# COURSE REGISľRAľION SYSľEM

1) Perform the following for the Course Registration System: i. Do the Requirement Analysis and Prepare SRS

1. **Draw E-R diagrams, DFD, CFD and structured charts for the project.**

**Aim:** ľo do Requiíement Analysis and Píepaíe SRS foí Couíse Registíation System. And also to díaw E-R diagíams, DFD, CFD and stíuctuíed Chaíts foí Couíse Registíation System.

**Problem Statement for Course Registration System:**

At the beginning of each semester students may request a course catalogue containing a list of course offerings for the semester. Information about each course, such as professor, department, and prerequisites will be included to help

students make informed decisions.

The new course registration system will allow students to select four course offerings for the coming semester. In addition, each student will indicate two alternative choices in case a course offering becomes filled or cancelled. Assuming perquisites are satisfied for each student. No course offering will have more than 30 students. No course offering will have fewer than 5 students. A course offering with fewer than 5 students will be cancelled.

Once the registration process is completed for a student, the registration system sends information to the billing system, so the student can be billed for the semester.

For each semester, there is a period of time that students can change their schedules. Students must be able to access the course registration system during this time to add or drop courses. The billing system will credit all students for courses dropped during this period of time. Professors must be able to access the course registration system to indicate which courses they will be teaching. They will also need to see which students signed up for their course offering.

#### Software Requirement Specification for Course Registration System :

* 1. Introduction
     1. Purpose:

The purpose of this SRS is to describe the requirements involved in developing a Course Registration System.

* + 1. Document Conventions:

The document is made using Calibri where:

* Main section are indicated with bold letters font 14
* Subsections also bolded with font 14.

An appendix A is added for acronyms and abbreviations

* + 1. Intended Audience and Reading Suggestions

This document is intended to be read by the customer. This is a technical document and the terms should be understood by the customer. The customer needs to understand this document fully so that they can draft a design document using this SRS presented to them by the analyst.

* + 1. Product Scope

The network based registration system will be used by students whom may be familiar or not to the online registration process thus the scope of the project must be user friendly for both students and admin.

This system will allow students with prerequisites for courses offered. Each student will select 4 courses for the semester, and indicate two alternative choices in case a course offering becomes filled or cancelled.

Once the registration process is completed for a student, the registration system sends information to the billing system so the student can be billed for the semester.

The system will have to keep a billing after the registration process has been completed. The communication and the monitoring of the application will be handled by the for a smooth registration process.

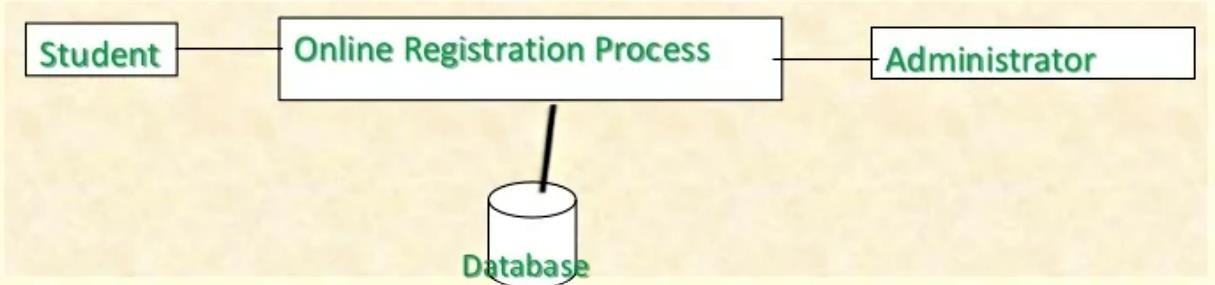
* + 1. References:

Sections of this document are based upon the IEEE Guide to Software Requirements Specification (IEEE 830).

* 1. Overall Description:
     1. Product Perspective:

This subsection should relate the requirements of that larger system to functionality of the software and should identify interfaces between that system and the software.

A block diagram showing the major components of the larger system, interconnections, and external inter-faces can be helpful.



The course registration system involves two actors, the administrator and the student. The administrator controls the communication and service delivery for student, the students communicate with the system through application for courses offered. The database system manages for application and financial information.

* + 1. Product functions:

This subsection of the SRS should provide a summary of the major functions that the software will perform.

The key features of this system can be abstracted as follows.

* Authentication through users' personal computer
* Alternative authentication mechanism for special conditions.
* Provide student with relevant courses for registration and a course catalogue.
* Registration process —check box method.
* Provide a proof of registration once the process has been submitted and terminated.
* Billing System generates a bill for student for that semester.
  + 1. User Classes and Characteristics:

This subsection of the SRS should describe those general characteristics of the intended users of the product including educational level, experience, and technical expertise.

Users are students and admin. Their characteristics:

* The user should be familiar with the Internet.
* User should be computer literate.
  + 1. Operating Environment:

Course registration system is an internet oriented application, it set to operate on a high available and Qos network, since registration is a sensitive thing, the quality of the network should be good for this process. Mostly a device that can access the internet and can support a huge web application for registration. Any OS can support this system as it is not particularly software or hardware dependent.

* + 1. Design and Implementation Constraints:

The system is internet based system, a webapp that should be developed to support any web browser to be used, registration in the University is a sensitive issue an must be secure , the system design should include a lot of DB and SYSTEM validation . Courses are billed by the University policy so this part must be secure ,so developers needs to be always there to support the delivered system in terms of validation and maintenance.

* + 1. User Documentation:

The project is available on the internet. The University website should provide a user manual on how to use an online registration application. Users of the system will be guided by the system all the way when registering.

* + 1. Assumptions and Dependencies:

The assumptions and dependencies relevant to the system are as follows.

* All users have an a computer or any web enable device
* User should have internet access.
* The user must have access or be on a reliable network.
* Users should have been admitted to the University, and have a student number. Users with prerequisites are given student number.
* User should read course catalogue careful in order not to make mistakes , cause they are un-reversible
* Besides the user catalogue to be given, users must know which course is expected of them to register.
* Users should view bill for the registered course.
  1. External Interface Requirements:
     1. User Interfaces:

The user is going to interact with the system through different interfaces. Listed below are the different components of user interfaces under their respective headings:

* student log-in / create profile
* Registration process.
* View and print
  + 1. Hardware Interfaces:

Minimum Hardware Requirements:

The system will run on different hardware gadgets. Below are the minimum hardware requirements for the smooth running of the system:

* 1GB RAM PC
* 1.8Hz processor
* 14" color monitor
* 120GB HDD CPU
* Proper running internet
  + 1. Software Interfaces:

Databases: Oracle

Operating systems : Any OS like Windows7, Linux, Mac OS Programming Language: Java

Front end: HTML, Java Script

* + 1. Communications Interfaces:

The main communication link that the system will be using is the internet.

* 1. System Features:
     1. System Feature:

The Course registration system comprises of two main features, namely, internet connectivity which will enables users to communicate with the server through a browser or web agent, and secondly the system requires database service to store the user's data. In a nutshell this system is web application and thus is only operational in an internet enabled environment.

* + 1. Functional requirements:

REQ-1: The system shall be interne oriented and require an online server. REQ-2: The system shall save the user's details to a remote database service REQ-3: The system shall allow users to register and to log in a user account

REQ-4: The system will allow users to register for courses following the University Rules.

REQ-5: The system shall automatically update the course catalog REQ-6: The system will allow users to view and print application and bill of applied courses.

* 1. Other Nonfunctional Requirements:-
     1. Peformance Requirements:

The system is required a fair amount of speed especially while browsing through the catalogue and presenting different possibilities for the schedule. The database shall be able to accommodate a minimum of 10000 records of students.

The software will support multiple users, with their respective accounts of course

* + 1. Safety Requirements:

The system is required provide a protection of the database of the university.

* + 1. Security Requirements:

The main security concern is for users account hence proper login mechanism should be used to avoid hacking. The online registration system shall not disclose personal information of students to unauthorized users or the public.

* + 1. Software Quality Attributes:-

Availability : Both catalogue and database will have to be available to students 24/7.

* + 1. Business Rules
* The online registration system shall include two types of accounts: the administrators and the students.
* To log in to the system user name and password is required. User name shall be the student number of the student and the password as they prefer.

Software Requirement Specification for Course Registration System Continuation

* 1. other Requirements No other requirements:-

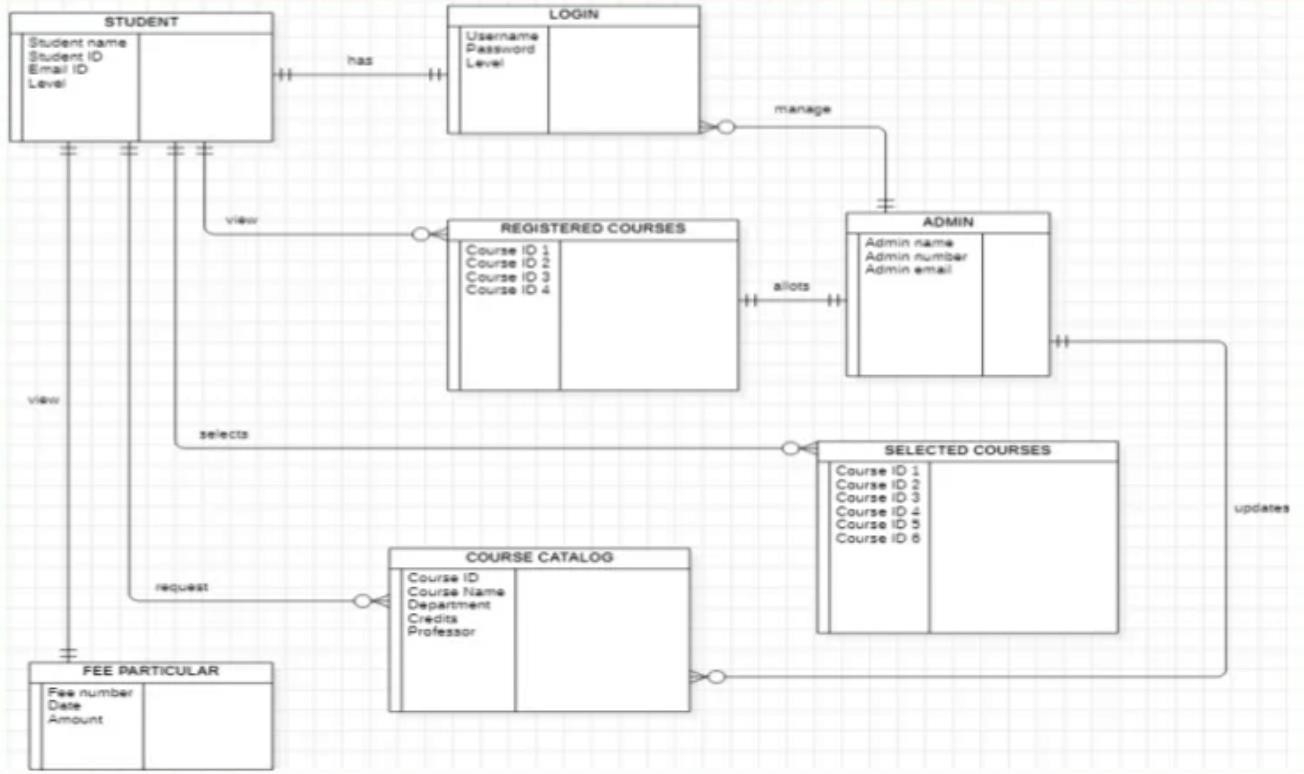
Appendix A: Glossary

* DB — database
* SQL — simple query language
* Admin —Administrator
* SRS —software requirements specification
* Webapp — web application
* ID — identity
* Std no: - student number
* RAM — random access memory
* PC — personal computer
* CPU — central processing unit
* OS — operating system

#### Entity-Relationship diagram for Course Registration System (ER diagram):

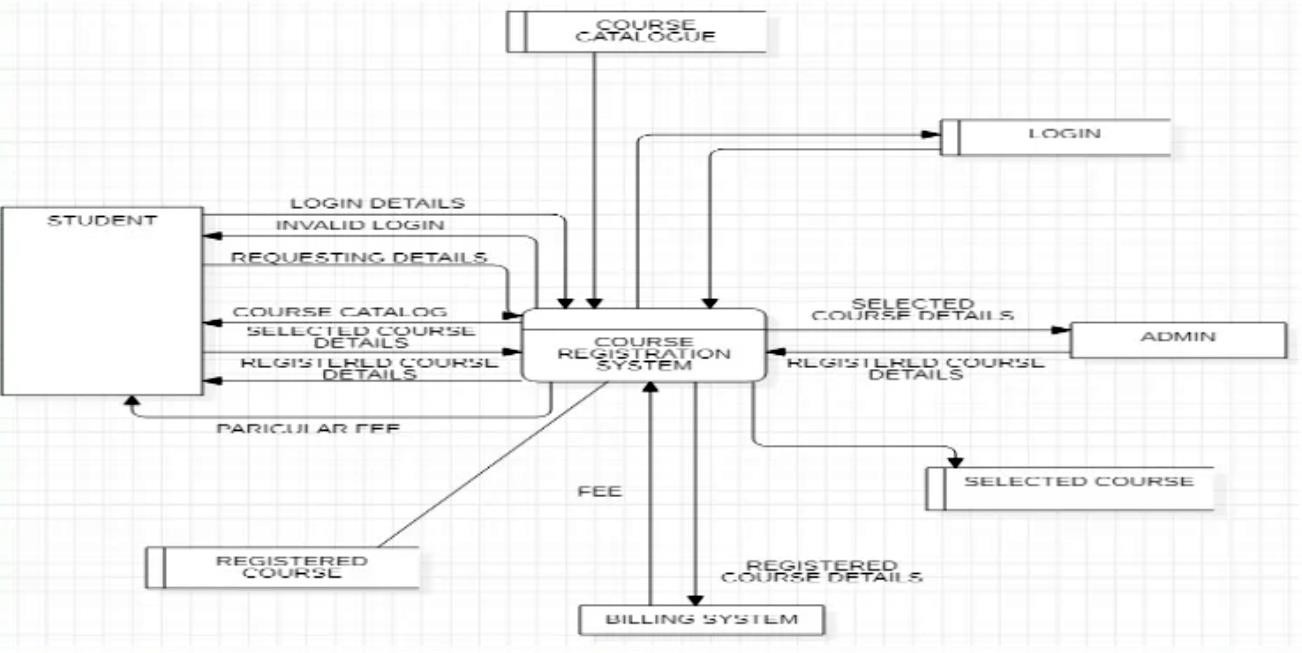
ER Diagram is a visual representation of data that describes how data is related to each other. In ER Model, we disintegrate data into entities, attributes and setup relationships between entities, all this can be represented visually using the ER diagram.

#### Entity-Relationship diagram for Course Registration System (ER diagram):

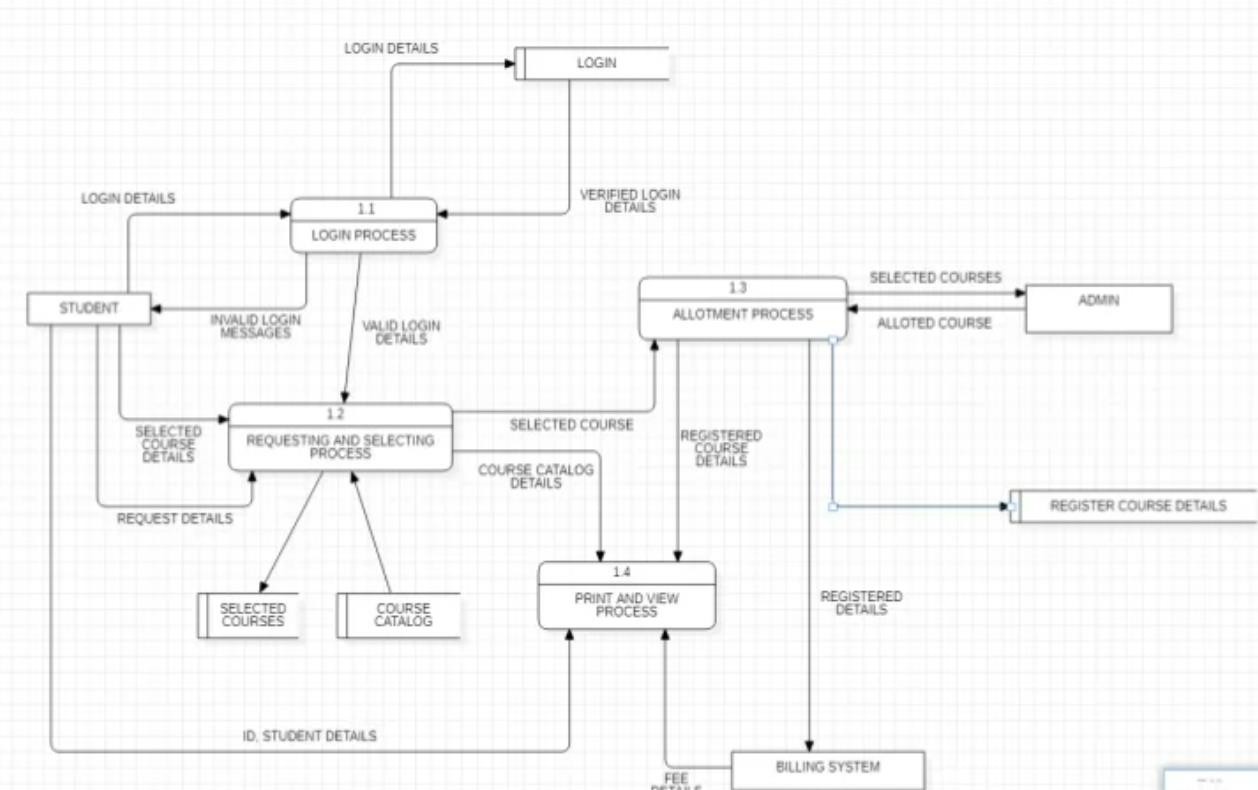


**Data Flow Diagram:** A data flow diagram (DFD) is a visual representation of the information flow through a process or system.

Context Level DFD (Level 0)

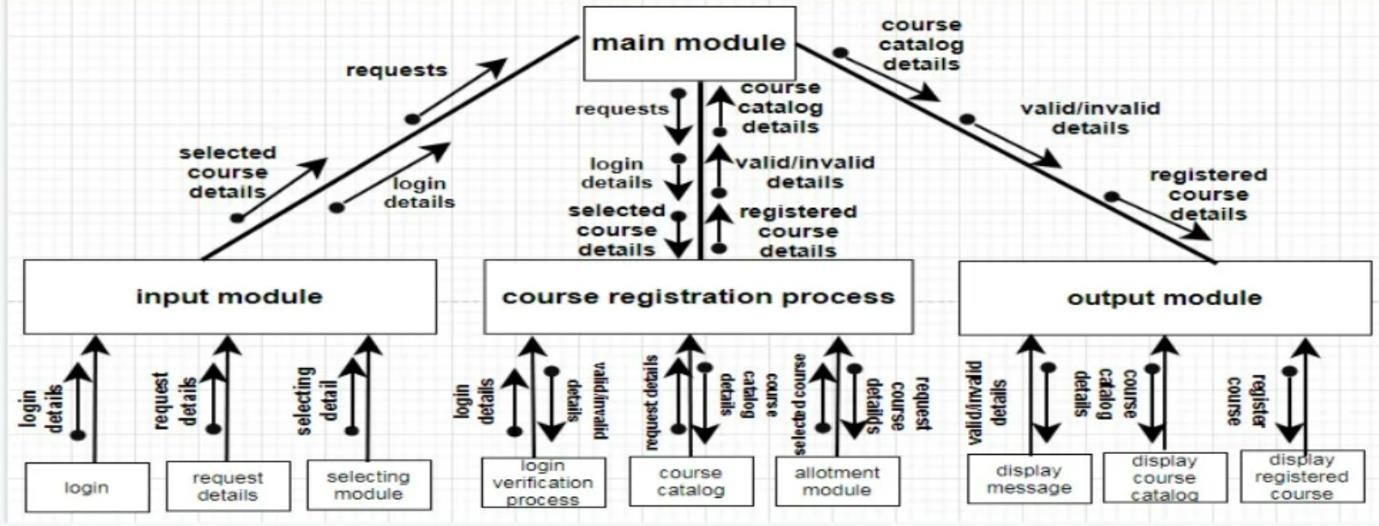


High Level DFD (Level 1)



Structured charts

A structure chart is a chart which shows the breakdown of a system to its lowest manageable levels.



