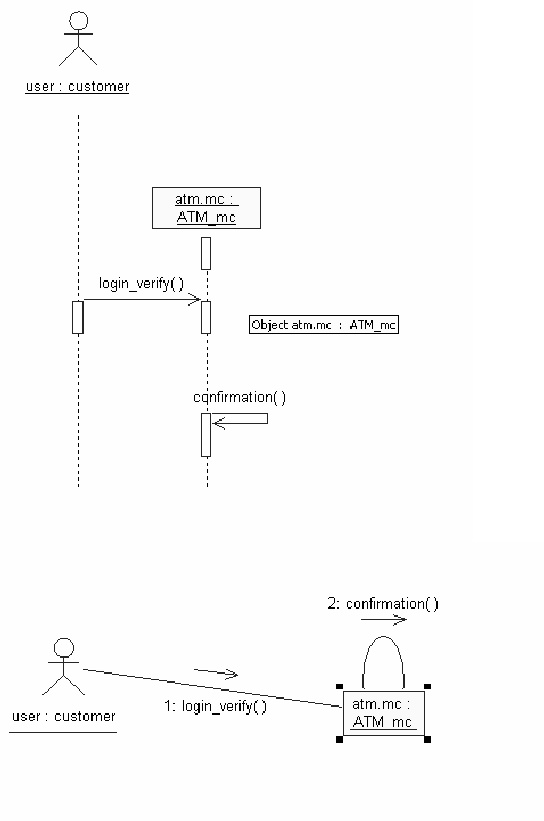
The sequence diagram for each USE-CASE that exists when a user withdraws, deposits, needs information about ATM machine status and account are drowned

**Sequence and collaboration diagram for deposit process**



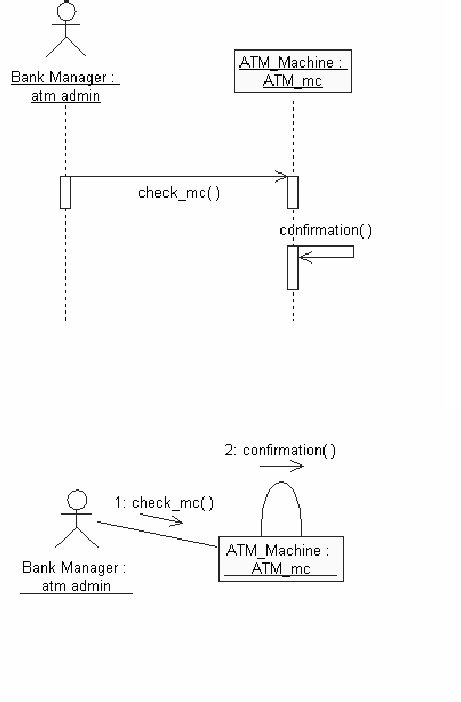
The diagrams show the entire deposit process in an ATM system. The user has to login to the ATM machine and deposit the amount of money as required by the user. The user may wish to get a mini statement and a screen about the details of the transaction.

**Sequence and collaboration diagram for login**

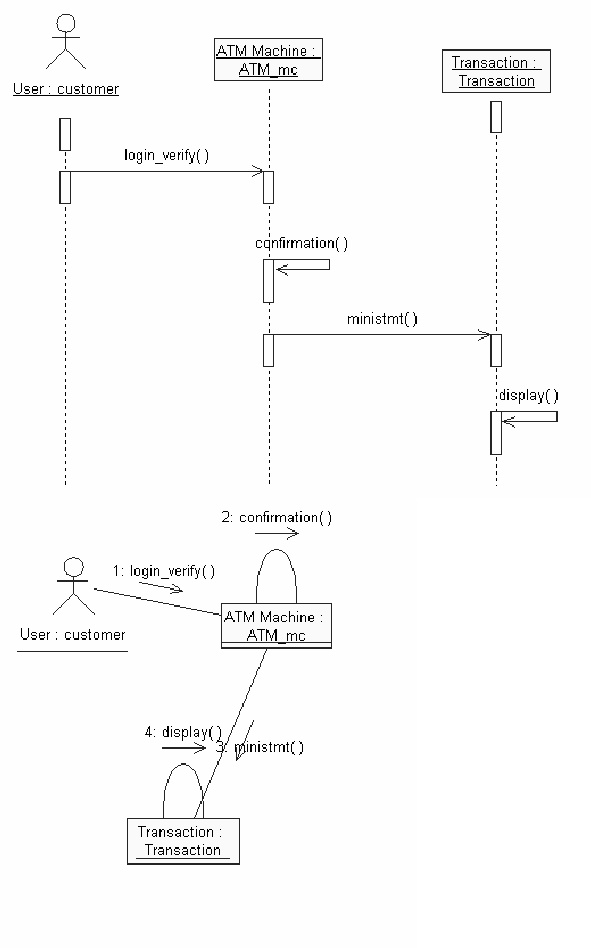


The diagrams show the process of login by the user to the ATM system. The user has to enter his details. The details entered are verified by the system and the user is approved if the details match, otherwise an appropriate error message is displayed.

