S NO	LIST OF EXPERIMENT
1	1) Perform the following, for the following experiments: i. Do the Requirement Analysis and Prepare SRS ii. Draw E-R diagrams, DFD, CFD and structured charts for the project.
2	Course Registration System
3	Students Marks Analyzing System
4	Online Ticket Reservation System, Stock Maintenance
5	Consider any application, using COCOMO model, estimate the effort
6	Consider any application, Calculate effort using FP oriented estimation model.
7	Draw the UML Diagrams for the problem 1,2, 3, 4.
8	Design the test cases for e-Commerce application (Flipcart, Amazon)
9	Design the test cases for a Mobile Application (Consider any example from Appstore)
10	Design and Implement ATM system through UML Diagrams.

1 A).Aim: **Requirement analysis and SRS forCourse Registration system.**

Requirements:

Hardware Requirements: PC with 300 megahertz or higher processor clock speed recommended; 233 MHz

• minimum required. 128 megabytes (MB) of RAM or higher recommended (64 MB minimum supported)

• 1.5 gigabytes (GB) of available hard disk space

• CD ROM or DVD Drive

• Keyboard and Mouse(compatible pointing device).

Software Requirements:

Rational Rose, Windows XP,

Theory:

An SRS is basically an organization's understanding (in writing) of a customer or potential client's system requirements and dependencies at a particular point in time (usually) prior to any actual design or development work. It's a two-way insurance policy that assures that both the client and the organization understand the other's requirements from that perspective at a given point in time.

The SRS document itself states in precise and explicit language those functions and capabilities a software system (i.e., a software application, an eCommerce Web site, and so on) must provide, as well as states any required constraints by which the system must abide. The SRS also functions as a blueprint for completing a project with as little cost growth as possible. The SRS is often referred to as the "parent" document because all subsequent project management documents, such as design specifications, statements of work, software architecture specifications, testing and validation plans, and documentation plans, are related to it.

It's important to note that an SRS contains functional and nonfunctional requirements only; it doesn't offer design suggestions, possible solutions to technology or business issues, or any other information other than what the development team understands the customer's system requirements to be.

 A well-designed, well-written SRS accomplishes four major goals:

It provides feedback to the customer. An SRS is the customer's assurance that the• development organization understands the issues or problems to be solved and the software behavior necessary to address those problems. Therefore, the SRS should be written in natural language (versus a formal language, explained later in this article), in an unambiguous manner that may also include charts, tables, data flow diagrams, decision tables, and so on.

It decomposes the problem into component parts. The simple act of writing down• software requirements in a well-designed format organizes information, places borders around the problem, solidifies ideas, and helps break down the problem into its component parts in an orderly fashion.

It serves as an input to the design specification. As mentioned previously, the SRS serves• as the parent document to subsequent documents, such as the software design specification and statement of work. Therefore, the SRS must contain sufficient detail in the functional system requirements so that a design solution can be devised.

It serves as a product validation check. The SRS also serves as the parent document for• testing and validation strategies that will be applied to the requirements for verification.

SRSs are typically developed during the first stages of "Requirements Development," which is the initial product development phase in which information is gathered about what requirements are needed--and not. This information-gathering stage can include onsite visits, questionnaires, surveys, interviews, and perhaps a return-on-investment (ROI) analysis or needs analysis of the customer or client's current business environment. The actual specification, then, is written after the requirements have been gathered and analyzed.

SRS should address the following The basic issues that the SRS shall address are the following:
a) Functionality. What is the software supposed to do?

b) External interfaces. How does the software interact with people, the system's hardware, other hardware, and other software?

c) Performance. What is the speed, availability, response time, recovery time of various software functions, etc.?

d) Attributes. What are the portability, correctness, maintainability, security, etc. considerations?
e) Design constraints imposed on an implementation. Are there any required standards in effect, implementation language, policies for database integrity, resource limits, operating environment(s) etc.?

Chracteristics of a good SRS An SRS should be

a) Correct

b) Unambiguous

c) Complete

d) Consistent

e) Ranked for importance and/or stability

f) Verifiable

g) Modifiable

h) Traceable

Correct - This is like motherhood and apple pie. Of course you want the specification to be correct. No one writes a specification that they know is incorrect. We like to say - "Correct and Ever Correcting." The discipline is keeping the specification up to date when you find things that are not correct.

Unambiguous - An SRS is unambiguous if, and only if, every requirement stated therein has only one interpretation. Again, easier said than done. Spending time on this area prior to releasing the SRS can be a waste of time. But as you find ambiguities - fix them.

Complete - A simple judge of this is that is should be all that is needed by the software designers to create the software.

Consistent - The SRS should be consistent within itself and consistent to its reference documents. If you call an input "Start and Stop" in one place, don't call it "Start/Stop" in another.