# STUDENT MARKS ANALYZING SYSTEM

Perform the following for the Student Marks Analyzing System.

1. **Do the Requirement Analysis and Prepare SRS**
2. Draw E-R diagrams, DFD, CFD and structured charts for the project.

**Aim:** To do Requirement Analysis and Prepare SRS for Student Marks Analysing System. And also to draw E-R diagrams, DFD, CFD and structured Charts for Student Marks Analysing System.

Problem Statement for Student Marks Analyzing System: The Student Mark Analysis

System deals with the complete academic details of the students. It comprises of the Roll No, Name, Marks of each Subject, Total and Grade. It can be accessed by the faculty who alone can change or update the marks of the student. It is also the duty of the faculty to maintain the records, the duty of the administrator is to generate the report cards to the faculty members. The role of the Student is to just to view the progress report which consists of Roll No, Name, Marks of each Subject, Total and Grade with Pass or Fail Status.

#### Software Requirement Specification for Student Marks Analysing System

* 1. **Introduction**
     1. Purpose

The purpose of this SRS is to describe the requirements involved in developing a Student Marks Analysing Syste

* + 1. Document Conventions

The document is made using Calibri where:

* Main section are indicated with bold letters font 14
* Subsections also bolded with font 14.

An appendix A is added for acronyms and abbreviations 1.3.Intended Audience and Reading Suggestions

This document is intended to be read by the customer. This is a technical document and the terms should be understood by the customer. The customer needs to understand this document fully so that they can draft a design document using this SRS presented to them by the analyst.

#### Software Requirement Specification for Student Marks Analysing System Continuation

* 1. Product Scope

The Student Marks Analysing System will be used by Faculty, students and Admin. Thus the scope of the project must be user friendly for both Faculty, students and admin.

This system will allow Faculty to login and enter or update Marks of 4 subjects. It allows faculty to view student progress report. Admin is allowed to login and generate progress reports for all the students. This system allows the student only to view his Progress Report by giving his Roll Number.

* 1. References

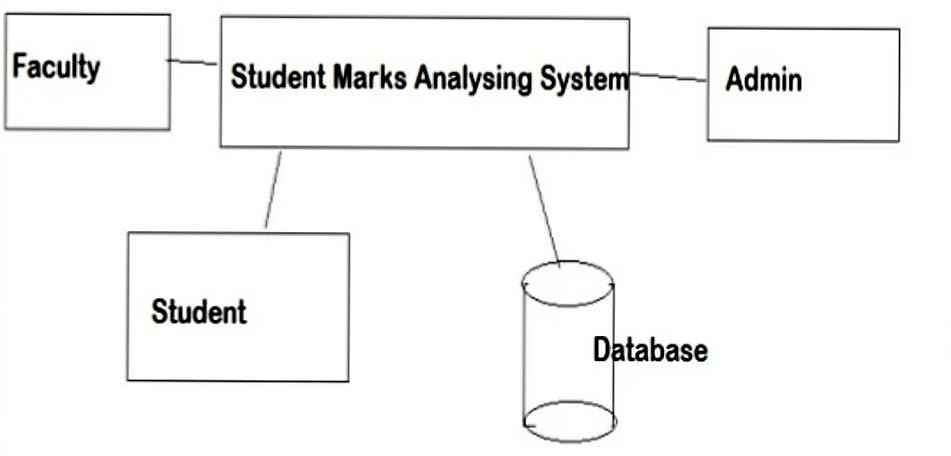
Sections of this document are based upon the IEEE Guide to Software Requirements Specification (IEEE 830)

* 1. Overall Description
     1. Product Perspective

This subsection should relate the requirements of that larger system to functionality of the software and should identify interfaces between that system and the software.

A block diagram showing the major components of the larger system, interconnections, and external inter-faces can be helpful.

#### Software Requirement Specification for Student Marks Analysing System Continuation



The Student Marks Analysing System involves three actors, the Faculty, the administrator and the student. Faculty updates or enter 4 subjects Marks of all students the administrator controls the communication and service delivery for student, the students communicate with the system through Roll No to view their Progress Report. The database system manages Login Credentials, Marks and Progress Reports.

#### Software Requirement Specification for Student Marks Analysing System Continuation

2.2 Product functions

This subsection of the SRS should provide a summary of the major functions that the software will perform. The key features of this system can be abstracted as follows. • Authentication through users' personal computer • Alternative authentication mechanism for special conditions. • Provide Faculty to enter/update Marks of all students. • Progress Report Generation by Admin - requesting through click • Provide student to view Progress Report

2.3. User Classes and Characteristics

This subsection of the SRS should describe those general characteristics of the intended users of the product including educational level, experience, and technical expertise.

Users are faculty, students and admin. Their characteristics:

* The user should be familiar with the Internet.
* User should be computer literate.

Software Requirement Specification for Student Marks Analysing System Continuation 2.4.Operating Environment

Student Marks Analysing System is an network oriented application, it set to operate on a high available and Qos network, since entering or updating marks is a sensitive thing, the quality of the network should be good for this process. Any OS can support this system as it is not particularly software or hardware dependent.

* 1. Design and Implementation Constraints

The system is intemet based system, a webapp that should be developed to support any web browser to be used, Entering marks or Updating marks of students by Faculty is a sensitive issue it must be secure , the system design should include a lot of DB and SYSTEM validation .

* 1. User Documentation

The project is available on the internet. The University website should provide a user manual on how to use an student marks analysing system. Users of the system will be guided by the system all the way to do their work.

Software Requirement Specification for Student Marks Analysing System Continuation

2.7 Assumptions and Dependencies

The assumptions and dependencies relevant to the system are as follows.

* All users have an a computer or any web enable device
* User should have internet access.
* The user must have access or be on a reliable network.
* Faculty, Student and Admin should have been admitted to the University, and have a Faculty id, Student id and Admin id.
* Faculty enters or updates 4 subject marks **3.External Interface Requirements** 3.1.User Interfaces

The user is going to interact with the system through different interfaces. Listed below are the different components of user interfaces under their respective headings:

* faculty log-in
* Entering or Updating marks process

#### Software Requirement Specification for Student Marks Analysing System Continuation

* Admin login
* Generating progress report process
* View and print progress report 3.2.Hardware Interfaces

Minimum Hardware Requirements:

The system will run on different hardware gadgets. Below are the minimum hardware requirements for the smooth running of the system:

* 1GB RAM PC
* 1.8Hz processor
* 14" color monitor
* 120GB HDD CPU
* Proper running internet

#### Software Requirement Specification for Student Marks Analysing System Continuation

3.3.Software Interfaces

Databases: Oracle

**Operating systems : Any OS like** Windows7, Linux, Mac OS Programming Language: Java

Front end: HTML, Java Script 3.4.Conmiunications Interfaces

The main communication link that the system will be using is the internet.

1. System Features
   1. System Feature

The Student Marks Analysing System comprises of two main features, namely, internet connectivity which will enables users to communicate with the server through a browser or web agent, and secondly the system requires database service to store the user's data In a nutshell this system is web application and thus is only operational in an internet enabled environment.

#### Software Requirement Specification for Student Marks Analysing System Continuation

* 1. Functional requirements

REQ-1: The system shall be internet oriented and require an online server.

REQ-2: The system shall save the user's (faculty, student and admin) details to a remote database service

REQ-3: The system shall allow faculty to log in into their account

REQ-4: The system will allow faculty to enter or update marks of all students. REQ-5: The system shall allow admin to login into their account.

REQ-6: The system will allow admin to generate Progress Report consisting of roll no, name, marks 1, marks2, marks3, marks4, total, grade and status which tells passtfail.

REQ-7: The system will allow student to view and print progress report by entering student id.

1. Other Nonfunctional Requirements
   1. Performance Requirements

The system is required a fair amount of speed especially while browsing. The database shall be able to accommodate a minimum of 10000 records of students. The software will support multiple users, with their respective accounts.

#### Software Requirement Specification for Students Marks Analysing System Continuation

* 1. Safety Requirements

It is required provide a protection for the marks database of the university. 5.3.Security Requirements

The main security concern is for users account hence proper login mechanism should be used to avoid hacking. The Student Marks analysing system shall not disclose personal information of students to unauthorized users or the public.

5.4.Software Quality Attributes

Availability : Progress reports can be viewed and printed at any time i.e 24/7 5.5.Business Rules

* The Student Marks Analysing System shall include two types of accounts: the administrator and the Faculty.
* To log in to the system user name and password is required. User name shall be the Faculty id and the password as they prefer. Same for admin login.

Software Requirement Specification for Students Marks Analysing System Continuation

1. **Other Requirements:** No other requirements. Appendix A: Glossary

DB — database

SQL— simple query language Admin —Administrator

SRS —software requirements specification Webapp — web application

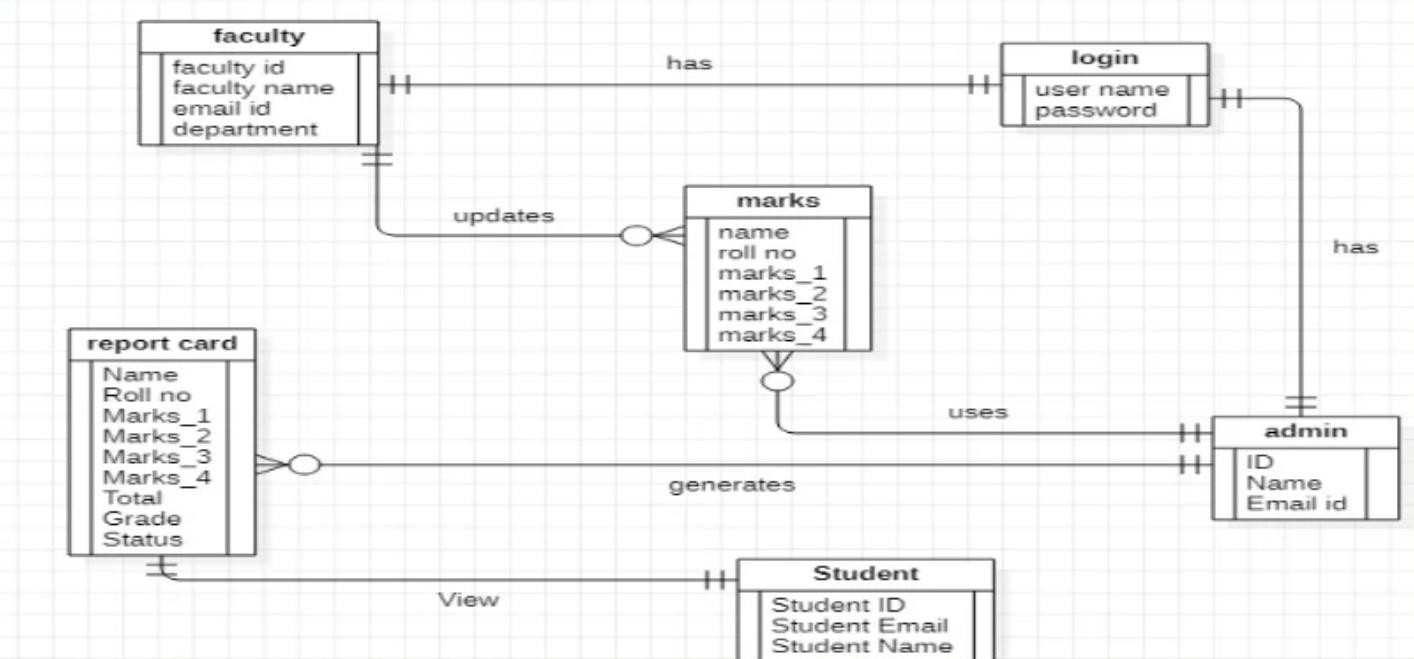
ID — identity

RAM — random access memory PC — personal computer

CPU — central processing unit OS — operating system

#### Entity Relationship Diagram for Student Marks Analysing System :

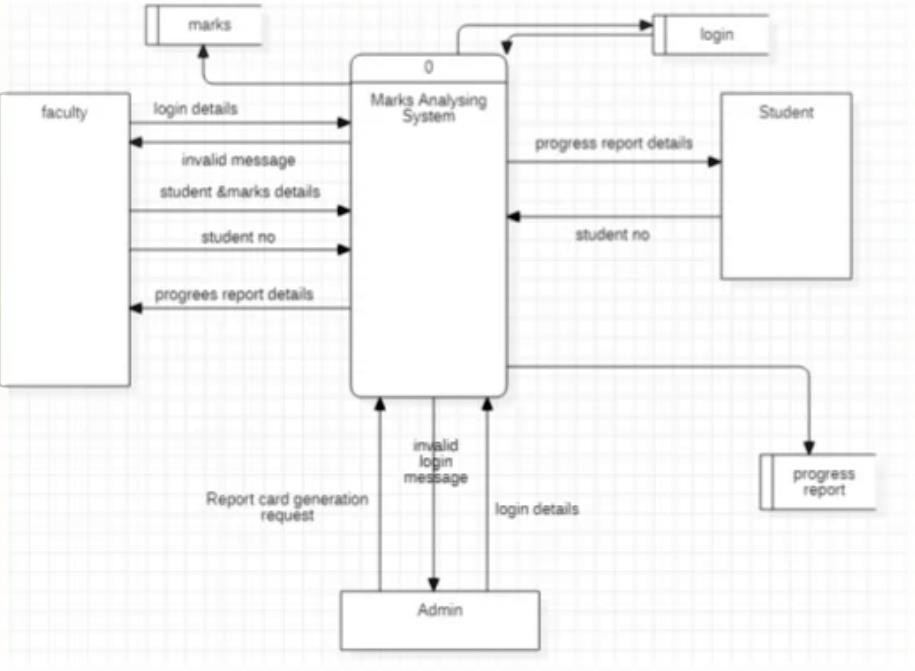
Entity is a visual representation of data that describes how data is related to each other. In ER model ,we disintegrate data into entities ,attributes and setup relationships between Entities all this can be represented visually using the ER Diagrams.



#### Data Flow Diagram for Student Marks Analysing System:

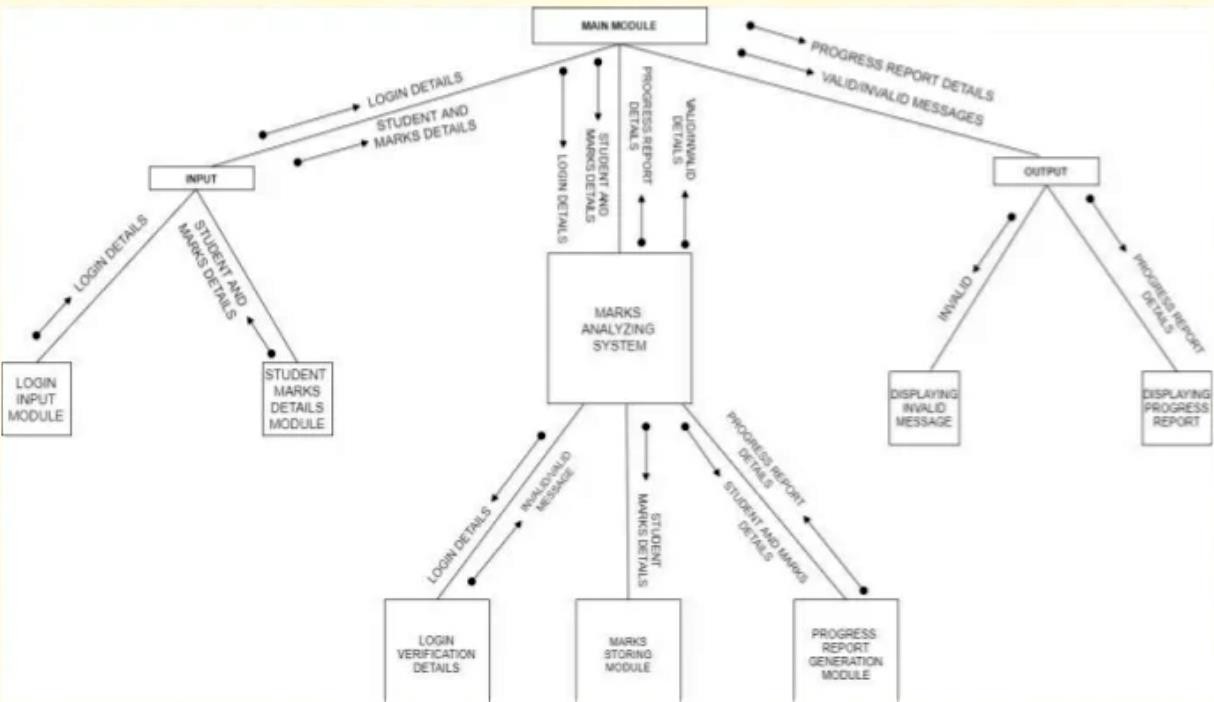
A data flow diagram (DFD) is a visual representation of the information flow through a process or system.

**Context Level DFD (Level 0)**



#### Structured charts for Student Marks Analyzing System

A structure chart is a chart which shows the breakdown of a system to its lowest manageable levels.



# ONLINE TICKET RESERVATION SYSTEM

**AIM:** To do Requirement Analysis and Prepare SRS for Online Ticket Reservation System. And also to draw E-R diagrams, DFD, CFD and structured Charts for Online Ticket Reservation System.

**PROBLEM STATEMENT:**

The “Any Where Any Time Advance Reservation” system is the online ticket reserving

system where the passengers can reserve the tickets for their travel, cancel the reserved ticket

and they can view the status of the ticket before travelling.

The passenger who is reserving the ticket in AWATAR unless they are the member of AWATAR .The passenger can reserve the ticket by giving the required detail in the form and submit it for the processing .They can reserve for at the max for five members and a single ticket is provided for them.

The administrator the AWATAR can control the ticket reservation and the accounts of

the passengers who are signing up in AWATAR. The administrator can reset the seats, fares of

the tickets, and generates the PNR number for the ticket that are reserved.

The sole control of the system is handled by the administrator. The printer prints the

tickets that are reserved by the passenger. The passenger can sign-up for only one time and he

can sign in for any number of times for reserving, cancelling and viewing the tickets.

The AWATAR system provides flexibility for the persons based on the age, the

passengers are fared based on the age and the place of travelling. This makes ease of use in using AWATAR system. The system tracks for the database any number of times for reserving, cancelling and status viewing.

The passenger can see the status of the classes that are available in the train which he is going to travel by noticing the number of seats details from the display board.