**Sequence and collaboration diagram for checking machine status**

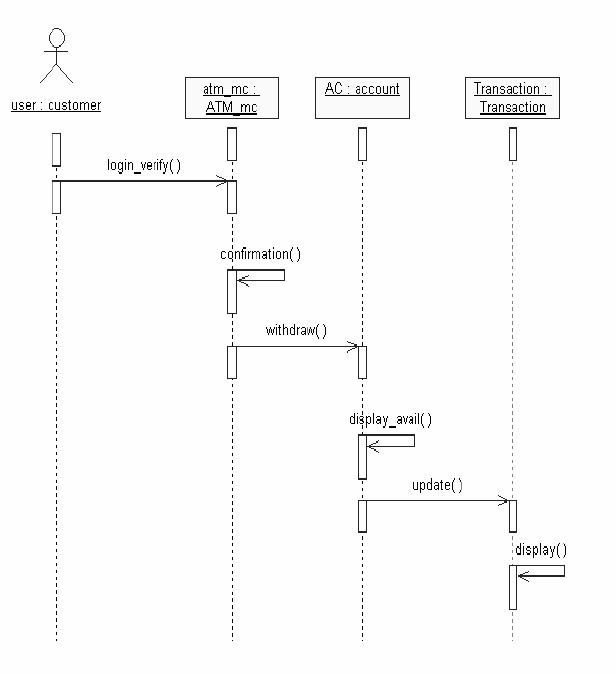


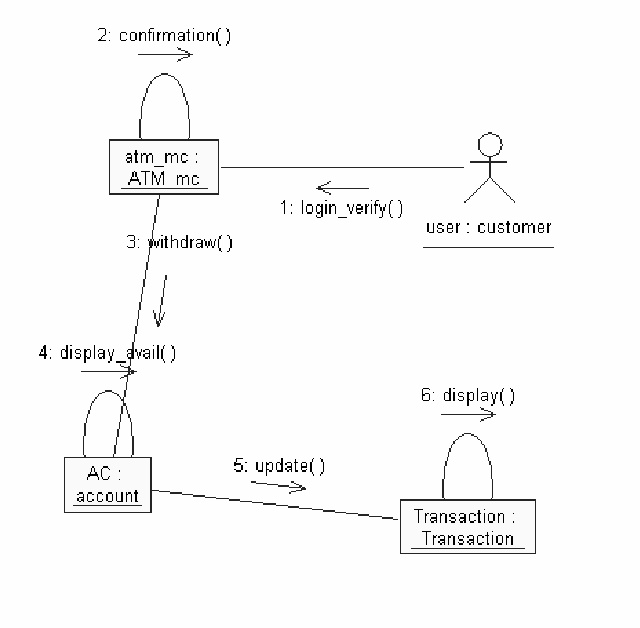
The Administrator of the ATM system has to maintain the details about the ATM, He has to check if there is enough money in the ATM and if the ATM is functional without major errors. For this, he may check the ATM machine status occasionally. The process is shown in the above diagrams.