Ranked for Importance - Very often a new system has requirements that are really marketing wish lists. Some may not be achievable. It is useful provide this information in the SRS.

Verifiable - Don't put in requirements like - "It should provide the user a fast response." Another of my favorites is - "The system should never crash." Instead, provide a quantitative requirement like: "Every key stroke should provide a user response within 100 milliseconds."

Modifiable - Having the same requirement in more than one place may not be wrong - but tends to make the document not maintainable.

Traceable - Often, this is not important in a non-politicized environment. However, in most organizations, it is sometimes useful to connect the requirements in the SRS to a higher level document. Why do we need this requirement?

A sample of basic SRS Outline

1. Introduction

1.1 Purpose

1.2 Document conventions

1.3 Intended audience

1.4 Additional information

1.5 Contact information/SRS team members

1.6 References

2. Overall Description

2.1 Product perspective

2.2 Product functions

2.3 User classes and characteristics

2.4 Operating environment

2.5 User environment

2.6 Design/implementation constraints

2.7 Assumptions and dependencies

3. External Interface Requirements

3.1 User interfaces

3.2 Hardware interfaces

3.3 Software interfaces

3.4 Communication protocols and interfaces

4. System Features

4.1 System feature A

4.1.1 Description and priority

4.1.2 Action/result

4.1.3 Functional requirements

4.2 System feature B

5. Other Nonfunctional Requirements

5.1 Performance requirements

5.2 Safety requirements

5.3 Security requirements

5.4 Software quality attributes

5.5 Project documentation

5.6 User documentation

6. Other Requirements Appendix A: Terminology/Glossary/Definitions list Appendix B: To be determined

Conclusion: The Requirement Analysis and SRS was made successfully by following the steps described above.

1.B).Requirement analysis and SRS for Student marks Analyzing System.

1. Introduction
This document aims at defining the overall software requirements for Student marks analyzing System. The final product will be having only features or functionality mentioned in this document.

Purpose

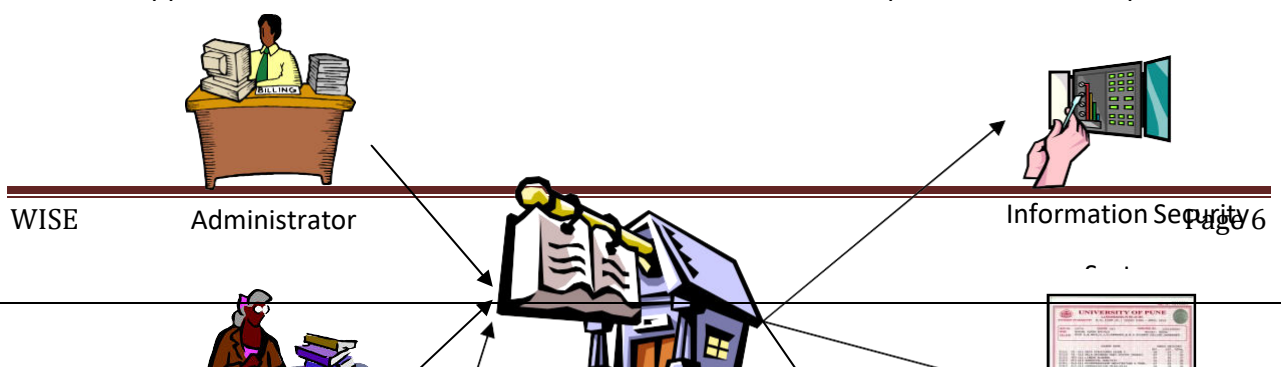**Scope**

Definitions, Acronyms and Abbreviations

References

2. General / Overall Description

Product Perspective

The application will be windows based self contained and independent software product

Mark-
sheet

Various Reports

Product Functionality

A summary of the major functions that the software will perform:

1) A login facility for enabling only authorized access to the system.
2) User will be able to add/modify/delete information about different students data enrolled for the course in different year.
3) User will be able to add/modify/delete information about different subjects that are offered in particular semester. The semester wise list of subjects along with their credit points and type will be displayed.
4) User will be able to add/modify/delete information about elective subjects opted by different students in different semester.
5) User will be able to add/modify/delete information regarding marks obtained by different students in different semester.
6) User will also be able to print mark sheets of students
7) User will be able to generate printable reports.
8) User will be able to reset the system leading to deletion of all existing information from the backend database.
9) User will be able to create/modify/delete new/existing user accounts.

User Characteristics

The different users of the system are Administrator, Marks Entry Clerk, and Coordinator.

Design and Implementation Constraints

1) Since the DBMS used in MS Access 2000, which is not a very powerful DBMS. It will not be able to store a very huge number of records.
2) Due to very limited features of DBMS being used, database auditing will not be provided

Assumption and Dependencies

1) The number of subjects to be taken by the students in each semester does not change.
2) The subject types do not change
3) The number of semester in MCA program does not change.