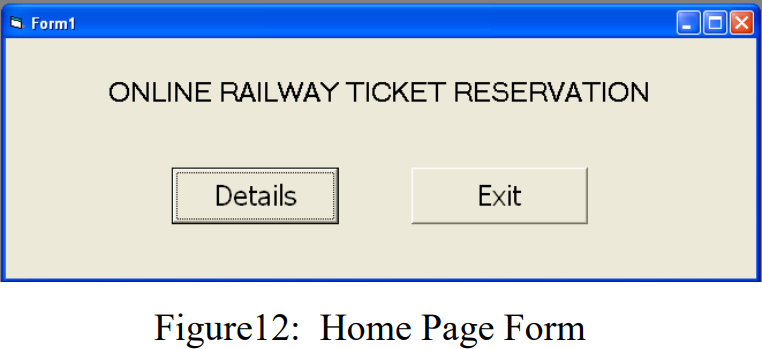
The cancellation of the tickets is also very easy so that the passenger can cancel the

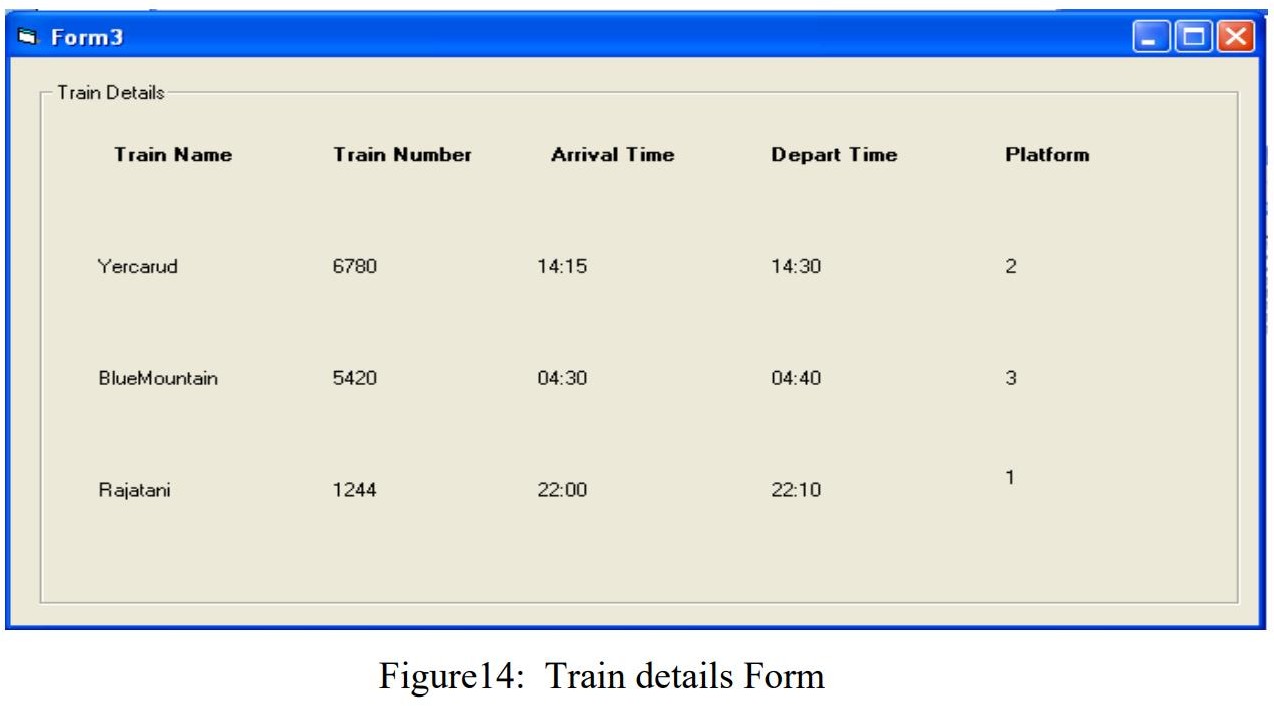
tickets that he has booked.

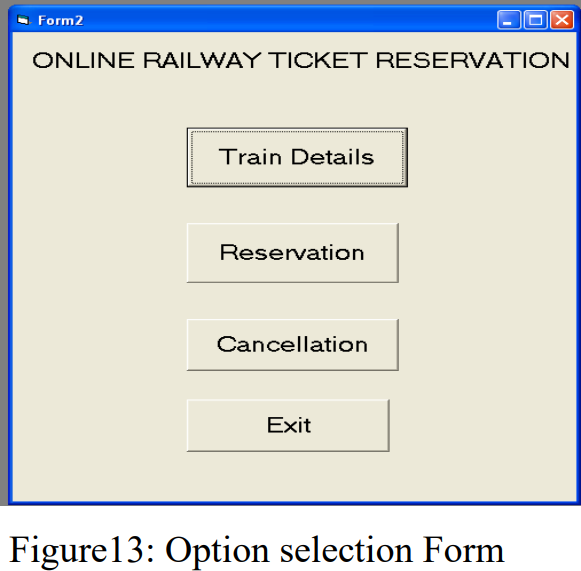
PROBLEM REQUIREMENTS:

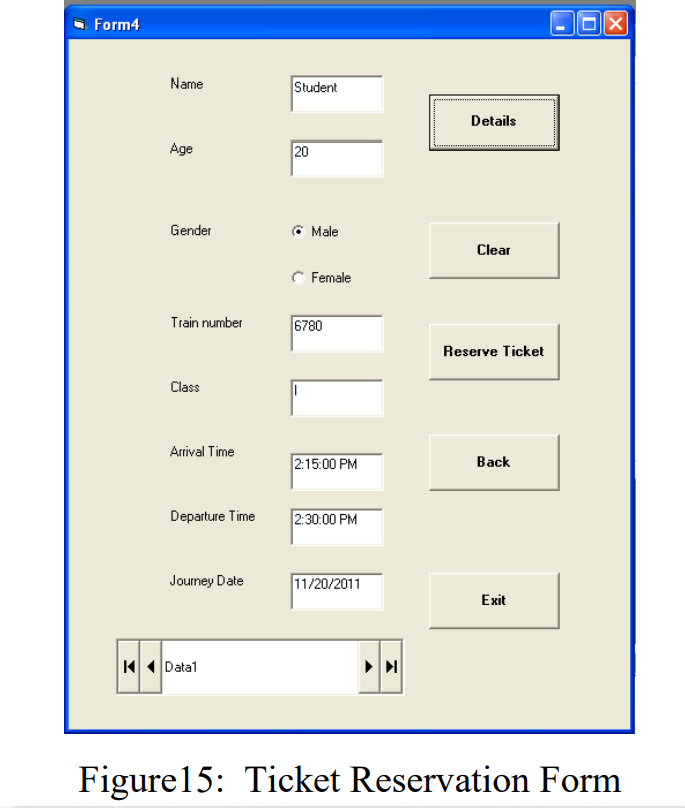
1. **Basic Requirements:**
2. Source place.
3. Destination place.
4. Date of journey.
5. **Functional Requirements:**
6. Details of passengers.
7. Details of the coach.
8. Pay the amount using visa card or debit card.
9. Give out the balance
10. Show the detail of ticket.
11. **Non-Functional Requirements:**
12. Trading system failure.
13. Unavailability of date.
14. Coach Unavailability
15. Insufficient amount for making payment.
16. Unavailability of berth.

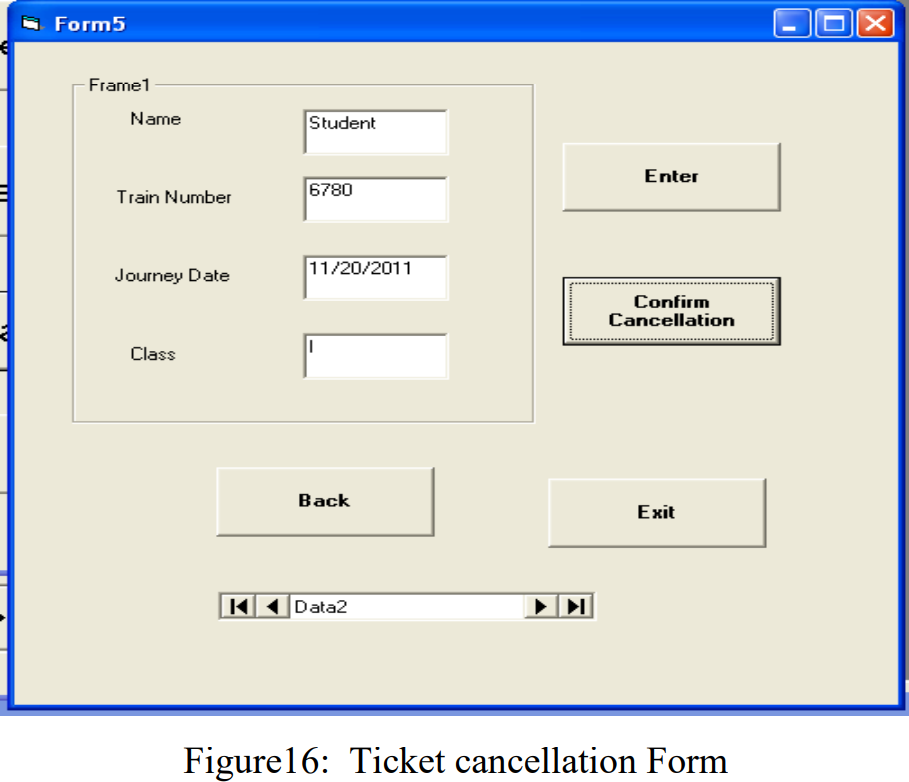
**SCREENSHOTS:**

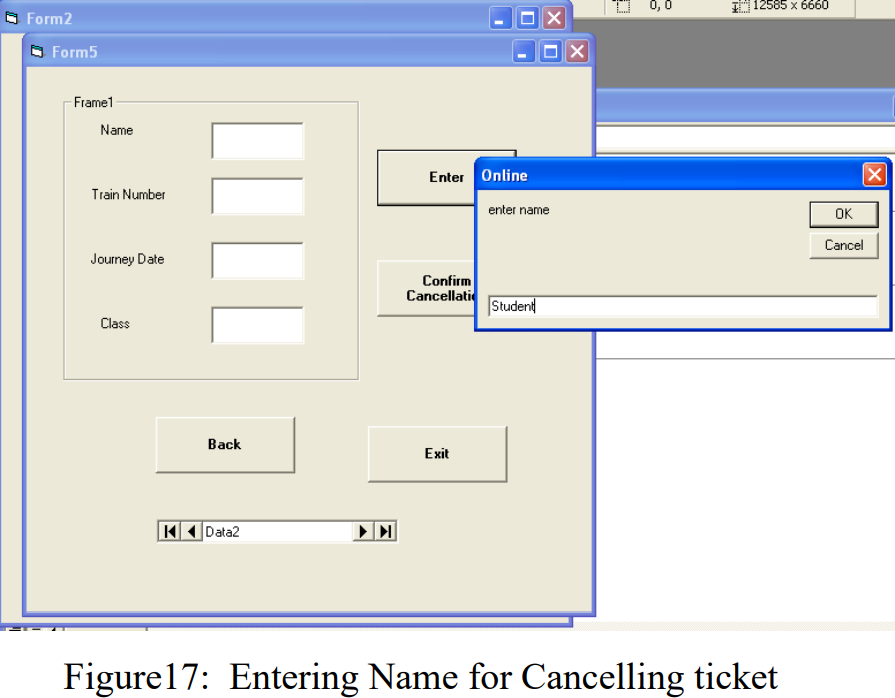












##### SOFTWARE TESTING:

TEST CASE REPORT:

TEST CASE NAME: Train number Availability OBJECTIVE: Usability Test

TEST CASES:

SCENARIO 1: Train Number: 6655 EXPECTED OUTPUT: Error Message **TEST RESULT:**

ACTUAL OUTPUT: Train number not available

##### CONCLUSION:

Thus the application on student mark list analysis system is developed using rational

rose and implemented using visual basic. The main aspects that are behind this application is

that they enabled us to bring out the new ideas that sustained within us for many days. This application enables the student to retrieve their student details and mark details at anywhere with a system.

# STOCK MAINTENANCE SYSTEM

##### Perform the following for the Stock Maintenance System

1. **Do the Requirement Analysis and Prepare SRS**

##### Draw E-R diagrams, DFD, CFD and structured charts for the project.

**Aim:** To do Requirement Analysis and Prepare SRS for Stock Maintenance System. And also to draw E-R diagrams, DFD, CFD and structured Charts for Stock Maintenance System.

##### Problem Statement for Stock Maintenance System:

Now a day's people should purchase things in stores. So the stores must be maintaining the product details and also the stocks. So in this stock maintenance have the details about the Product, Purchase, Sales and Stock.

The product details contain Product code, Product name, Opening Stock, Prices. These details are maintained in database. Purchase details contain the stock, quantity and also price. The Sales Details contain Date, Customer name, Product code, Quantity and Prices. The Stock Details contain product id, opening stock, purchase stock, sales stock and current stock.

##### Software Requirement Specification for Stock Maintenance System

1. Introduction
   1. Purpose

The purpose of this SRS is to describe the requirements involved in developing Online Stock Maintenance System.

* 1. Document Conventions

The document is made using Calibri where: Main section are indicated with bold letters font 14 Subsections also bolded with font 14.

An appendix A is added for acronyms and abbreviations 1.3.Intended Audience and Reading Suggestions

This document is intended to be read by the customer. This is a technical document and the terms should be understood by the customer. The designer needs to understand this document fully so that they can draft a design document using this SRS presented to them by the analyst.

##### Software Requirement Specification for Stock Maintenance System Continuation

* 1. Product Scope

The Online Stock Maintenance System will be used by customers, suppliers, managers and stock adminstrator who may be familiar or not familiar to the automated process thus the scope of the project must be user friendly for all stakeholders. This system will allow stock admin to manage and update stock of the products. Customer can view products and request to buy the products. Manager can view Stock details, sales details, purchase details, suppliers details, customer details and generate reports by giving date as input. Supplier receives purchase orders from the system and invoice is sent to the system.

* 1. References

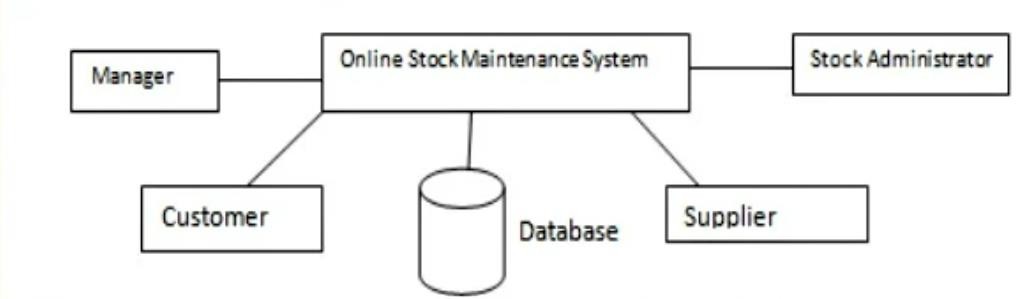
Sections of this document are based upon the IEEE Guide to Software Requirements Specification (IEEE 830)

##### Software Requirement Specification for Stock Maintenance System Continuation

1. Overall Description
   1. Product Perspective

This subsection should relate the requirements of that larger system to functionality of the software and should identify interfaces between that system and the software.

A block diagram showing the major components of the larger system, interconnections, and external inter-faces can be helpful.



##### Software Requirement Specification for Stock Maintenance System Continuation

The online Stock Maintenance system involves actors - the stock administrator, Manager, Customer and the Supplier. The stock administrator controls the communication and services. The database system manages for application and financial information. Manager communicates with system to get report on stock details, product details, supplier details, customer details, sales details and purchase order details.

2.2 Product functions

This subsection of the SRS should provide a summary of the major functions that the software will perform.

The key features of this system can be abstracted as follows.

-> Authentication through users' personal computer

-> Alternative authentication mechanism for special conditions.

-> Customer is provided with product list, from which he can select and buy products.

->Provide Manager Supplier details, product details, Stock details, Sales details, Purchase order details.

-> Provide supplier to receive purchase orders and send invoice

-> Provide Stock Admin to the manage and update stock, product details, supplier details etc.

##### Software Requirement Specification for Stock Maintenance System Continuation

* 1. User Classes and Characteristics

This subsection of the SRS should describe those general characteristics of the intended users of the product including educational level, experience, and technical expertise. Users are students and admin. Their characteristics: The user should be familiar with the Internet. User should be computer literate.

* 1. Operating Environment

The Online Stock Maintenance system is an intemet oriented application, it set to operate on a high available and Qos network. Mostly a device that can access the intemet and can support a huge web application is used for Sales and Purchases of products. Any OS can support this system as it is not particularly software or hardware dependent.

* 1. Design and Implementation Constraints

The system is intemet based system, a webapp that should be developed to support any web browser to be used. Customer Account, Manager account, Customer payment and payment for product purchase should be secure , the system design should include a lot of DB and SYSTEM validation. Developers needs to be always there to support the delivered system in terms of validation and maintenance.

##### Software Requirement Specification for Stock Maintenance System Continuation

* 1. User Documentation

The project is available on the internet. The Online Stock Maintenance System should provide a help on Product Sales, Product Purchases and maintenance of Product Stock. Users of the system will be guided by the system all the way when using.

2.7 Assumptions and Dependencies

The assumptions and dependencies relevant to the system are as follows.

-> All users have an a computer or any web enable device

-> User should have internet access.

-> The user must have access or be on a reliable network.

-> Registered Customers and Registered Suppliers can login into the system.

##### Software Requirement Specification for Stock Maintenance System Continuation

1. External Interface Requirements
   1. User Interfaces

The user is going to interact with the system through different interfaces. Listed below are the different components of user interfaces under their respective headings:

-> Customer log-in

-> Manager log-in

-> Supplier log-in

-> Stock Admin log-in

-> Product Sales process –

> Product Purchase process

-> Product Stock Maintenance/updation process

-> View and Print process

* 1. Hardware Interfaces Minimum Hardware-Requirements:

The system will run on different hardware gadgets. Below are the minimum hardware requirements for the smooth running of the system:

-> 1GB RAM PC

-> 1.8Hz processor

-> 14" color monitor

-> 120GB HDD CPU

-> Proper running internet

##### Software Requirement Specification for Stock Maintenance System Continuation

* 1. Software Interfaces Databases: Oracle

Operating systems : Any OS like Windows7, Linux, Mac OS Programming Language: Java

Front end: HTML, Java Script 3.4.Conununications Interfaces

The main communication link that the system will be using is the internet.

1. System Features
   1. System Feature

The Online Stock Maintenance System comprises of two main features, namely, internet connectivity which will enables users to communicate with the server through a browser or web agent, and secondly the system requires database service to store the user's data. In a nutshell this system is web application and thus is only operational in an internet enabled environment.

##### Software Requirement Specification for Stock Maintenance System Continuation

* 1. Functional requirements

REQ-1: The system shall be internet oriented and require an online server.

REQ-2: The system shall save the product details, customer details, supplier details, sales details, purchase details and stock details in remote database.

REQ-3: The system shall allow to customer to log in and buy the product i.e sales process. REQ-4: The system will allow manager to view and print product details, customer details, supplier details, sales details, purchase details and stock details.

REQ-5: The system will allow to send purchase order to Supplier and receives Invoice.

REQ-6: The system will allow admin to manage and update product details, stock details, customer details and supplier details.

1. Other Nonfunctional Requirements
   1. Performance Requirements

The system is required a fair amount of speed especially while browsing through the products, selling and purchasing products

The database shall be able to accommodate a minimum of 100000 records of stock, cutomers and suppliers.

The software will support multiple users.

##### Software Requirement Specification for Stock Maintenance System Continuation

* 1. Safety Requirements

The system is required provide a protection of the database.

* 1. Security Requirements

The main security concern is for payment details during sales and purchase process, hence proper login mechanism should be used to avoid hacking. The online Stock Maintenance system shall not disclose personal information of passengers to unauthorized users or the public.

* 1. Software Quality Attributes Availability :

The database will have to be available to passengers 24/7.

* 1. Business Rules

-> The online ticket reservation system shall include four types of accounts: the stock administrator, manager, supplier and the customers.

-> To log in to the system user name and password is required.

-> Manager will take decision when to purchase stock by observing demand of the product.