**Sequence and collaboration diagram for printing mini statement**

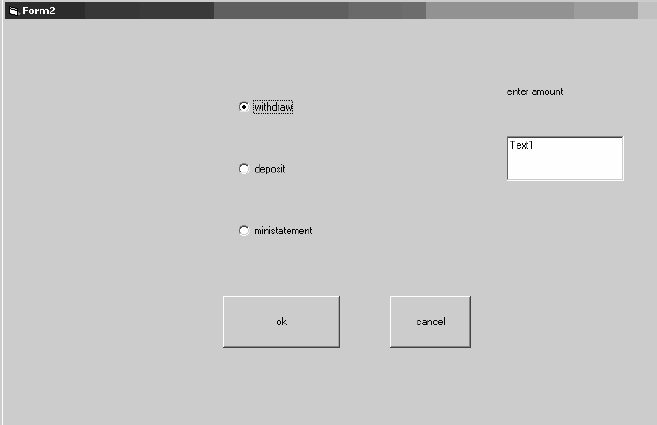
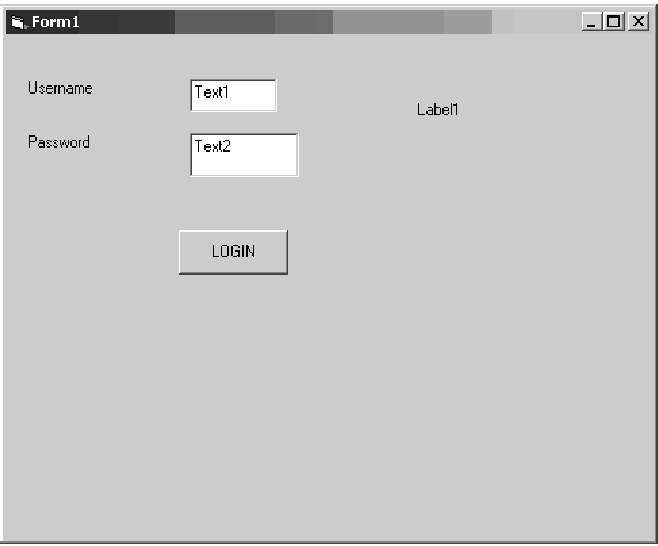
After a transaction is carried out successfully, the user must get a mini statement to tell him his account’s details such as balance and transaction number. This process is depicted in the above diagrams

**Sequence and collaboration diagram for withdraw process**

****

****

The user can make withdraw money from his account. The process is depicted in the diagrams above. The user has to login to the system using his username and password, which are verified by the system. After successful verification, the user can choose the amount of money he wants to withdraw from his account. The amount specified by the user is checked by the system to make sure there is enough balance in his account to carry out the transaction. After the transaction is carried out the resulting amount is displayed and the details are updated to the database.

****

**STUDENT MARK ANALYSIS SYSTEM**

**AIM:**

To create a system to analyze the students marks stored in the database.

**PROBLEM STATEMENT:**

The purpose of this system is to analyze and perform operation on data stored in the database and to provide authentication to avoid unauthorized access by using MS-Access as back end and VB to use as the front end. The entire system is divided into various modules and it has its own objects and classes.

**Reports are provided on their basis:**

* Student marks
* Average
* Percentage

**System requirements**

Microsoft visual basic 6.0 is used as the front-end for our project and ms-access is used as the back-end.

**Use-case diagram**

1. Login
2. Student personal detail
3. Student marks
4. Display details
5. Calculate percentage

**Actors involved**

1. Student
2. Administrator

**Employee Payroll system**

**Aim:**

To create a computer system to process employee’s payroll.

**Problem statement**

This system is built for employee and manager.

The employee can view his salary and the manager can add a new employee, update employee details and calculate and credit the salary.

The employee and manager can log into the system providing their respective user names at login, if the entered input is incorrect an appropriate message is displayed.

**System requirements**

Microsoft visual basic 6.0 is used as front-end for our project and ms-access as our back-end.

**Use-case diagram**

The payroll use cases in our system are:

1 .Login

2. Add employee

3. Update employee details

4. Delete employee

5. Calculate salary

6. View employee details

7. Logout

**Actors involved**

1. Manager

2. Employee

**AIRLINE TICKET RESERVATION SYSTEM**

**Introduction:**

The manual system of ticket reservation takes more time and the number of reservations per day is limited. To increase the efficiency of the process, we go for online ticket reservation system. This system supports online ticket booking.

**Problem statement**

This system is built for user to directly access the system online to book tickets. The user can book, print, delete tickets without the help of a clerk. The administrator has control over the adding flights available for booking and has control over deleting flights that are not necessary. The administrator and user can both enter the system using their respective login details

**System requirements**

Microsoft visual basic 6.0 is used as the front-end for our project and ms-access is used as the back-end.

**Use-case diagram**

The online ticket reservation system uses the following use cases:

1. Login

2. Book ticket

3. Print ticket

4. Cancel ticket

5. View flight

6. Add flight

7. Delete flight

8. Logout

**Actors involved**

1) Administrator

2) Passenger

**Course Registration System**

**Aim**

To create a system through which students can register to the courses desired by them.

**Problem statement**

* The system is built to be used by students and managed by an administrator.
* The student and employee have to login to the system before any processing can be done.
* The student can see the courses available to him and register to the course he wants.
* The administrator can maintain the course details and view all the students who have registered to any course.

**System requirements**

Microsoft visual basic 6.0 is used as the front end of our project and ms-accessas the back end.

**Use-case diagram**

The course registration system has the following use-cases

1. Login
2. View course details
3. Registration
4. Display details
5. Maintain course details
6. Logout

**The actors involved in the system are**

1. Student

2. Administrator