3. Specific Requirements

External Interface Requirements

User Interfaces

Login Screen: various fields available on this screen will be

1) User id
2) Password
3) Role: Admin/Clerk/Coordinator

Subject Information Parameter Screen: This screen will be accessible onlyto the user with admin role. The various fields available on this screen will be

1) Subject Code
2) Subject Name
3) Category / Type
4) Credits

Student Information Screen: various fields available on this screen will be

1) Student Enrollment Number
2) Student Name
3) Batch Year

Student Subject Choice Screen: It will allow user to add/modify/delete student choices for elective subjects of the semester and batch year selected in student subject choice screen. The screen will display list of available choices for elective I & elective II for the selected semester. The screen will also display the list of students enrolled during the selected batch year and currentlystudying in the selected semester.

Marks Entry Screen: various fields available on this screen will be

1) Student Enrollment Number
2) Student Name
3) Subject Name
4) Internal Marks
5) External Marks
6) Total Marks

Mark Sheet Screen: It will allow user to enter the enrollment number and the semester number of the student for whom the user want to view/print the mark sheet.

Student List Report Screen: It will allow the user to enter the batch year for which the user wants to view/print the student list report.

Rank wise List Report Screen: It will allow the user to enter the batch year and the semester number for which the user wants to view/print the Rank wise list report.

Student Subject Choices List Report Screen: It will allow the user to enter the batch year and the semester number for which the user wants to view/print the Student's choices list report.

Hardware Interfaces

1) Support for printer i.e. appropriate drivers are installed
2) Screen resolution of at least 800*600 required for proper and complete viewingof screens.

Software Interfaces

Any window based operating system
MS Access 2000 as the DBMS Crystal
Report 8
Visual Basic 6

Communication Interfaces

None

Functional Requirements

1) **Subject Information Management**
   The system will maintain information about various subjects being offered during different semesters of the course. The following information would be maintainedfor each subject. Subject code, Subject Name, Subject Type (Core / Elective / Lab1 / Lab2 / Mini Project) Semester, Credits.

2) **Student Information Management**

System will maintain information about various students enrolled in the MCA course in different years. The following information would be maintained for eachstudent:

Student Enrollment No., Student Name, Year of Enrollment. The system will allow creation/modification/deletion of new/existing students and also have theability to list all the students enrolled in a particular year.

3) Student's Subject Choice Information Management

The system will maintain information about choice of different Elective subjects opted by various students of different enrollment years in different semesters. Thefollowing information would be maintained:

Student enrollment no, Semester, Student's choices for a particular semester.

4) Marks Information Management

The system will maintain information about marks obtained by various students of different enrollment years in different semesters. The following information would be maintained:

Student enrollment no, semester, subject code, internal marks, external marks,total marks and credits.

The system will also have creation/modification/deletion of marks information.

5) Mark sheet Generation

The system will generate mark-sheet for every student in different semesters.

6) Report Generation

1) Student List Reports

For each year a report will be generated containing the list of students enrolled inthat batch year.

2) Student Subject Choice List Report

For each batch year a report will be generated containing the list of studentsand their choices for Elective subject in the selected semester.

3) Semester-wise mark lists

4) Rank-wise List Report

7) User Account Management

The system will maintain information about various users who will be able toaccess the system. The following information would be maintained.

User Name, User ID, Password and Role.

Use Cases

The various use cases will be

1) **Add/Update/Delete Student Information**
2) **Add/Update/Delete Subject Information**
3) **Add/Update/Delete Student Subject's choice Information**
4) **Generate Mark Sheet**
5) **Create/Delete User accounts**

Classes / Objects :

The various classes will be

1) Student
2) Subject
3) User

Conclusion: The Requirement Analysis and SRS was made successfully by following the steps described above

1 C). Requirement Analysis And SRS for online Railway ticket Reservation System.

OBJECTIVE

To develop software for railway reservation system with various functional and non- Functional part of design namely,

1. PROBLEM ANALYSIS AND REQUIREMENT ANALYSIS.
2. TRAIN ENQUIRY
3. TICKET GENERATION
4. TICKET CANCELLATION

The ultimate goal of this project is to develop a database that integrates the process of the Reservation of railway

INTRODUCTION

The purpose of this source is to describe the railway reservation system which provides the train timing details, reservation, billing and cancellation on various types of reservation namely,

1. Confirm Reservation for confirm Seat.
2. Reservation against Cancellation.
3. Waiting list Reservation.
4. Online Reservation.
5. PNR Generation

TECHNOLOGY USED

1. USER INTERFACE:

- Keyboard and Mouse

2. HARDWARE REQUIREMENT:

- Printer
- Normal PC
- CPU – Intel Core 2 Duo E7300
- RAM – 512MB (MIN)
- Hard Disk – 80GB

3. SOFTWARE REQUIREMENT:

- Turbo C++, C

4. OPERATING ENVIRONMENT:

The OS used are

- Windows 97
- Windows XP

INTENDED AUDIENCE:

The different types of readers are

1. Developers
2. Customers
3. Management people specifically,
4. Passengers
5. Clerk

DEFINITIONS, ACRONYMS AND ABBREVIATIONS

1. NTES – National Train Enquiry System
2. IVRS – Interactive Voice Response system
3. PRS – passenger reservation system

   It consists of

   - o Train details
   - o Reservation form
   - o Billing
   - o Cancellation.