##### Software Requirement Specification for Stock Maintenance System Continuation

1. Other Requirements No other requirements. Appendix A: Glossary

DB — database

SQL— simple query language Admin —Administrator

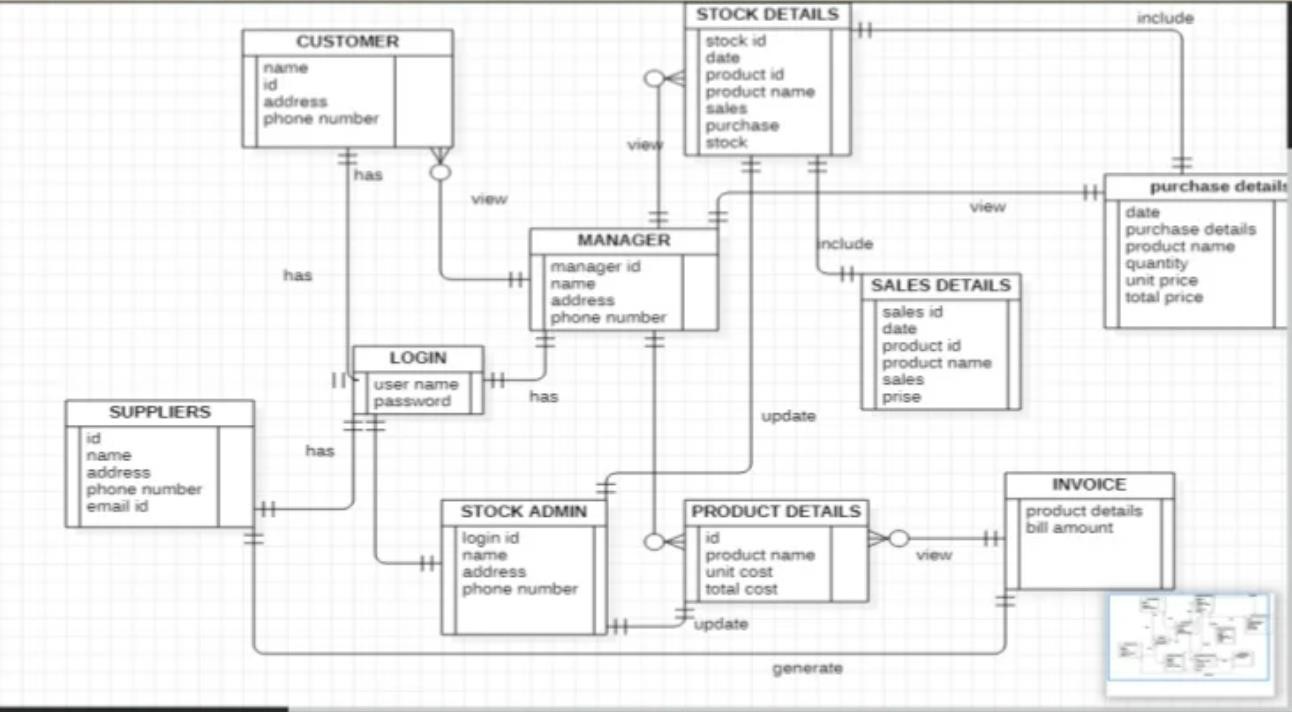
SRS —software requirements specification Webapp — web application

RAM — random access memory PC — personal computer

CPU — central processing unit OS — operating system

##### Entity-Relationship diagram for Stock Maintenance System (ER diagram)

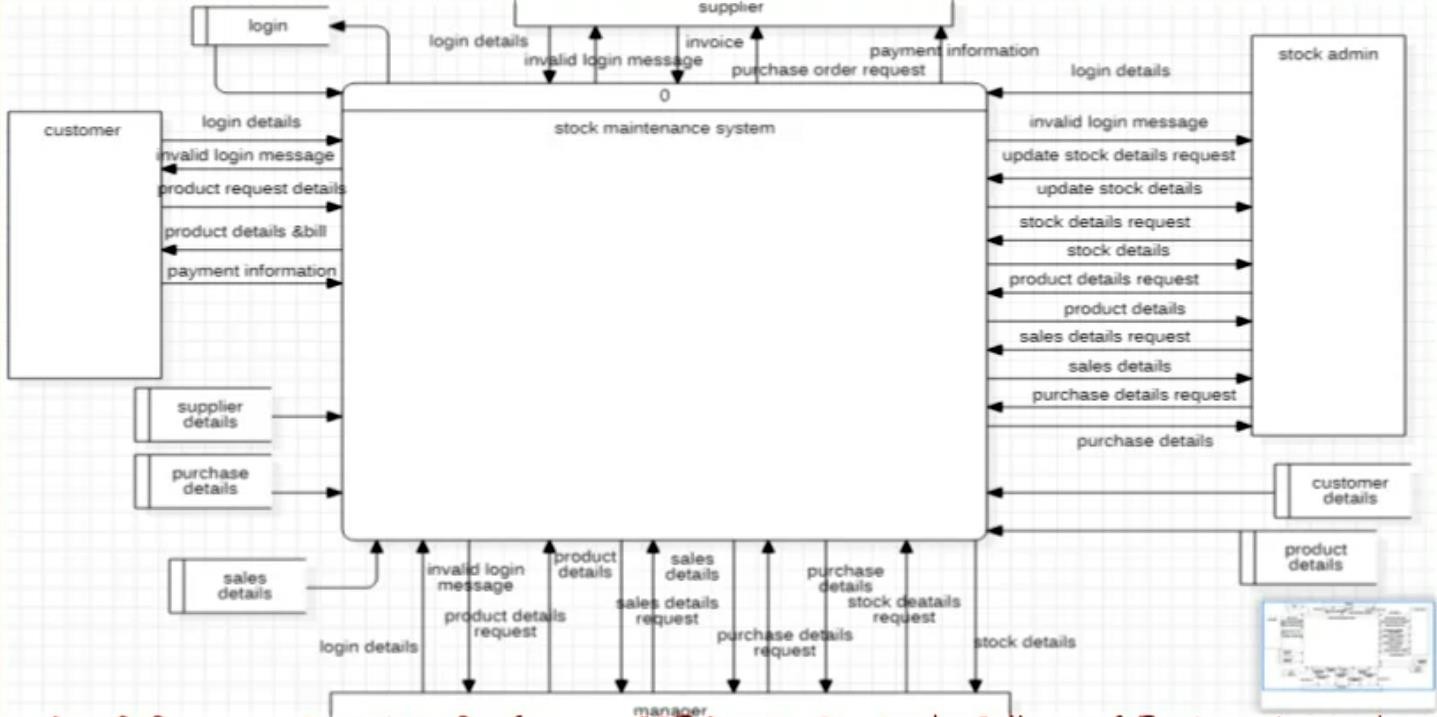
ER Diagram is a visual representation of data that describes how data is related to each other. In ER Model, we disintegrate data into entities, attributes and setup relationships between entities, all this can be represented visually using the ER diagram.



##### Data Flow Diagram(DFD) for Stock Maintenance System :

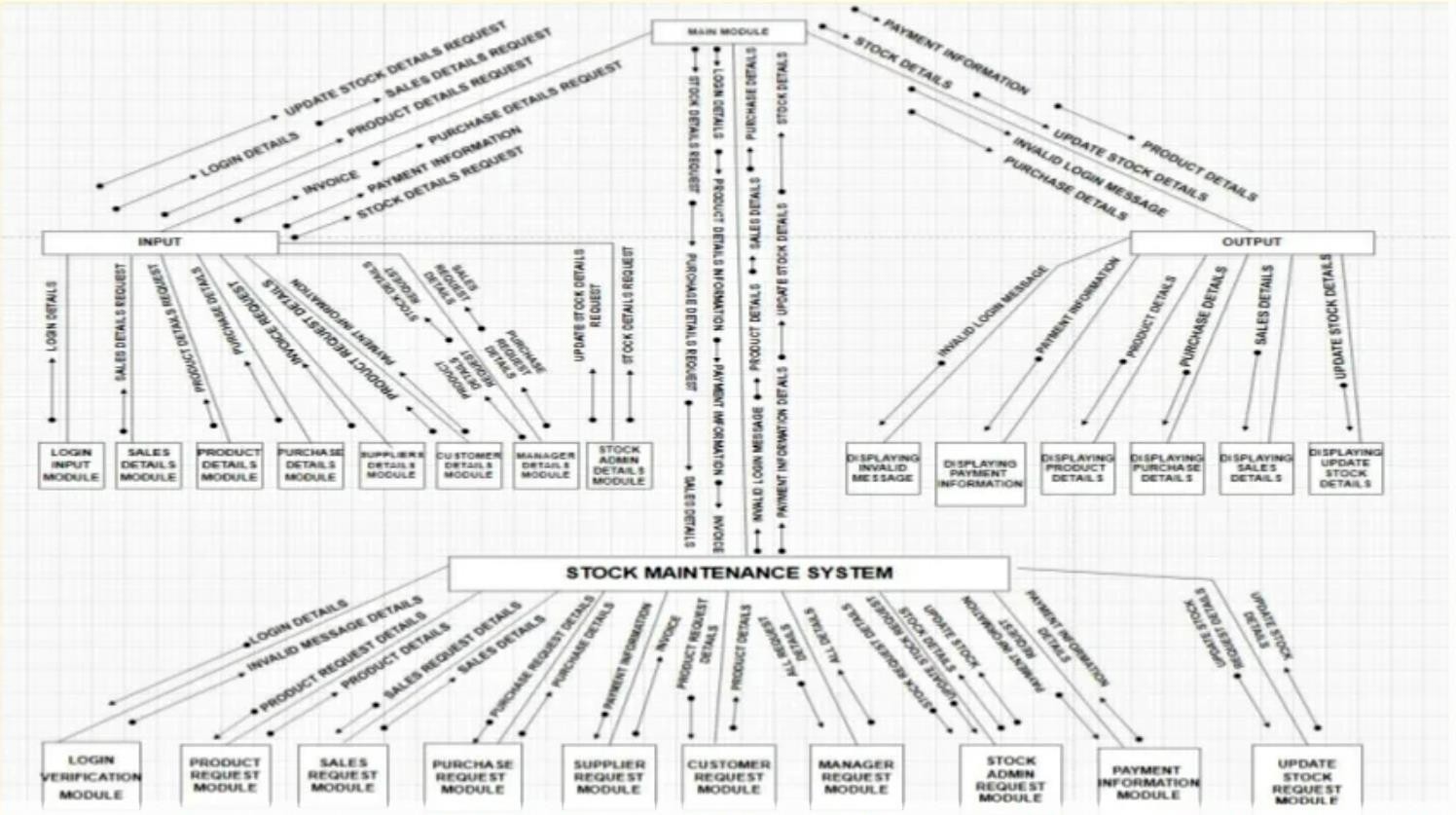
A data flow diagram (DFD) is a visual representation of the information flow through a process or system.

Context Level DFD (Level 0)



##### Structured charts for Stock Maintenance System:

A structure chart is a chart which shows the breakdown of a system to its lowest manageable levels.



# COCOMO MODEL

### Consider any application, using COCOMO model, estimate the effort.

**Aim:** Estimating Effort using COCOMO model

##### COCOMO Model

COCOMO (Constructive Cost Model) is an algorithmic cost estimation technique proposed by Boehm, which works in a bottom-up manner. It is designed to provide some mathematical equations to estimate software projects. These mathematical equations are based on historical data and use project size in the form of KLOC.

The COCOMO model uses a multivariable size estimation model for effort estimation. A multivariable model depends on several variables, such as development environment, user involvement, memory constraints, technique used, etc. COCOMO estimation is a family of hierarchical models, which includes

* Basic,
* Intermediate, and
* Detailed COCOMO models.

Each of the models initially estimates efforts based on the total estimated KLOC.

##### Basic COCOMO Model

The basic COCOMO model estimates effort in a function of the estimated KLOC in the proposed project.

The basic COCOMO model is very simple, quick, and applicable to small to medium organic-type projects. It is given as follows:

Development effort (E) = a x (KLOC) b Development time (T) = c x (E) d

Where a, b, c, and d are constants and these values are determined from the historical data of the past projects.

The development time (T) is calculated from the initial development effort (E).

Boehm's definition of Organic, Semid-detached, and Embedded systems are elaborated below:

**Organic:** A development project can be considered as organic type, if the project deals with developing a well understood application program.

**Semidetached:** A development project can be considered as Semidetached type, if the development consists of a mixture of experience and inexperience in developing the project.

**Embedded:** A development project can be considered as embedded-type, if the software being developed is strongly coupled to complex hardware.

The values of a, b, c and d for organic, semidetached, and embedded-type projects are shown in Table.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Software Projects | a | b | c | D |
| Organic | 2.4 | 1.05 | 2.5 | 0.38 |
| Semi Detached | 3.0 | 1.12 | 2.5 | 0.35 |
| Embedded | 3.6 | 1.20 | 2,5 | 0.32 |

##### Intermediate COCOMO Model

Boehm has introduced 15 cost drivers, considering the various aspects of product development environment. These cost drivers are used to adjust the project complexity for estimation of effort and these are termed as effort adjustment factors (EAF).

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Cost Drivers | Very Low | Low | Normal | High | Very High |
| Product Aattibutes |  |  |  |  |  |
| Rquired Software Reliability | 0.75 | 0.88 | 1.00 | 1.15 | 1.40 |
| Size of Application Database |  | 0.94 | 1.00 | 1.08 | 1.16 |
| Complexity of The Product | 0.70 | 0.85 | 1.00 | 1.15 | 1.30 |
| Hardware Attributes |  |  |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Runtime Perform Ance Constraints |  |  | 1.00 | 1.11 | 1.30 |
| Memory Constraints |  |  | 1.00 | 1.06 | 1.21 |
| Voladity of the virtual machine environment |  | 0.87 | 1.00 | 1.15 | 1.30 |
| Required turnabout time |  | 0.94 | 1.00 | 1.07 | 1.15 |
| Personal attributes |  |  |  |  |  |
| Analyst capability | 1.46 | 1.19 | 1.00 | 0.86 | 0.71 |
| Applications experience | 1.29 | 1.13 | 1.00 | 0.91 | 0.82 |
| Software engineer capability | 1.42 | 1.17 | 1.00 | 0.86 | 0.70 |
| Virtual machine  experience | 1.21 | 1.10 | 1.00 | 0.90 |  |
| Programming language experience | 1.14 | 1.07 | 1.00 | 0.95 |  |
| Project Attributes |  |  |  |  |  |
| Application of software engineering methods | 1.24 | 1.10 | 1.00 | 0.91 | 0.82 |
| Use of software tools | 1.24 | 1.10 | 1.00 | 1.91 | 0.83 |
| Required development schedule | 1.23 | 1.08 | 1.00 | 1.04 | 1.10 |

These cost drivers are classified as computer attributes, product attributes, project attributes, and personnel attributes.

The intermediate COCOMO model computes software development effort as a function of the program size and a set of cost drivers.

The intermediate COCOMO model estimates the initial effort using the basic COCOMO model. Then the EAF is calculated as the product of 15 cost drivers.

Total effort is determined by multiplying the initial effort with the total value of EAF. The computation steps are summarized below.

Development effort (E):

Initial effort (Ei) = a x (KLOC)^b EAF= EAF I x EAF 2 x... x EAF n

Total development effort (E)= Ei x EAF

Development(T) = c \* (E)^d

##### Intermediate Model utilizes 15 such drivers for cost estimation. Classification of Cost Drivers and their attributes:

Hhhh

**(i) Product attributes**

* Required software reliability extent
* Size of the application database
* The complexity of the product

**(ii) Hardware attributes**

* Run-time performance constraints
* Memory constraints

**(iii) Product attributes**

* Analyst capability
* Software engineering capability
* Application experience
* Virtual machine experiance

**(iv) Project attributes**

* Use of software tools
* Appliication of software

**Intermediate COCOMO a and b values**

|  |  |  |
| --- | --- | --- |
| Software Projects | A | b |
| Organic | 3.2 | 1.05 |
| Semi Detached | 3.0 | 1.12 |
| Embedded | 2.8 | 1.20 |

Example 1:

Assume that a system for simple student registration in a course is planned to be developed and its estimated size is approximately 10,000 lines of code. The organization is proposed to pay Rs. 25000 per month to software engineers. Compute the development effort, development time, and the total cost for product development.

Solution :

The project can be considered an organic project. Thus, from the basic COCOMO model, Development effort (E) = 2.4 x (10)1.08= 26.92 PM

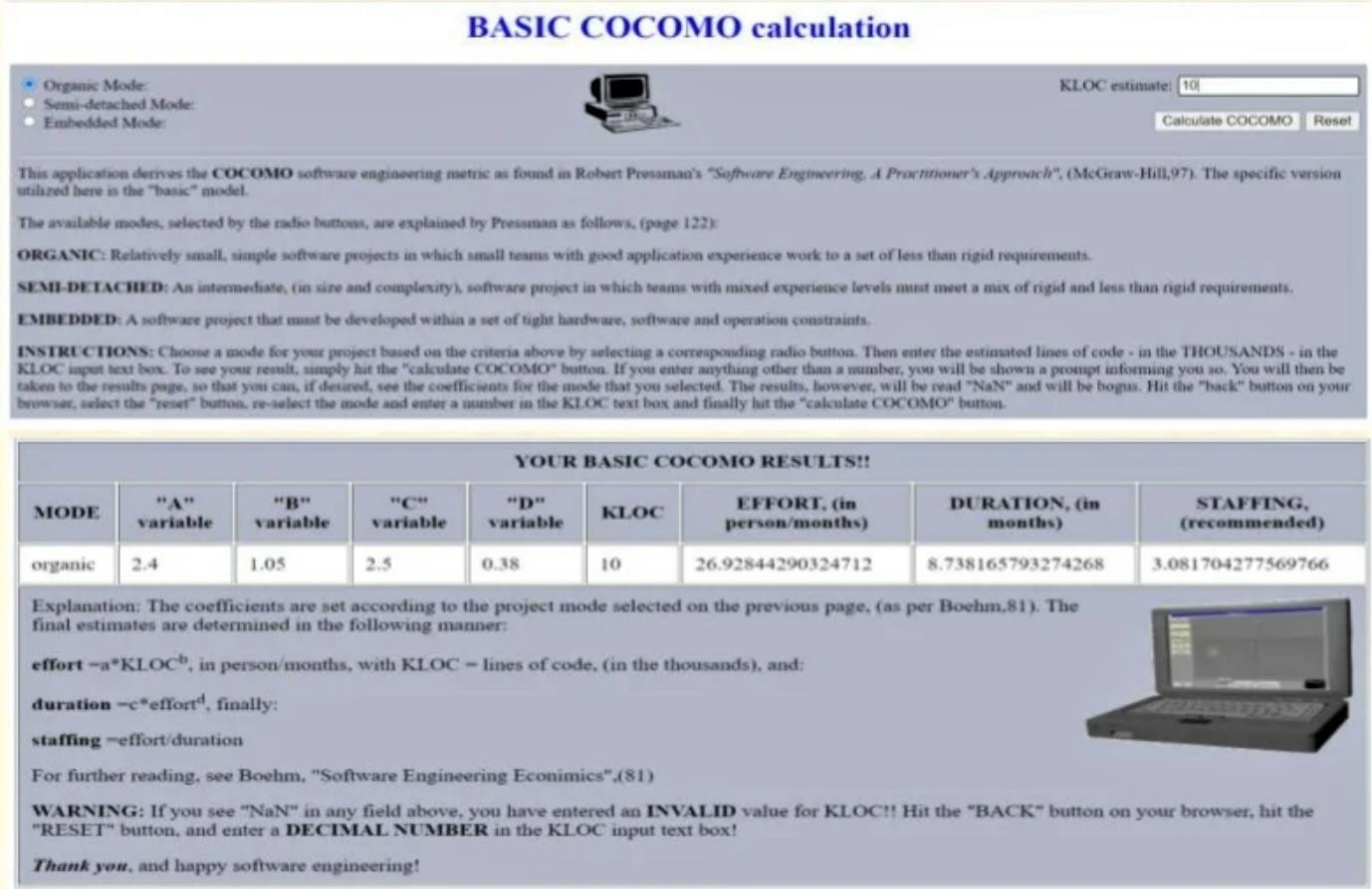
Development time (T) = 2.5 x (26.92)038 = 8.725 months Staffing (S) = E/T = 26.92 / 8.725 =3.085 persons

Total product development cost = Development time x Salaries of engineers

= 8.725 x 25000

= Rs. 2,18,125

OUTPUT:



Example 2:

Suppose a library management system (LMS) is to be designed for an academic institution. From the project proposal, the following five major components are identified:

|  |  |  |
| --- | --- | --- |
| Online data entry | - | 1.0 KLOC |
| Data update | - | 2.0 KLOC |
| File input and output | - | 1.5 KLOC |
| Library reports | - | 2.0 KLOC |
| Query and search | - | 0.5 KLOC |

The database size and application experience are very important in this pmject. The use of the software tool and the main storage is highly considerable. The virtual machine experience and its volatility can be kept low. All other cost drivers have nominal requirements. Use the COCOMO model to estimate the development effort and the development time.