GENERAL DESCRIPTION It enables us to maintain the railway train details like their timings, number of seat available and reservation billing and cancelling the tickets.

COMMUNICATION INTERFACES

- Indian Railway's web-site, www.indianrail.gov.in offers PRS enquiries on the internet Berth/Seat availability, Passenger Status, Fare, Train Schedule etc,.
- National Train Enquiry System (NTES) website, www.trainenquiry.comgives dynamic information about the running status of any train and its expected arrival/departure at any given station.
- Mobile telephone based SMS enquiry service. A new mobile phone based facility for rail users' which is. Country wide extension of Universal Rail Enquiry number "139"through setting up of Interactive Voice Response System (IVRS).

OPERATIONS

1. Any Reservation counter from 8 am to 8 pm.
2. Prior to 90 days of Journey.
3. One form for 6 persons only.
4. To save time & queues Agent is others guides.

PRODUCT FUNCTION

- It tells the short note about the product.

TRAIN DETAILS

- Customers may view the train timing at a date their name and number of tickets.
- Passengers operated Enquiry Terminals.

PERFORMANCE REQUIREMENTS

- o It is available during all 24 hours.
- o Offered through Mail express, super-fast, Rajdhani & Shatabdi Trains.
- o About 1520 Trains runs daily.

Variety of compartments based on comfort:

4. AC first class.
5. AC sleeper.
6. First class.
7. AC three tier.
8. AC chair car.
9. Sleeper class
10. Ordinary chair car.

Types of concerns & complexities:

11. 44 types of quotas.
12. 8 types of trains.
13. 9 types of classes.
14. 162 types of concessions.
15. 127 types of bogies

SOFTWARE SYSTEM ATTRIBUTES:

16. Reliable
17. Available
18. Secure

DOCUMENT APPROVAL

The bill passed on any proposals related to railway management needs approval of Ministry of railway department.

ER DIAGRAM:
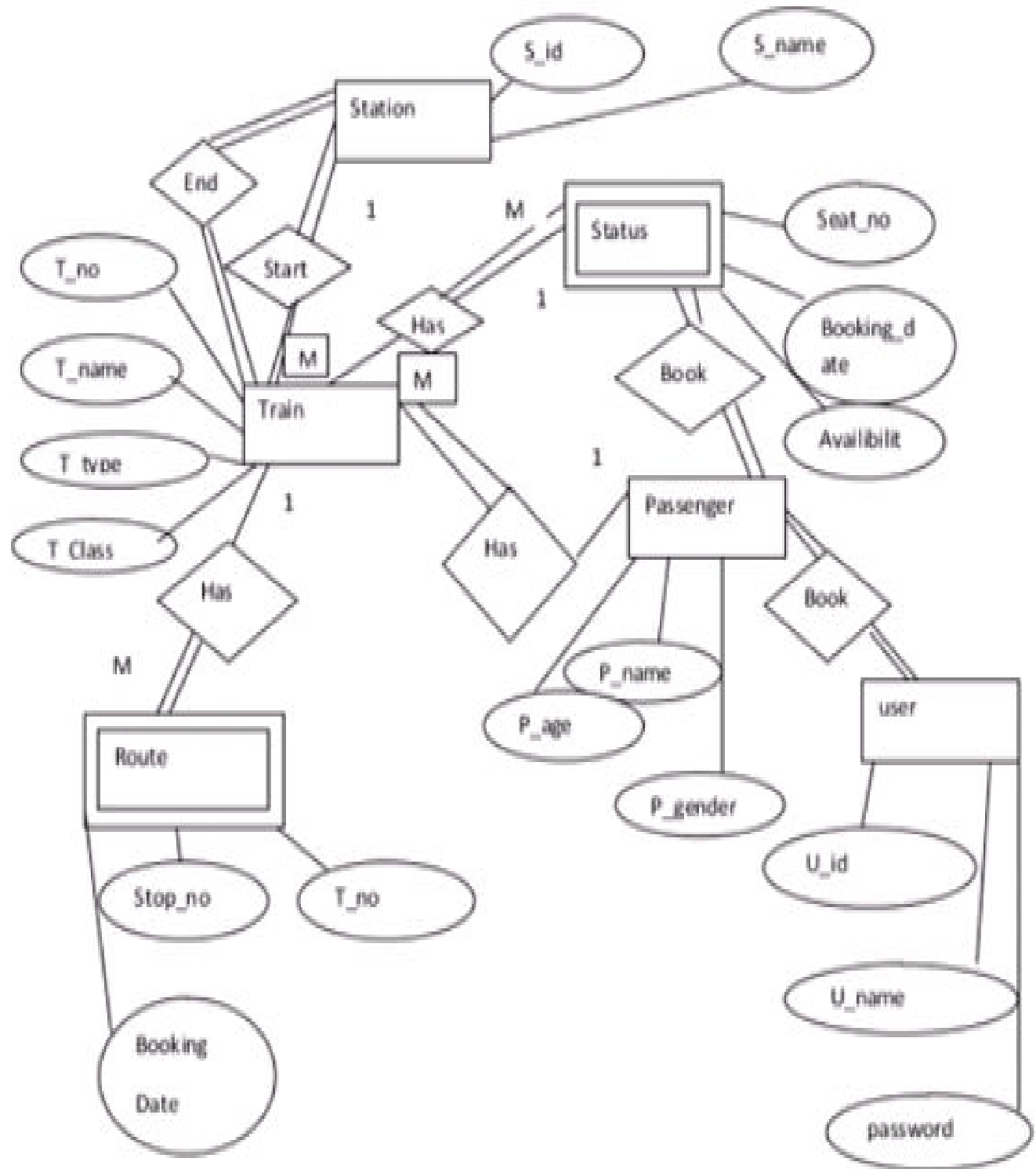
Conclusion: The Requirement Analysis and SRS was made successfully by following the steps described above.