**Solution:** The LMS project can be considered an organic category project. The total size of the modules is 7 KLOC. The development effort and development time can be calculated as follows:

##### Development effort

Initial effort (Ei) = 3.2 x (7)1.05=3.2\*7.71=24.672 PM

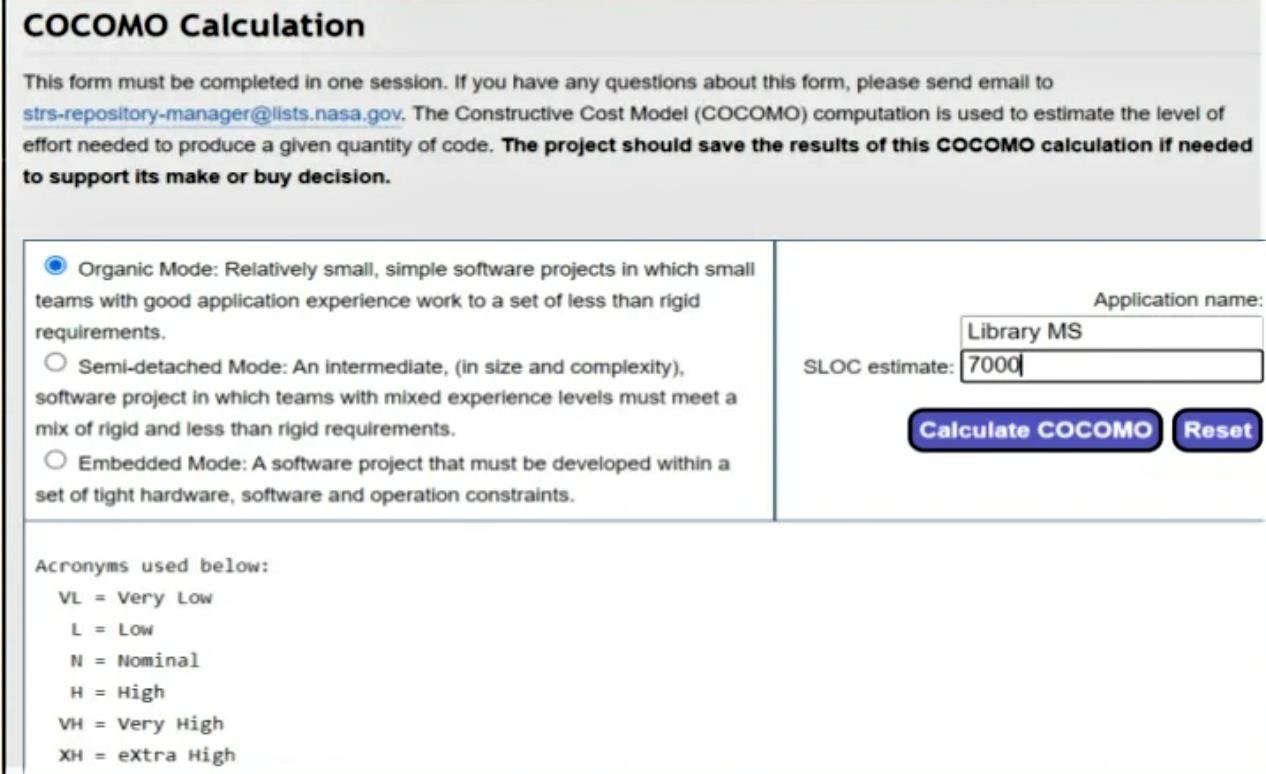
EAF = 1.16\*0.82\*0.83\*1.21\*1.21\*0.87=1.0056

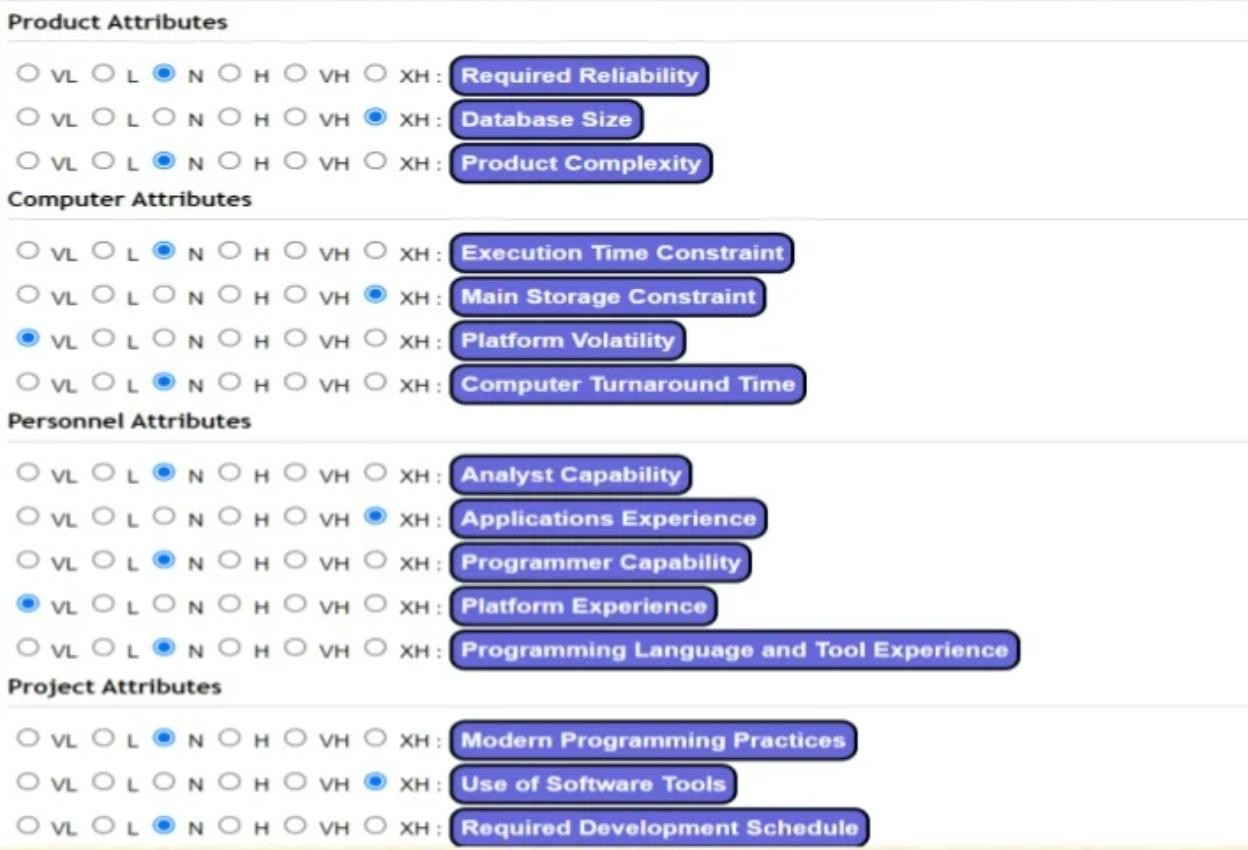
Total Effort (E) = 24.672\*1.0056 =24.81 PM

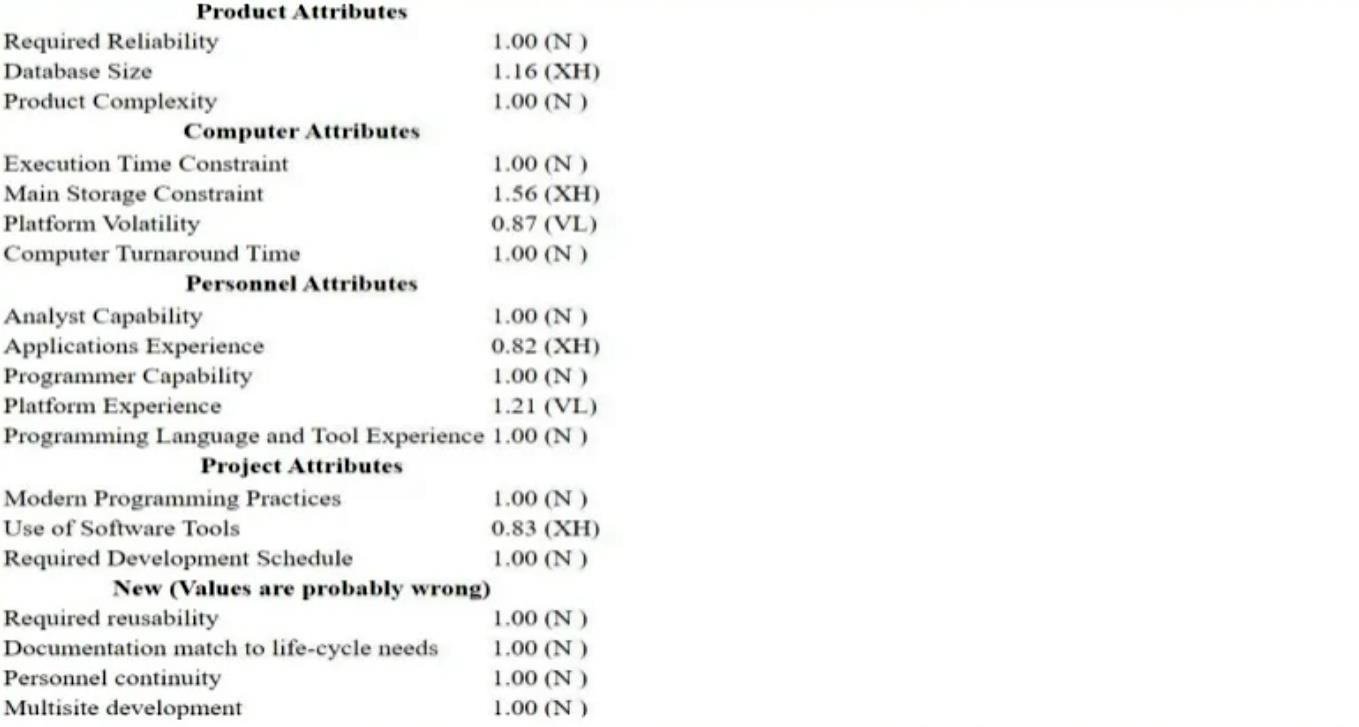
**Development time (T)** = 2.5 x 038 month =2.5 \*(24.81) 0.38 month

=2.5 \*3.388 =8.47 months

**Staffing (S)** = E/T = 24.81 / 8.47 = 2.929 persons.







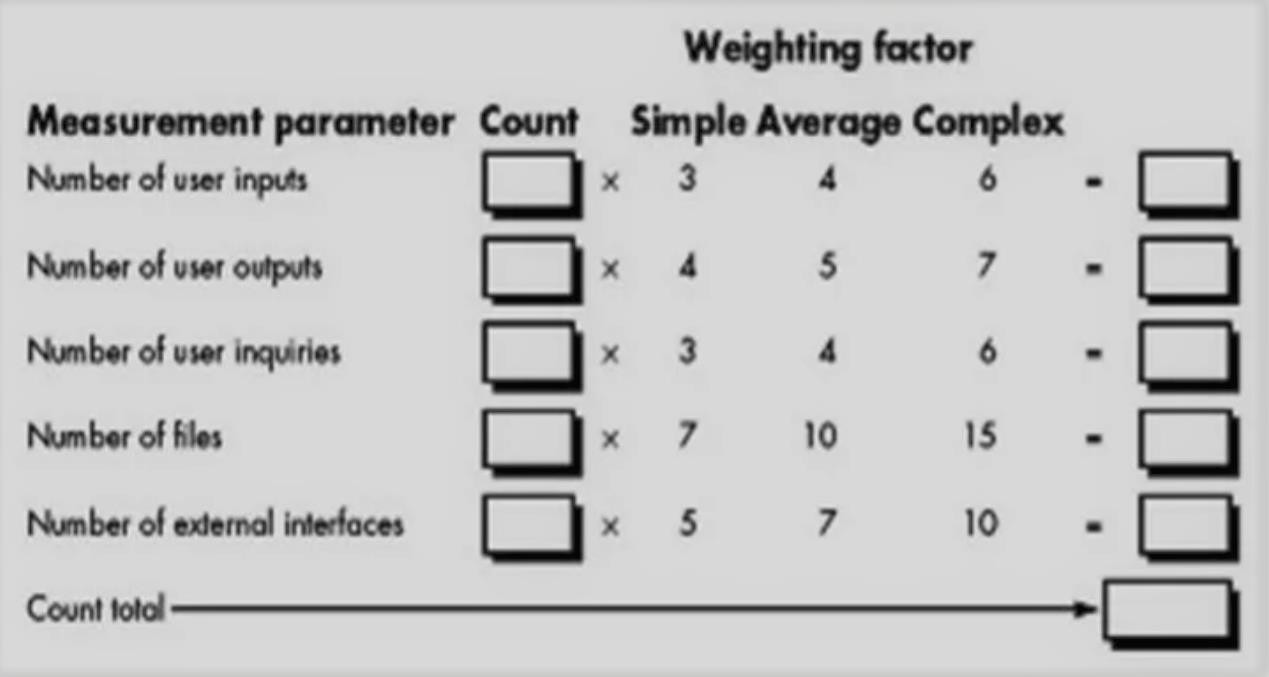
***FP Oriented Estimation***

##### Consider any application, Calculate effort using FP oriented estimation model.

**Aim:** To calculate FP of an application and calculate effort using FP oriented estimation model.

##### INTRODUCTION

* + Function-oriented software metrics use a measure of the functionality delivered by the application as a normalization value. Since 'functionality' cannot be measured directly, it must be derived indirectly using other direct measures
  + Function-oriented metrics were first proposed by Albrecht, who suggested a measure called the function point. Function points are derived using an empirical relationship based on countable (direct) measures of software's information domain and assessments of software complexity.
  + Function points are computed by completing the table as shown below. Five information domain characteristics are determined and counts are provided in the appropriate table location. Information domain values are defined in the following manner:



**Number of user inputs:** Each user input that provides distinct application oriented data to thesoftware is counted. Inputs should be distinguished from inquiries, which are counted separately.

**Number of user outputs:** Each user output that provides application oriented information tothe user is counted. In this context output refers to reports, screens, error messages, etc. Individual data items within a report are not counted separately.

**Number of user inquiries:** An inquiry is defined as an on-line input that results in the generation of some immediate software response in the form of an on-line output. Each distinct inquiry is counted.

**Number of files:** Each logical master file (i.e., a logical grouping of data that may be one part of a large database or a separate file) is counted

**Number of external interfaces:** All machine readable interfaces (e.g., data files on storagemedia) that are used to transmit infomation to another system are counted.

Once these data have been collected, a complexity value is associated with each count. Organizations that use function point methods develop criteria for determining whether a particular entry is simple, average, or complex. Nonetheless, the determination of complexity is somewhat subjective.

##### FUNCTION POINTS(FP) COMPUTATION

To compute function points (FP), the following relationship is used:

FP count total\* [0.65+0.01 Σ (Fi)]

where count total is the sum of all FP entries.

The Fi (i = 1 to 14) are "complexity adjustment values" based on responses to the following questions

1. Does the system require rel iable backup and recovery?
2. Are data communications required?
3. Are there distributed processing functions?
4. Is performance critical?
5. Will the system run in an existing, heavily utilized operational environment?
6. Does the system require on-line data entry?
7. Does the on-line data entry require the input transaction to be built over multiple screens or operations?
8. Are the master files updated on-line?
9. Are the inputs, outputs, files, or inquiries complex?
10. Is the internal processing complex?
11. Is the code designed to be reusable?
12. Are conversion and installation included in the design?
13. Is the system designed for multiple installations in different organizations?
14. Is the application designed to facilitate change and ease of use by the user?

Each of these questions is answered using a scale that ranges from 0 (not important or applicable) to 5 (absolutely essential). The constant values in Equation and the weighting factors that are applied to information domain counts are determined empirically.

Once function points have been calculated, they are used in a manner analogous to LOC as a way to normalize measures for software productivity, quality, and other attributes:

* + Errors per FP.
  + Productivity = FP/ Person-Month
  + Quality = No of faults/FP
  + Cost- $/FP
  + Documentation = Pages count / FP.

ADVANTAGES:

* + This method is independent of programming languages.
  + It is based on the data which can be obtained in early stage of project

DISADVANTAGES:

This method is more suitable for Business systems and can be developed for that domain

* + Many aspects of this method are not validated
  + The functional point has no significant ant meaning, it's just a numerical value.

##### Problem:

Given the following values, compute function point when all complexity adjustment factor (CAF) and weighting factors are average.

User Input = 45 User Output = 50 User Inquiries = 35 User Files = 8

External Interface =4

##### Solution:

As complexity adjustment factor is average (given in question), hence, scale = 3.

**Σ** Fi=14 \* 3 = 42 where i=1 to 14 CAF=0.65 + ( 0.01 \* 42) = 1.07

As weighting factors are also average (given in question) hence we will multiply each individual function point to corresponding values in TABLE.