1 D). Requirement Analysis and SRS for Stock Maintenance System.

INTRODUCTION

The Aim of the project is to implement the STOCK MAINTENANCE SYSTEM to manage the entiredetails of the stock system in an automated way.

PROBLEM STATEMENT:

A problem statement is a concise description of the issues that need to be addressed by a problem solving team and should be presented to them (or created by them) before they try to solve the problem. When bringing together a team to achieve a particular purpose provide them with a problem statement.

The main purpose of the stock maintenance system is to manage the entire details of the stock system in an automated way. The stocks which are purchased from the various dealers and suppliers are stored in the store keeper and their entries are recorded into the database. The software system provides facilities for adding new item, removing an item, updating the stock, calculating the turn over, sales amount, total number of stocks. It also involves purchasing of stocks by the customers.

DOCUMENT CONVERSION:

The document follows the IEEE format standard (IEEE Std. 830 – 1998). The format of this SRS is simple. Bold face and Indentations are used for general topics and for specific point of interest and font used is Times New Roman. The remainder of this document will be written using standard Arial font.

INTENDED AUDIENCE AND READING SUGGESSIONS:

The intended audience of this document are shop keepers, innovative team members, our department members, and all business entrepreneurs. This document will be reviewed frequently by the aboveaudience, to check if different phases of the project are completed by meeting given

requirements. If any changes made, it should be accounted.

PROJECT SCOPE:

- Effective storage of stocks
- Advanced and customized search options
- Disciplined customer service
- Secured storage area
- Improved and optimized service

REFERENCES:

Pressman, Roger S. Software Engineering: A Practitioner's Approach. New York, NY: McGraw-Hill,2005.

Lecture slides

www.scribd.com/26141396/srs-Stock_Master ,
www.scribd.com/26623608/StockMaintenanceSystem

www.a1vbcode.com/app-4464.asp

The user manual can be read in order to understand the specification. In case of classification anddetails of access the manual can be referred.

OVERALL DESCRIPTION:

Product Perspective:

The proposed stock maintenance system is an on-line system. This system will provide availablestock products for customer (consumer) needs. This system also provides customer feedback service.

PRODUCT FUNCTION:

The main functions of the specified product are:

- Automatic customer registrations

- Obtain the current stock rates and log the required information

- Provides friendly relationship with shopkeeper and customers

- Gives complete statistical data on particular stock

- Updates the stock values periodically and automatically

USER CHARACTERISTICS:

There are various kinds of users for this product. These products are purchased via online by many customers (E-Shopping). User classes may be differentiated based on frequency of use, subset of product functions used, technical enterprise, security or privilege levels.

** Naive Users:

Customers who require product for daily use, Such as common people, office goers etc...

** Warehouse Manager:

Manages items in the warehouse does packaging & delivery.

** Inventory Management:

Re-ordering and ordering based on arrival of stocks.

** Shipping Vendor:

Picks up the packages from the warehouses and delivers to the users, gives facility for order tracking.

** Managing Administrator:

Manages the whole process in a shop involving bill calculation, resolves financial problems, approverefunding and order acceptance and cancellation.

OPERATING ENVIRONMENT:

REQUIREMENTS	PARTICULARS
Operating System	Windows Environment
Processor	Pentium 4 and above
Hard Disk	100-150 GB
RAM	Minimum 1 GB
Web Browser	Internet Explorer 5.0 or Mozilla Firefoxor Google Chrome
Software Required	Stock Master 1.0, My SQL, Oracle 9i

DESIGN AND IMPLEMENTATION CONSTRAINTS:

Each and every user and manager have separate id and password. Detail regarding every user isvisible to administrator but not vice versa. Credit card authentication will be approved by the administrator. Inventory and Warehouse management are not automated. Feedbacks from users areaccounted.

- **User Documentation**

This product will include user manual. This will include product overview, complete configuration of the software's to be used and hence the description of the overall product<List the user documentation components (such as user manuals, on-line help, and tutorials) that will be delivered along the software. Identify any known user documentation delivery formats or standards.>

- **Assumptions and Dependencies**

Each user must be provided with id and password on his/her first appearance. Administrator is responsible for all activities happening. He should hold the authority of permitting any activities regarding stocks. Database system must be connected across the network for any time and any where access. Customers alone are responsible for the safety of their products.

FUNCTIONAL REQUIREMENTS:

External Interface Requirements:

The system uses the GUI – Graphical User Interface for easy interaction with the customer. The system maintains a relationship with the Rational Rose Tool. According to the code generated by the Rose tool, the system is developed. This gives more sequential access for the functions and the functions can be coded easily.

User Interfaces:

The User Interface should be very attractive featuring importance of our system. The introductory screen consists of a Welcome note and product advertisement. The next screen is the user login screen. The next interface displays the product details under different categories and the subsequent screens contain the details of each and every product and finally provided with provision for getting card number for settling the cash by the customers. The final screen displays salutations to the customer by displaying Thank You. At the same time the stock details are updated.

Hardware Interfaces:

- Needed: Computers

- Hard Disk: 100-150 GB

- RAM: 512-1 GB required

- Internet Connection required.

- Cables, wires, Network adapters are required

Software Interfaces:

SYSTEM SOFTWARE REQUIRED: Windows XP or Windows 7 with 32 bit (recommended).

APPLICATION SOFTWARE REQUIRED: Stock Master v 1.0, Oracle 9i, MY SQL, Tally 5.0.

Source of input:

The input is given by the user who wishes to use the Stock Maintenance system. The user feelsit easy to give the inputs, as the system is more user-interactive. They find the option to perform their work more easily rather than waiting for a long time to get the transactions to be completed manually.

Destination of output:

The input given by the user is updated into the database where the account details corresponding to the user are stored. With the help of the database the account details of the customer can be administered and monthly statements can be generated.

Accuracy:

The Stock Maintenance system that is developed is more accurate and is efficient. The details are maintained accurately and updated properly.

Information flows:

The data given by the user flows over stage by stage and reaches the database finally for making insertionor updating for storing the details. This can be represented by the following Data Flow Diagrams.

Communication Interfaces:

The local system must be connected to the server via Internet Connection. Email and file transfer servicesare provided. E-Shopping is the key concept.

SYSTEM FEATURES

Automated Functioning:

This system provides automated functionalities like stock updating, product listing, calculating the total stocks available etc.. This feature is of high priority only based on this feature other aspects are designed. No risk, high expenses are involved in implementation of this system. Risk rate is 2 (very low).

OTHER NON-FUNCTIONAL REQUIREMENTS:

Performance Requirements:

The number of users is not confined to any specific value. Any number of users can use the system. But the only constraint is that only one user can use this system at a time. The response time is greater for the system. It gives the output quickly so that the user will feel easy to proceed with the next transaction. The amount of information to be handled is enormous. The database stores a large amount of data and it also helps in quick insertion and retrieval of data from it. So, the system furnishes all the required information at the request of the user to view the details.

Safety Requirements:

Avoid frequent usages of flash devices such as pen drive to prevent database collapse. System must be installed with original version of Windows OS with perfect license to avoid duplication problems. Take regular backup of data. Store the stocks in a closed place to avoid loss of stocks by theft, misplacement etc...

Security Requirements:

There will be proper security mainly regarding data accessibility. Security to user can be provided by login authentication. Data stored in database should be private that is it must be known only to administrator who is authorized using a secured id. The whole system is prevented from unauthorized access.

Software Quality Attributes:

The additional quality characteristics are important both for the customers and the shop administrators and also for the developers. Some of the attributes of our system are as follows:

Adaptability: The software designed must be suitable for managing any kind of products. It should be well received both in a provision store and in a medical dispensary.

Availability:

The system must be available at an affordable rate. Also must be provided with proper license only for a period of days.