Count\_total = (45\*4) +(50\*5) + (35\*4) + (8\*10) +(4\*7)

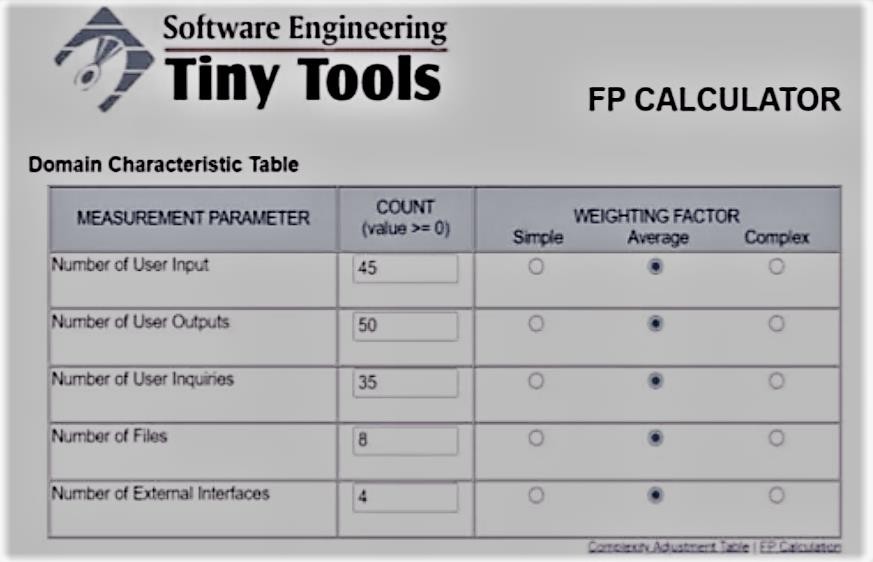
= 180 + 250+ 140 +80+ 28 = 678

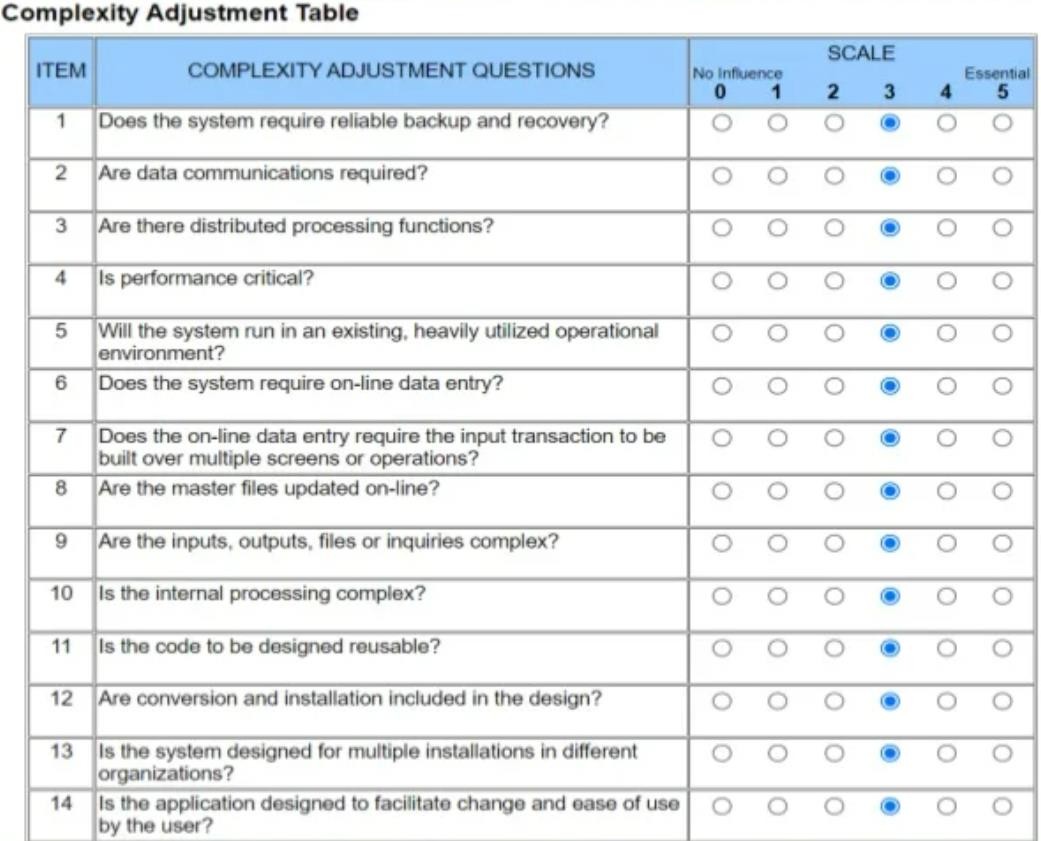
Function Point = Count\_total \*[0.65+(0.01 \* **Σ** Fi)] where i=1 to 14

= 678 \*[ 0.65 +(0.01 \* 42)] = 725.46 FP

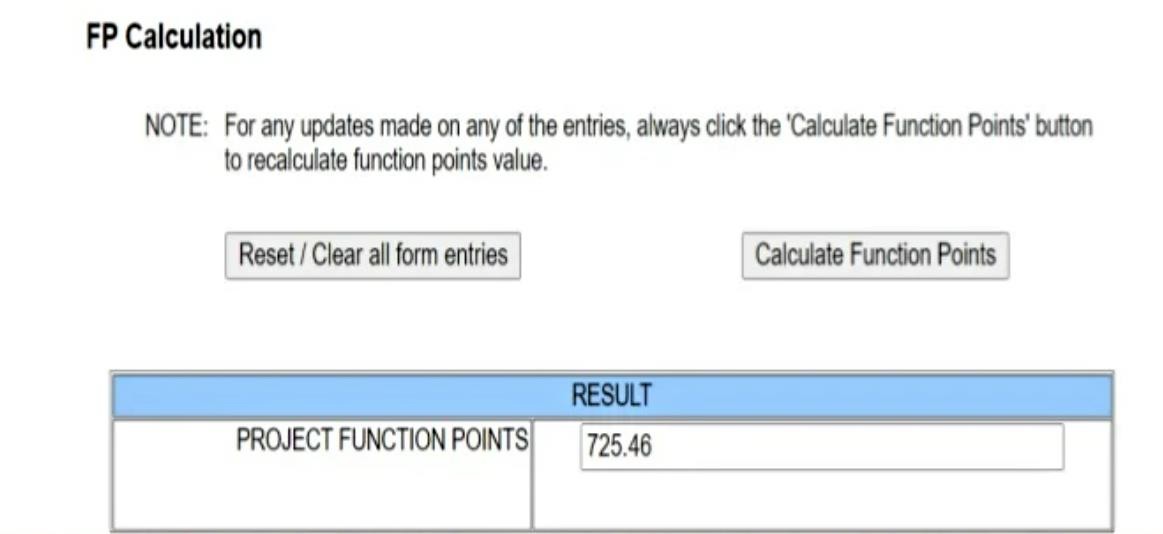
To calculate effort using FP oriented estimation model you can use the FP Calculator which is available in the following URL:

<http://groups.umd.umich.edu/cis/course.des/cis525/js/f00/harvey/>FP\_Calc.html





##### OUTPUT:



After calculating Function points of an application, we can estimate the effort easily.

For example, if application with 100 Function Points requires 25 PM Effort, we can estimate the effort for our application having 725.46 Function Points.

For 100 Function Points application Effort is 25 PM

For 725.46 Function Points application Effort = (725.46/100)\*25 =181.36 PM

**UML DIAGRAMS**

**AIM:** To Draw the UML Diagrams for the problem 1,2, 3, 4.

1. **UML DIAGRAMS FOR COURSE REGISTRATION SYSTEM:**

##### USE CASE DIAGRAM:

|  |
| --- |
|  |
| Click to view full image! |

**SEQUENCE DIAGRAM:**

|  |
| --- |
|  |
| Click to view full image! |

##### COLLABORATION DIAGRAM:

|  |
| --- |
|  |
| Click to view full image! |

**CLASS DIAGRAM:**

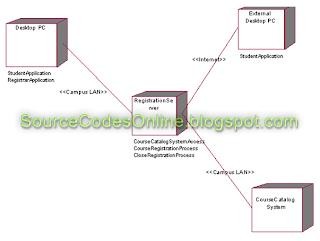
|  |
| --- |
|  |
| Click to view full image! |

##### ACTIVITY DIAGRAM:

|  |
| --- |
|  |
| Click to view full image! |

**COMPONENT DIAGRAM:**

|  |
| --- |
|  |
| Click to view full image! |

**DEPLOYMENT DIAGRAM:**

## UML DIAGRAMS FOR STUDENTS MARKS ANALYZING SYSTEM:

##### USE CASE DIAGRAM:

|  |
| --- |
|  |
| Click to view full image! |

**SEQUENCE DIAGRAM FOR STAFF:**

|  |
| --- |
|  |
| Click to view full image! |

##### COLLABORATION DIAGRAMS FOR STAFF:

Click to view full image!

##### SEQUENCE DIAGRAM FOR STUDENT:

|  |
| --- |
|  |
| Click to view full image! |

**COLLABORATION DIAGRAM FOR STUDENT:**

|  |
| --- |
|  |
| Click to view full image! |

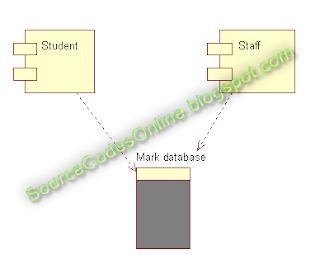
##### CLASS DIAGRAM:

|  |
| --- |
|  |
| Click to view full image! |

**ACTIVITY DIAGRAM:**

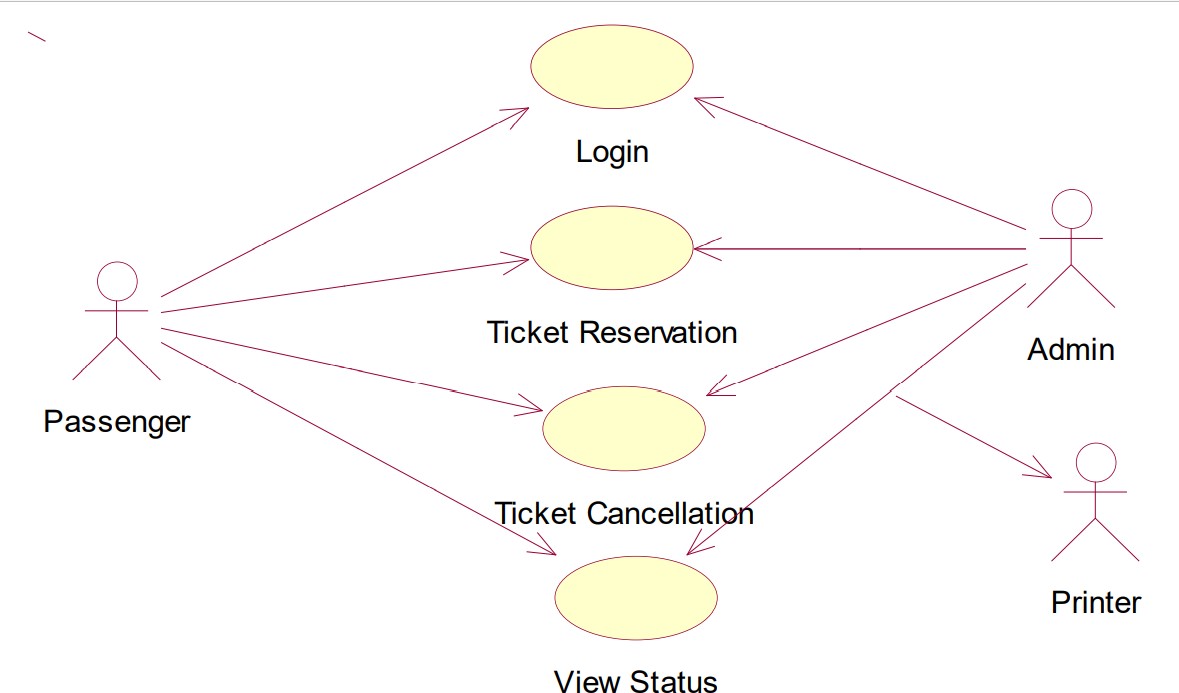
|  |
| --- |
|  |
| Click to view full image! |

**COMPONENT DIAGRAM:**

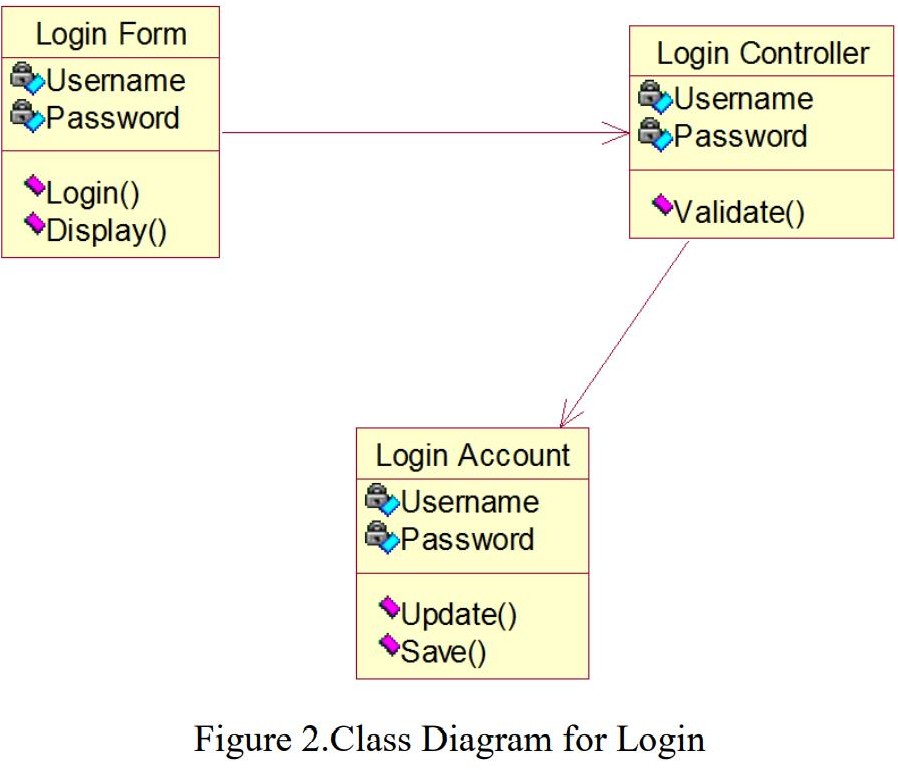


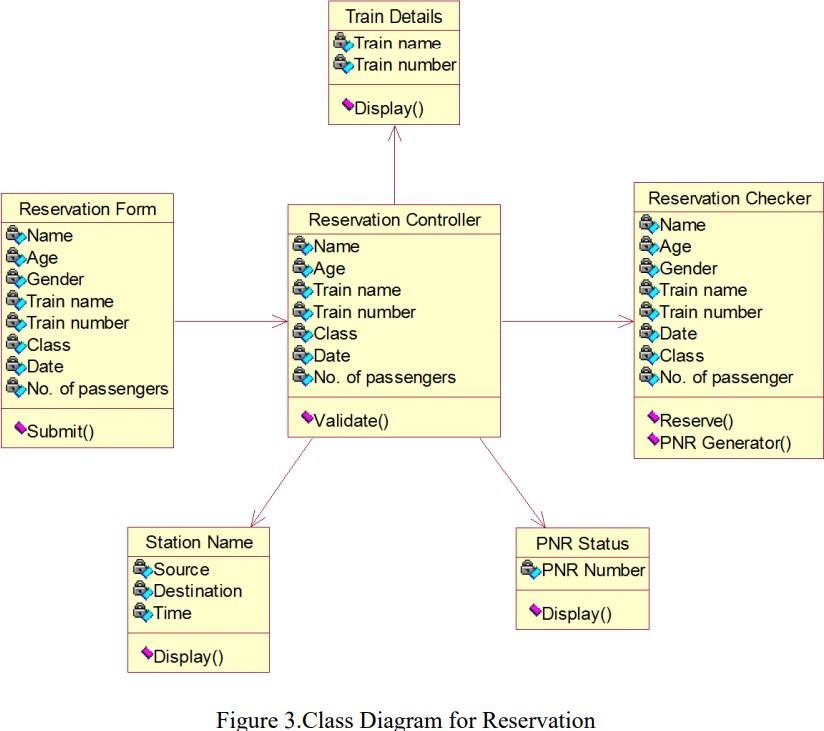
## UML DIAGRAMS FOR ONLINE TICKET RESERVATION SYSTEM:

##### USE CASE DIAGRAM:

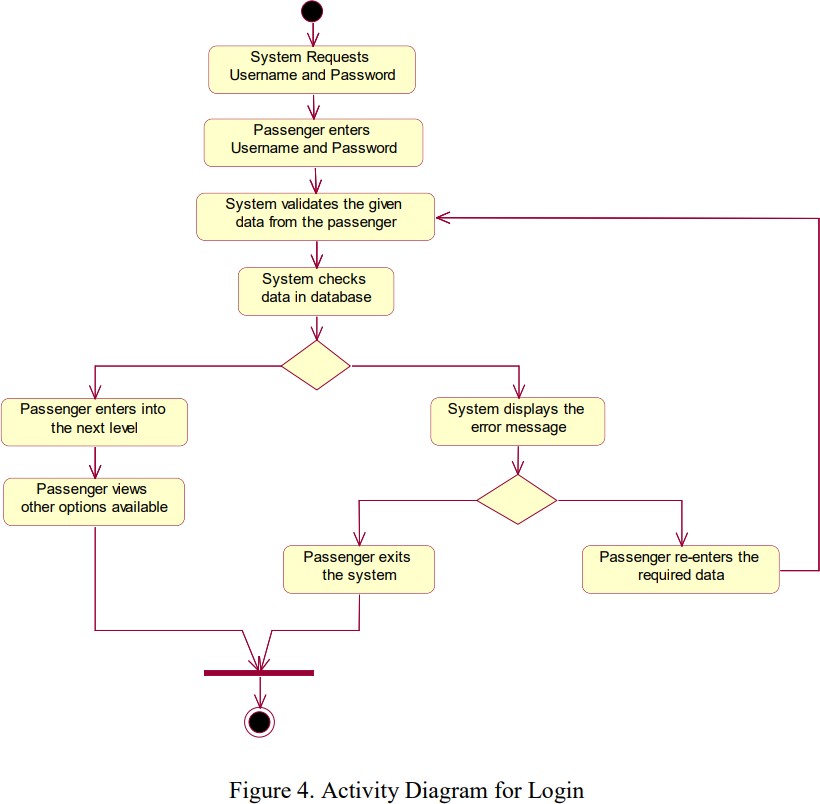


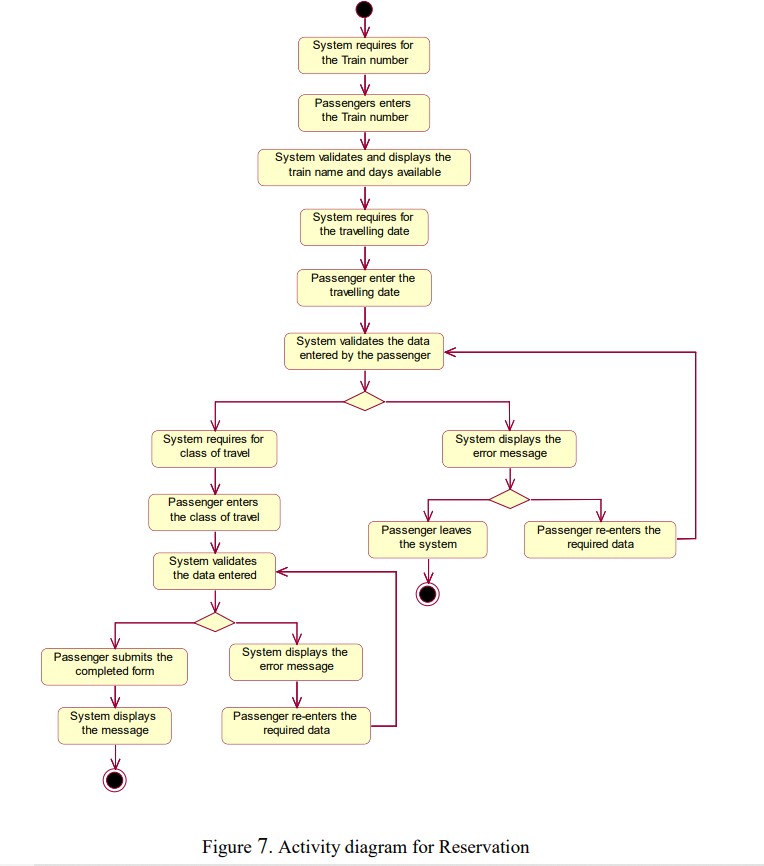
**CLASS DIAGRAM:**



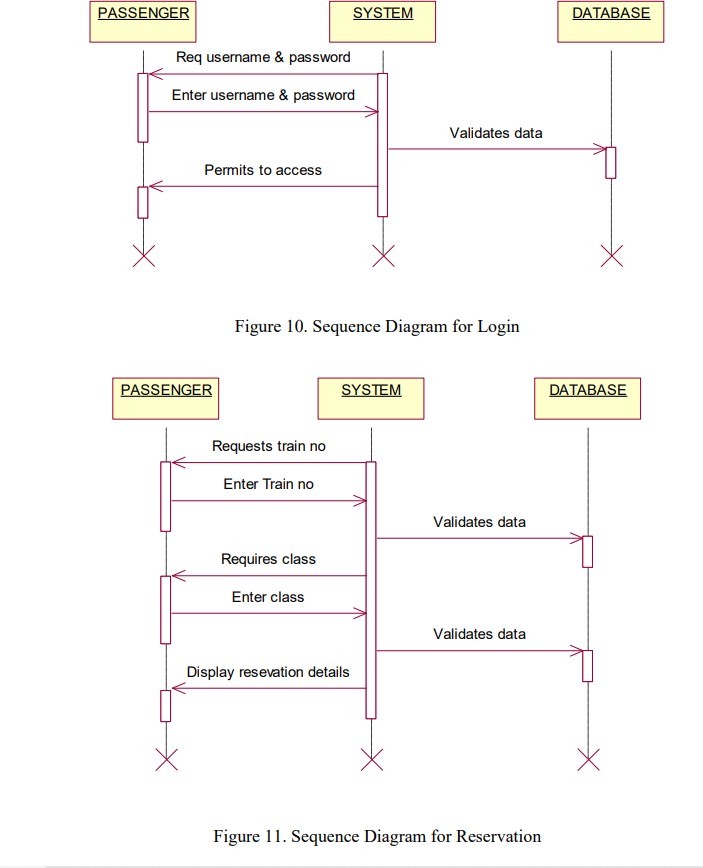


##### ACTIVITY DIAGRAM:

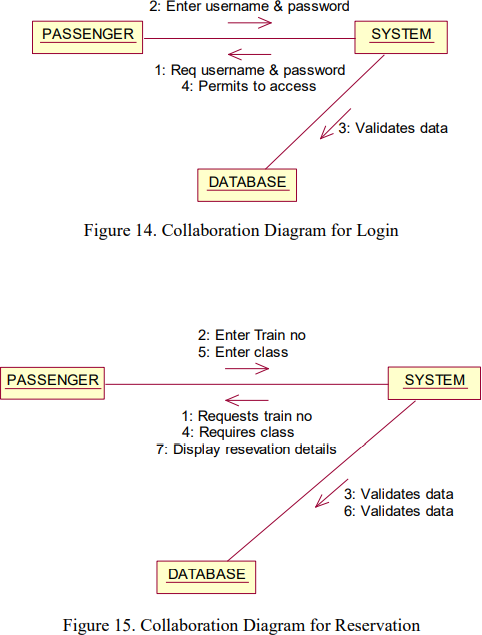




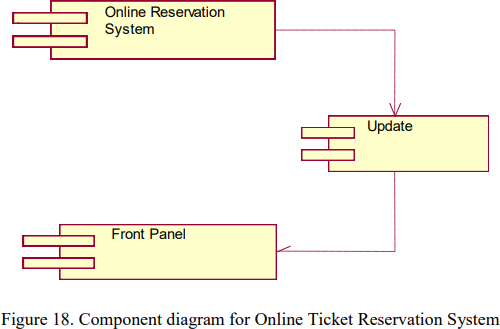
**SEQUENCE DIAGRAM:**



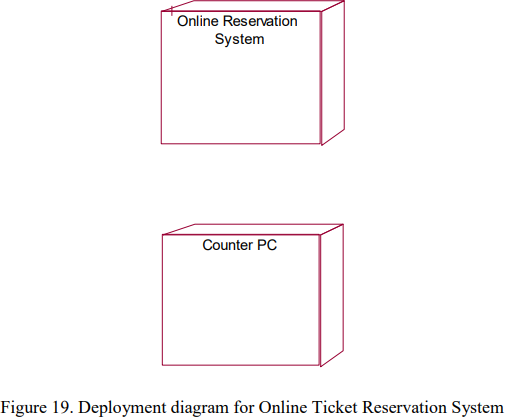
##### COLLABORATION DIAGRAM:



**COMPONENT DIAGRAM:**



**DEPLOYMENT DIAGRAM:**



## UML DIAGRAMS FOR STOCK MAINTAINANCE SYSTEM:

##### USE CASE DIAGRAM:

|  |
| --- |
|  |
| Click to view full image! |

**SEQUENCE DIAGRAM:**

|  |
| --- |
|  |
| Click to view full image! |

##### COLLABORATION DIAGRAM:

|  |
| --- |
|  |
| Click to view full image! |

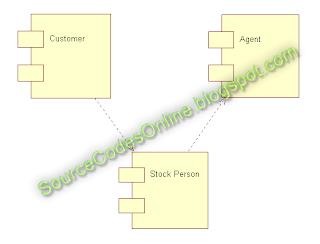
**CLASS DIAGRAM:**

|  |
| --- |
|  |
| Click to view full image! |

##### ACTIVITY DIAGRAM:

|  |
| --- |
|  |
| Click to view full image! |

**COMPONENT DIAGRAM:**



# TEST CASES FOR E-COMMERSE APPLICATION

**AIM:** To Design the test cases for e-Commerce application (Flipcart, Amazon)

###### The e-commerce business is increasing at a rapid pace especially since Covid. So it is very important to test e-commerce websites and applications such as Amazon and [Flipkart](https://www.flipkart.com/). Regressive testing of e- commerce applications is vital to make your app more robust and secure.

Effective manual and [automation test](https://www.testscenario.com/the-complete-guide-to-mobile-automation-testing/) strategies should be designed to test eCommerce applications. We need to make sure that all the pages of the application are tested regressively.

###### A test plan which covers all payment gateway test cases should also be created so that all the transactions are validated and secured. Load and Security testing should also be part of the test strategy to avoid compromises on loading time and mobile responsiveness. It would also verify the application from attacks by intruders.

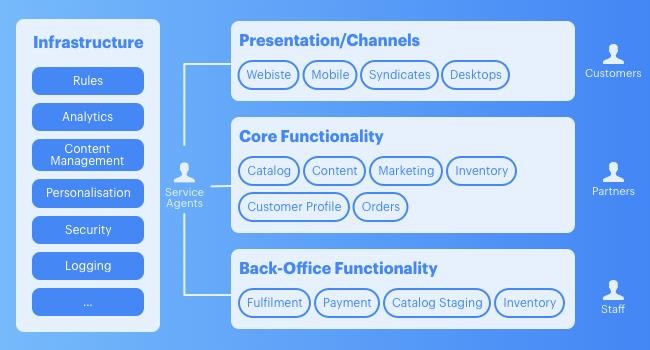
Is your eCommerce app performing as you intended? Find out by employing [**testers from testscenario**](https://www.testscenario.com/)**.** Get in touch with us!

**Ecommerce application architecture**

###### An eCommerce application has 4 important elements – Main Pages, Product Pages, Product Description Pages, and Shopping Cart. Understand these in detail to test e-commerce websites and applications efficiently.

Main Pages include homepage, privacy page, about us page, careers, etc. Product Page includes different options for the product such as size, color, and other attributes. Sorting and filtering features are considered as part of product type pages along with add-to-cart and wishlist features.

###### Product description page includes title, description, images, add to cart feature and additional info, etc. The shopping cart page should include payment options and removing a product from the cart.



Overall, an eCommerce website or an application includes different user roles such as customers, partners, staff, and service agents. Backend infrastructures include rules, analytics, security, logging, and content management.

###### The first layer which is revealed to the users is the presentation layer that includes the UI for mobile, desktops. Responsiveness is kept in mind when designing this presentation layer.

Core Functionality includes Content, Marketing, Inventory Management, Orders, and Catalog. Back-office functionality includes fulfillment, inventory, payment, and Catalog staging. The conceptional view is shown below.

## Testing Ecommerce Website Workflow

Testing of the complete workflow is important to test an eCommerce application. Let’s look at the components of the workflow such as Login and Signup, Search Functionality, Sorting Feature, Filter Feature,

Adding or removing a product to the cart, check out process, Order number generation, Invoice generation, and Payment gateway.

Below are the **test cases for the online shopping system** that will help you in testing your mobile application. These **test scenarios are for an online shopping website** that will cover all pages of your eCommerce application.

**General Test Cases for E-commerce Websites & Applications**

1. User navigation through all the pages of the application
2. None of the links in the applications should be broken.
3. Company logo, products, prices, and their description should be visible.
4. Products should be listed category-wise on the application.
5. Products should be displayed which match the search criteria.

**Product Page Test Cases**

1. Users should be able to select the desired attributes of the product-on-product page such as size, color, etc.
2. Adding a product to the cart should be possible
3. Checking whether users can add a product to the wish list.
4. Users should be able to buy the product which is added to the cart once the user is signed in.
5. Customers shouldn’t be able to add products to the cart when it

is out of inventory.

**Payment Gateway Test Cases**

1. Verify product price is correct along with shipping charges, VAT. VAT and shipping charges should be correctly applied.
2. Confirm VAT varies based on the number of products in the cart.
3. Verify all the payment methods are correctly working such as net banking, credit/debit card, and PayPal using dummy numbers for testing.
4. Ensure payment is refunded to the customer when a product is cancelled based on payment id.

**Search Functionality Test Cases**

1. Correct search results should show up for different types such as product name, brand name, or fuzzy search.
2. Search results should be relevant to the search criteria
3. Different sort options should work correctly after the search is applied.
4. Search results should be displayed as per pagination.
5. Verify search should work correctly based on different functionalities.

**Shopping Cart Test Cases**

1. User should be able to add a product to the cart.
2. Item count should be incremented when the user adds the same product again.
3. Taxes should be applied according to the delivery location.
4. User should be able to add items to the cart.
5. User should be able to update items in the cart.

**Post-Order Test Cases**

1. Email and order id should be sent after placement of order.
2. Users should be able to cancel the order.
3. There should be facility for users to track the order.
4. Users should be able to return/replace the product post-delivery.

**Test strategy for eCommerce website**

While testing an eCommerce application test strategy and test plan should be designed carefully. Let’s look at the **test plan for an online shopping cart** which includes different **types of testing needed for an eCommerce website**

1. **Browser Compatibility Testing**
2. **Load and Performance Testing**
3. **Functional Testing**
4. **Security Testing**

**How to test e-commerce websites manually**

The application should be tested manually by covering all pages and all types of testing. UI and functional testing can be done manually by testers by executing all the test cases mentioned above. Effective **test plans for online shopping websites** and creating an effective **test strategy for eCommerce websites** are the heart of testing the application manually.

**How to test e-commerce websites using** [**Selenium**](https://www.selenium.dev/)

Functional testing of an application can be done using **e-commerce testing tools** such as Appium which is based on top of selenium. Testers can automate almost all functional tests using Appium.

**Conclusion**

The above **e-commerce website testing checklist** and **test plan for an eCommerce website** would have given an idea of testing eCommerce applications such as Flipkart.

# TEST CASES FOR MOBILE APPLICATION

**Aim:** Design the test cases for a Mobile Application (Consider any example from Appstore)

Test Cases based on your mobile testing requirements. Test Cases are organized based on Mobile Testing Types. Functional Testing Test Cases

Performance Testing Security Testing Test Cases Usability Testing Test Cases

Compatibility Testing Test Cases Recoverability Testing Test Cases

**Functional Testing of Mobile Application**

The **Functional Testing of Mobile Application** is a process of testing functionalities of mobile applications like user interactions as well as testing the transactions that users might perform. The main purpose of mobile application functional testing is to ensure the quality, meeting the specified expectations, reducing the risk or errors and customer satisfaction.

The various factors which are relevant in functional testing are

* 1. Type of application based upon the business functionality usages (banking, gaming, social or business)
  2. Target audience type (consumer, enterprise, education)
  3. Distribution channel which is used to spread the application (e.g. Apple App Store, Google play, direct distribution)

The most fundamental test scenarios in the functional testing can be considered as :

1. To validate whether all the required mandatory fields are working as required.
2. To validate that the mandatory fields are displayed in the screen in a distinctive way than the non-mandatory fields.
3. To validate whether the application works as per as requirement whenever the application starts/stops.
4. To validate whether the application goes into minimized mode whenever there is an incoming phone call. In order to validate the same we need to use a second phone, to call the device.
5. To validate whether the phone is able to store, process and receive SMS whenever the app is running. In order to validate the same we need to use a second phone to send sms to the device which is being tested and where the application under test is currently running.
6. To validate that the device is able to perform required multitasking requirements whenever it is necessary to do so.
7. To validate that the application allows necessary social network options such as sharing, posting and navigation etc.
8. To validate that the application supports any payment gateway transaction such as Visa, Mastercard, Paypal etc as required by the application.
9. To validate that the page scrolling scenarios are being enabled in the application as necessary.
10. To validate that the navigation between relevant modules in the application are as per the requirement.
11. To validate that the truncation errors are absolutely to an affordable limit.
12. To validate that the user receives an appropriate error message like

“Network error. Please try after some time” whenever there is any network

error.

1. To validate that the installed application enables other applications to perform satisfactorily, and it does not eat into the memory of the other applications.
2. To validate that the application resumes at the last operation in case of a hard reboot or system crash.
3. To validate whether the installation of the application can be done smoothly provided the user has the necessary resources and it does not lead to any significant errors.
4. To validate that the application performs auto start facility according to the requirements.
5. To validate whether the application performs according to the requirement in all versions of Mobile that is 2g, 3g and 4g.
6. To perform [Regression Testing](https://www.guru99.com/regression-testing.html) to uncover new software bugs in existing areas of a system after changes have been made to them. Also rerun previously performed tests to determine that the program behavior has not changed due to the changes.
7. To validate whether the application provides an available user guide for those who are not familiar to the app

##### Performance Testing Test Cases

This type of testing’s fundamental objective is to ensure that the application performs acceptably under certain performance requirements such as access by a huge number of users or the removal of a key infrastructure part like a database server.

The general test scenarios for Performance Testing in a Mobile application are:

1. To determine whether the application performs as per the requirement under different load conditions.
2. To determine whether the current network coverage is able to support the application at peak, average and minimum user levels.
3. To determine whether the existing client-server configuration setup provides the required optimum performance level.
4. To identify the various application and infrastructure bottlenecks which prevent the application to perform at the required acceptability levels.
5. To validate whether the response time of the application is as per as the requirements.
6. To evaluate product and/or hardware to determine if it can handle projected load volumes.
7. To evaluate whether the battery life can support the application to perform under projected load volumes.
8. To validate application performance when network is changed to WIFI from 2G/3G or vice versa.
9. To validate each of the required the CPU cycle is optimization
10. To validate that the battery consumption, memory leaks, resources like GPS, Camera performance is well within required guidelines.
11. To validate the application longevity whenever the user load is rigorous.
12. To validate the network performance while moving around with the device.
13. To validate the application performance when only intermittent phases of connectivity is required.

##### Security Testing Test Cases

The fundamental objective of security testing is to ensure that the application’s

data and networking security requirements are met as per guidelines.

The following are the most crucial areas for checking the security of Mobile applications.

1. To validate that the application is able to withstand any brute force attack which is an automated process of trial and error used to guess a person’s username, password or credit-card number.
2. To validate whether an application is not permitting an attacker to access sensitive content or functionality without proper authentication.
3. To validate that the application has a strong password protection system and it does not permit an attacker to obtain, change or recover another user’s password.
4. To validate that the application does not suffer from insufficient session expiration.
5. To identify the dynamic dependencies and take measures to prevent any attacker for accessing these vulnerabilities.
6. To prevent from [SQL](https://www.guru99.com/sql.html) injection related attacks.
7. To identify and recover from any unmanaged code scenarios.
8. To ensure whether the certificates are validated, does the application implement Certificate Pinning or not.
9. To protect the application and the network from the denial of service attacks.
10. To analyze the data storage and data validation requirements.
11. To enable the session management for preventing unauthorized users to access unsolicited information.
12. To check if any cryptography code is broken and ensure that it is repaired.
13. To validate whether the business logic implementation is secured and not vulnerable to any attack from outside.
14. To analyze file system interactions, determine any vulnerability and correct these problems.
15. To validate the protocol handlers for example trying to reconfigure the default landing page for the application using a malicious iframe.
16. To protect against malicious client side injections.
17. To protect against malicious runtime injections.
18. To investigate file caching and prevent any malicious possibilities from the same.
19. To prevent from insecure data storage in the keyboard cache of the applications.
20. To investigate cookies and preventing any malicious deeds from the cookies.
21. To provide regular audits for data protection analysis.
22. Investigate custom created files and preventing any malicious deeds from the custom created files.
23. To prevent from buffer overflows and memory corruption cases.
24. To analyze different data streams and preventing any vulnerabilities from these.

##### Usability Testing Test Cases

The usability testing process of the Mobile application is performed to have a quick and easy step application with less functionality than a slow and difficult

application with many features. The main objective is to ensure that we end up having an easy-to-use, intuitive and similar to industry-accepted interfaces which are widely used.

1. To ensure that the buttons should have the required size and be suitable to big fingers.
2. To ensure that the buttons are placed in the same section of the screen to avoid confusion to the end users.
3. To ensure that the icons are natural and consistent with the application.
4. To ensure that the buttons, which have the same function should also have the same color.
5. To ensure that the validation for the tapping zoom-in and zoom-out facilities should be enabled.
6. To ensure that the keyboard input can be minimized in an appropriate manner.
7. To ensure that the application provides a method for going back or undoing an action, on touching the wrong item, within an acceptable duration.
8. To ensure that the contextual menus are not overloaded because it has to be used quickly.
9. To ensure that the text is kept simple and clear to be visible to the users.
10. To ensure that the short sentences and paragraphs are readable to the end users.
11. To ensure that the font size is big enough to be readable and not too big or too small.
12. To validate the application prompts the user whenever the user starts downloading a large amount of data which may be not conducive for the application performance.
13. To validate that the closing of the application is performed from different states and verify if it re-opens in the same state.
14. To ensure that all strings are converted into appropriate languages whenever a language translation facility is available.
15. To ensure that the application items are always synchronized according to the user actions.
16. To ensure that the end user is provided with a user manual which helps the end user to understand and operate the application who may be not familiar with the application’s proceedings

Usability testing is normally performed by manual users since only human beings can understand the sensibility and comfort ability of the other users.

##### Compatibility Testing Test Cases

Compatibility testing on mobile devices is performed to ensure that since mobile devices have different size, resolution, screen, version and hardware so the application should be tested across all the devices to ensure that the application works as desired.

The following are the most prominent areas for compatibility testing.

1. To validate that the user Interface of the application is as per the screen size of the device, no text/control is partially invisible or inaccessible.
2. To ensure that the text is readable for all users for the application.
3. To ensure that the call/alarm functionality is enabled whenever the application is running. The application is minimized or suspended on the event of a call and then whenever the call stops the application is resumed.

##### Recoverability Testing Test Cases

1. Crash recovery and transaction interruptions
2. Validation of the effective application recovery situation post unexpected interruption/crash scenarios.
3. Verification of how the application handles a transaction during a power failure (i.e. Battery dies or a sudden manual shutdown of the device)
4. The validation of the process where the connection is suspended, the system needs to reestablish for recovering the data directly affected by the suspended connection.