Correctness:

The system must be accurate and less error prone. During design phase itself, the system must ensure that each and every module is accurate.

Flexibility:

The system must be flexible in the sense that it must able to handle different types of users and different types of administrators. Also it should run in different environments.

Maintainability:

The system should have the capability of self-maintenance. For a good performance, everything must be optimized time to time. This feature tests the maintenance and ability to maintain the system by administrator.

Portability:

The system developed as a whole should be of very small size. So that it is easily portable with the help ofhand-held device.

Reliability:

The system should be reliable. That it should be consistent and should provide good results over a period of time.

Reusability:

The system should have the provision of using it more number of times. If any problem occurs to the whole system, it must be reinstalled and again it should be installed and put to use.

Robustness:

The system should be incorruptible. It should be able to produce results for important details though someof the features have failed.

Testability:

The system should be demonstrated, predictable, confirmable, factual and absolute for all kinds of users.

Usability:

The effectiveness, efficiency and satisfaction with which user can achieve tasks in a particular environment of a product. High usability means a system is easy to learn and remember, efficient, visually pleasing and fun to use and quick to recover from errors.

Cost-Effective:

The system should be cost effective in the sense it should be bringing the best possible profits or advantages for the lowest possible costs.

Business Rules:

The simple rules to be followed are:

- If customer wants an enquiry of products he should definitely have id and password.

- Cell phones, cameras are not permitted inside the stores.

- No bribing must be involved.

- No unnecessary conflicts and nuisances are entertained

OTHER REQUIREMENTS:

Foreign exchanges and foreign credit cards are accepted through special card readers. The system requirescollaboration with any banks for financial handlings.

Appendix A: Glossary

   <Define all the terms necessary to properly interpret the SRS including acronyms and abbreviations. You may wish to build a separate glossary that spans multiple projects or the entire organization, and just include terms specific to a single project in each SRS.>

Appendix B: Analysis Models

   <Optionally, include any pertinent analysis models, such as data flow diagrams, class diagrams, statetransition diagrams or entity-relationship diagrams.>

Appendix c: To Be Determined List

   <Collect a numbered list of the TBD (to be determined) references that remain in the SRS so theycan be tracked to closure.>

CONCLUSION:

Thus, the computerization of STOCK MAINTENANCE SYSTEM does effectively reduce the manual work involved in managing the entire details of the stock system. It saves time and gives easy access for already stored information. It enables the system in providing faster services to the applicants. The system has effective management of records which holds all the information of a particular user. This system also provides additional facilities like calculating turnover, adding new item, removing an item, updating the stock.

1.E) Draw E-R Diagram for Course Registration system.

This ER (Entity Relationship) Diagram represents the model of Course Registration System Entity. The entity-re lationship diagram of Course Registration System shows all the visual instrument of database tables and the rela tions between Fees, Students, Course, Trainers etc. It used structure data and to define the relationships between structured data groups of Course Registration System functionalities. The main entities of the Course Registra tion System are Course, Fees, Syllabus, Students, Regis trations and Trainers.

Course Registration System entities and their attributes:

• Course Entity: Attributes of Course are course_id, course_student_id, course_registration, course_name, course_type, course_year, course_description • Fees Entity: Attributes of Fees are fees_id,

fees_amount, fees_type, fees_description

• Syllabus Entity: Attributes of Syllabus are syllabus_id, syllabus_course_id, syllabus_name, syllabus_type, syllabus_description

• Students Entity: Attributes of Students are student_id, student college_id, student_name, student_mobile, student email, student_username, student_password, student address

• Registrations Entity: Attributes of Registrations are registration_id, registration_student_id, registration_name, registration_type, registration number, registration_date, registration_description

• Trainers Entity: Attributes of Trainers are trainer_id, trainer_course_id, trainer_name, trainer_mobile, trainer_email, trainer_ username trainer_password, trainer_address

Description of Course Registration System Database:

• The details of Course is store into the Course tables respective with all tables

• Each entity (Trainers, Syllabus, Registrations, Fees, Course) contains primary key and unique keys.

• The entity Syllabus, Registrations has binded with

Course, Fees entities with foreign key

• There is one-to-one and one-to-many relationships available between Registrations, Students, Trainers, Course

• All the entities Course, Registrations, Syllabus, Trainers

are normalized and reduce duplicacy of records

• We have implemented indexing on each tables of Course Registration System tables for fast query

execution.