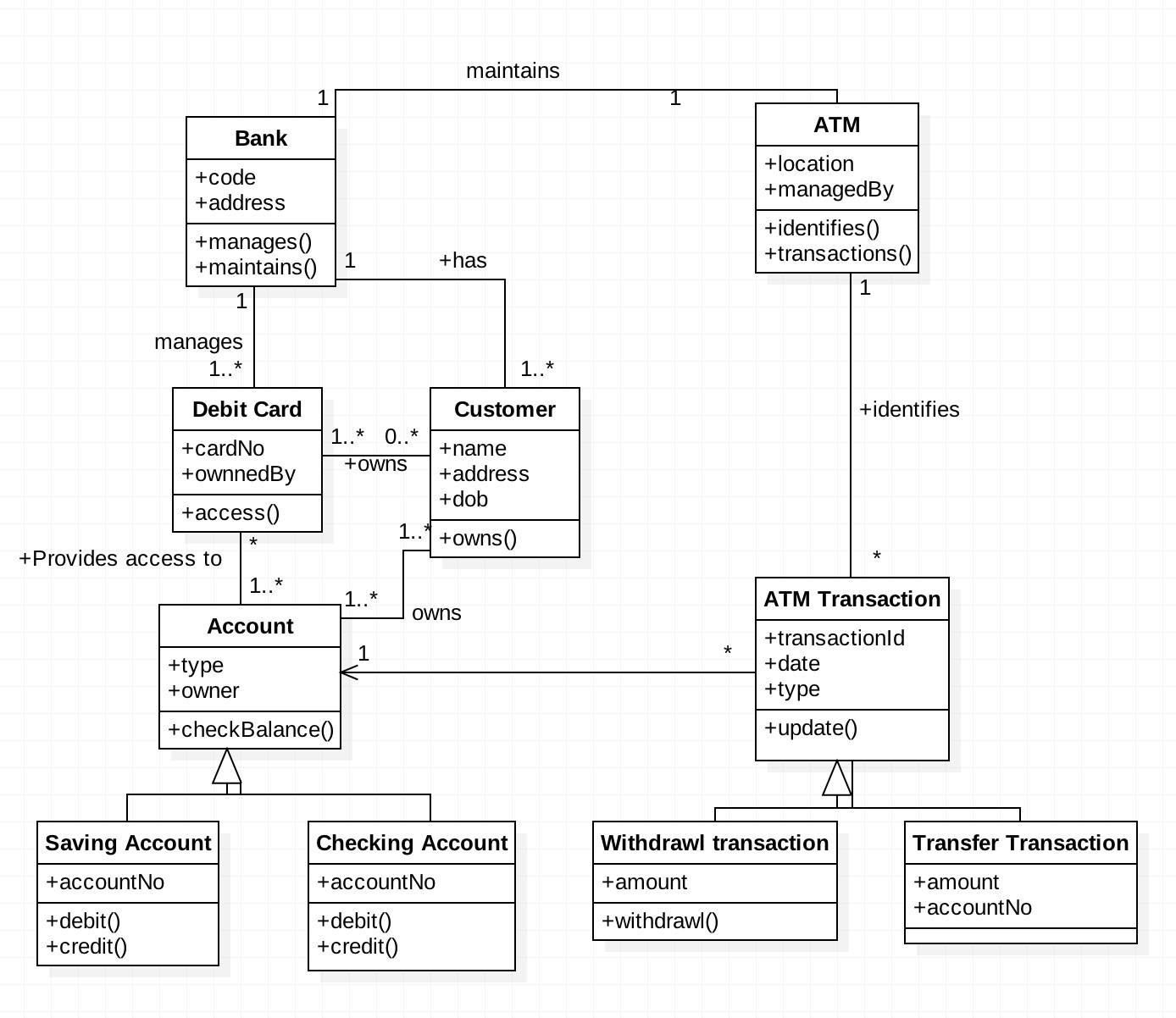
# Sample of UML Diagrams for ATM System

**AIM:** Design and Implement ATM system through UML Diagrams.

For Data: Class diagram

**Class Diagram:-** Class diagrams describe the static structure of a system, or how it is structured rather than how it behaves. These diagrams contain the following elements:

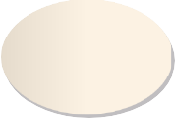
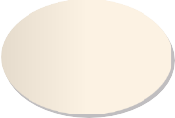
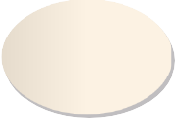
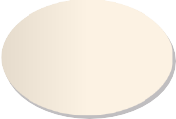
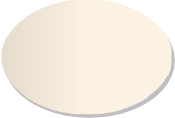
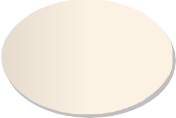
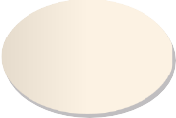
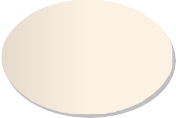
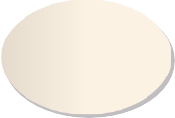
1. Classes , which represent entities with common characteristics or features. These features include attributes, operations, and associations.
2. Associations , which represent relationships that relate two or more other classes where the relationships have common characteristics or features. These features include attributes and operations.



For Function: Use case, Sequence, Collaboration/Communcation

**Use Case Diagram:** Use case diagrams describe the functionality of a system and users of the system. They contain the following elements:

1. Actors , which represent users of a system, including human users and other systems
2. Use cases , which represent functionality or services provided by a system to users Here, is a use case diagram for the ATM System.



**uc Use Case Model**

**Check Exceeding Balance**

«include»

**Withdraw Cash**

«include»

**Transfer Money**

**Check balance**

**Print Receipt**

**Log Out**

«extend»

**Bank**

**Customer**

**Deposit a Check**

**Deposit Cash**

**Log in**

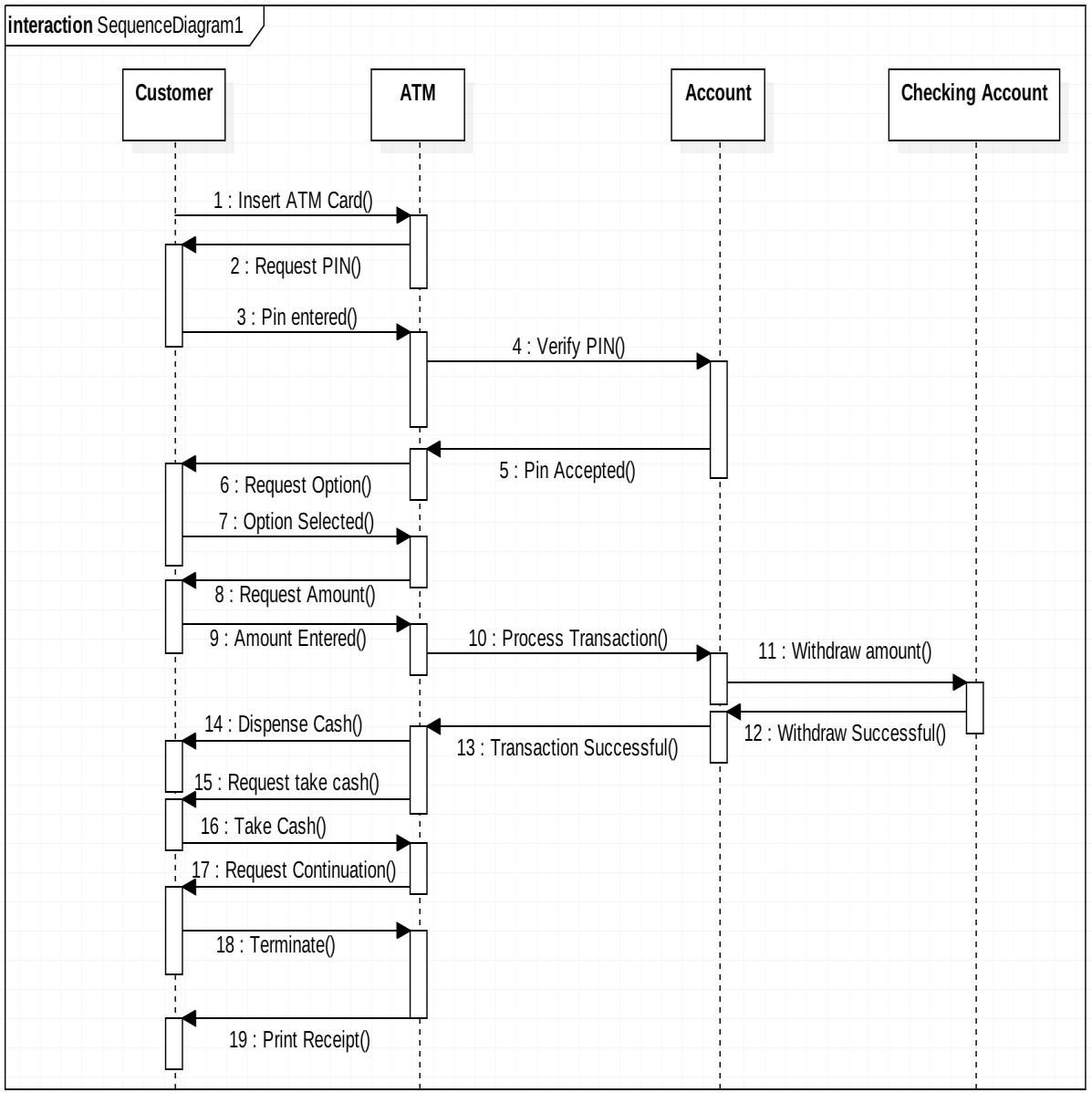
**Depit Card Login**

**Credit Card Login**

**Sequence Diagram:** Sequence diagrams typically show the flow of functionality through a use case, and consist of the following components:

1. Actors , involved in the functionality
2. Objects , that a system needs to provide the functionality
3. Messages , which represent communication between objects

Here, is an example of Sequence diagram for withdrawing amount from ATM.

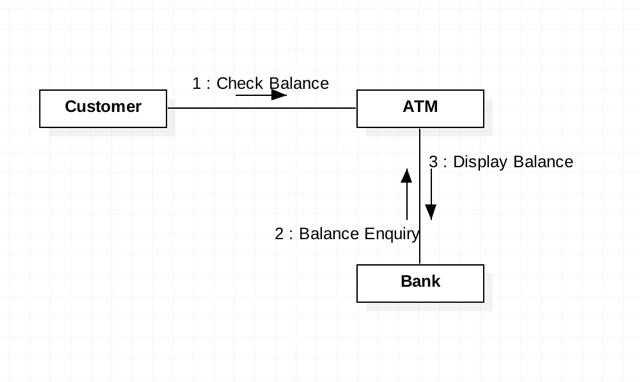


Communication/Collaboration Diagrams

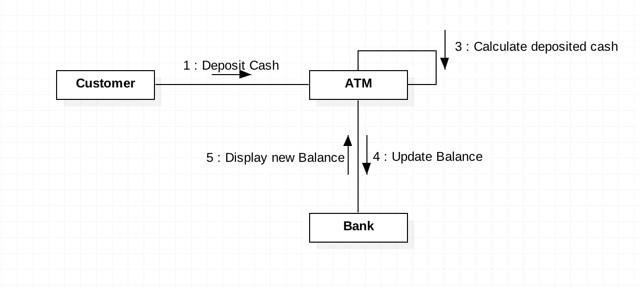
A Communication or Collaboration diagram, as shown is a directed graph that uses objects and actors as graph nodes. The focus of the collaboration diagram is on the roles of the objects as they interact to realize a system function. Directional links are used to indicate communication between

objects. These links are labeled using appropriate messages. Each message is prefixed with a sequence number indicating the time ordering needed to realize the system function.

Here is an example of the Check Balance communication diagram:



Here is an example of the Deposit Cash communication diagram:



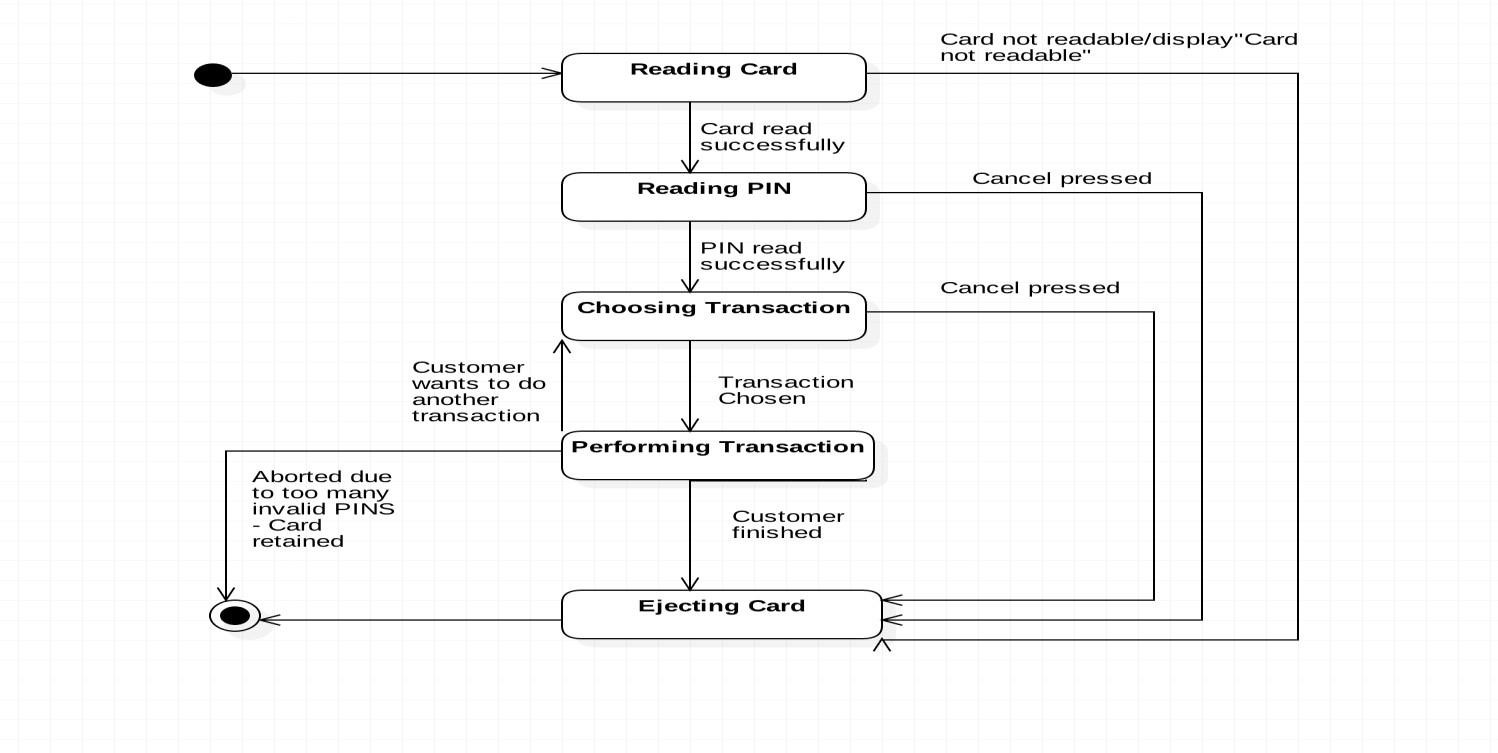
For behavior: State, Activity Diagram

State Diagram:- State transition diagrams provide a way to model the various states in which an object can exist. While the class diagram show a static

picture of the classes and their relationships, state transition diagrams model the dynamic behavior of a systen in response to extermal events (stimuli). State transition diagrams consist of the following: 1. States , which show the possible situations in which an object can find itself

2. Transitions , which show the different events which cause a change in the state of an object.

Here, is an example of the state diagram for the session of ATM.

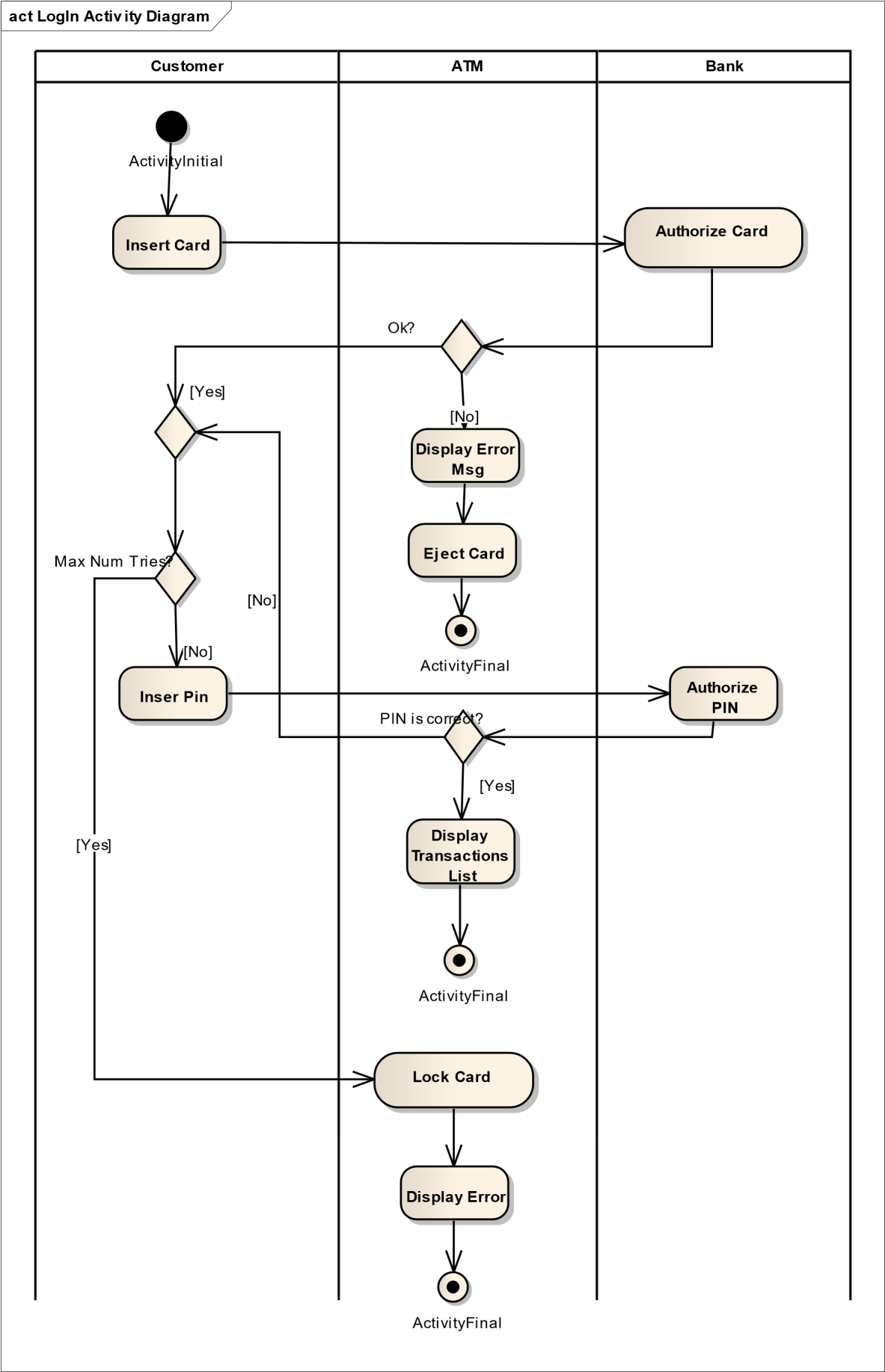


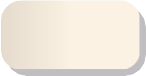
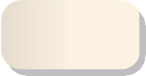
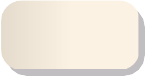
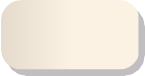
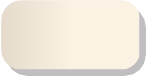
**Activity Diagram:-** Activity diagrams describe the activities of a class. They are similar to state transition diagrams and use similar conventions, but activity diagrams describe the behavior/states of a class in response to internal processing rather than external events. They contain the following elements:

1. Swimlanes , which delegate specific actions to objects within an overall activity 2. Action States , which represent uninterruptible actions of entities, or steps in the execution of an algorithm

1. Action Flows , which represent relationships between the different action states on an entity
2. Object Flows , which represent utilization of objects by action states, or influence of action states on objects.

Following are the examples of Login, Withdraw Activity Diagrams.





**act Withdraw Cash Activity Diagram**

**Customer**

**ATM**

**Bank**

ActivityInitial

**Inser Amount**

**Amount <= Balance**

**Display**

**Balance**

[No]

Ok?

[Yes]

**Eject Money**

**Update Balance**

**Display New Balance**

ActivityFinal