ER Diagram For Course Registration System
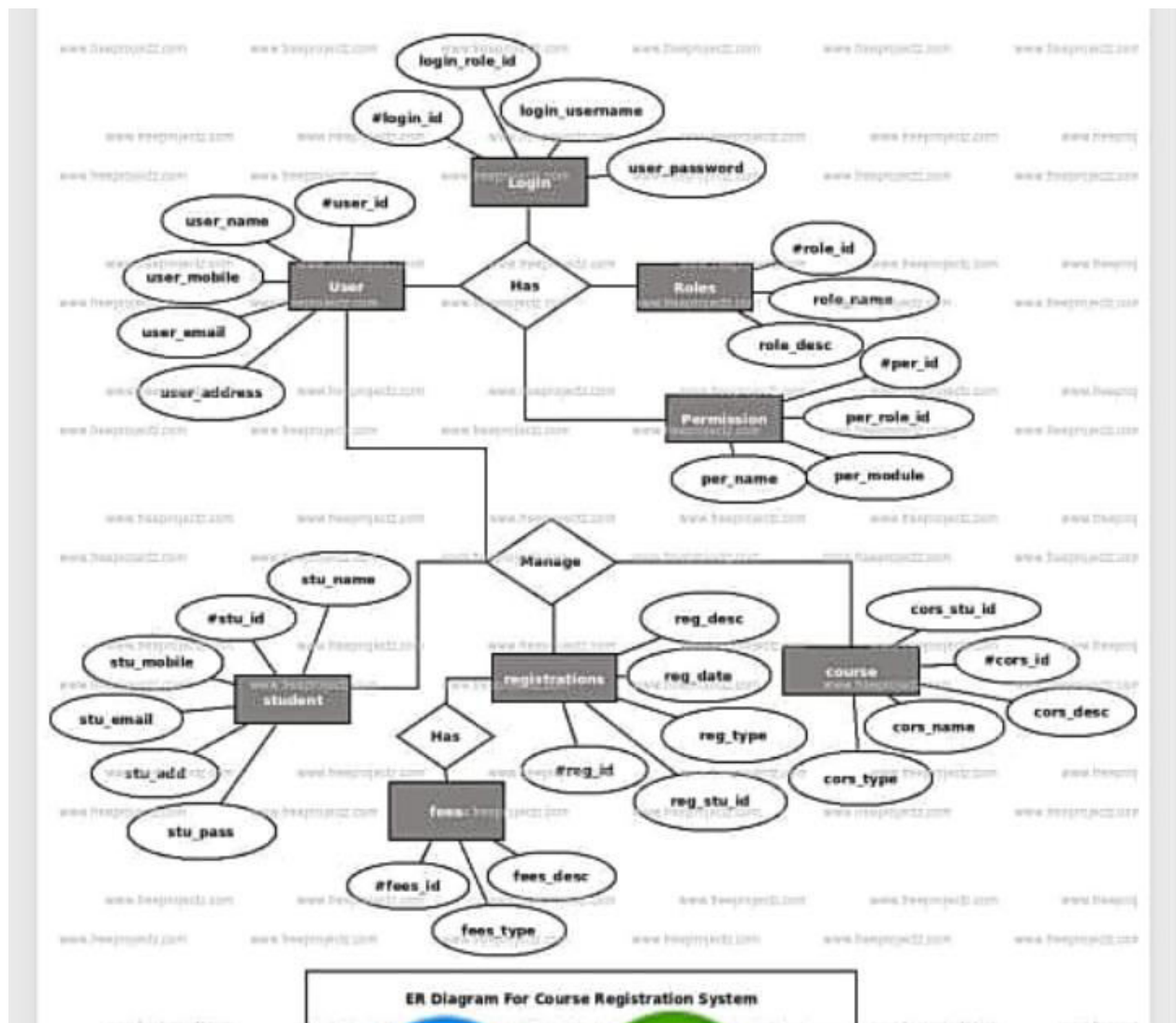
1.F) Draw DFD for Course Registration system.

Course Registration System Data flow diagram is often used as a preliminary step to create an overview of the Course without going into great detail, which can later be elaborated.it normally consists of overall application data flow and processes of the Course process. It contains all of the userflow and their entities such all the flow of Course, Fees, Syllabus, Students, Trainers, Registration, Login. All of the below diagrams has been used for the visualization of data processing and structured design of the Course process and working flow.

Zero Level Data Flow Diagram(0 Level DFD) Of Course Registration System :

This is the Zero Level DFD of Course Registration System, where we have eloborated the high level process of Course. It's a basic overview of the whole Course Registration System or process being analyzed or modeled. It's designed to be an at-a-glance view of Trainers,Registration and Login showing the system as a single high-level process, with its relationship to external entities of Course,Fees and Syllabus. It should be easily understood by a wide audience, including Course,Syllabus and Trainers In zero leve DFD of Course Registration System, we have described the high level flow of the Course system.

High Level Entities and proccess flow of Course Registration System:

• Managing all the Course

• Managing all the Fees

• Managing all the Syllabus

• Managing all the Students • Managing all the Trainers

• Managing all the Registration

• Managing all the Login

Zero Level DFD - Course Registration System
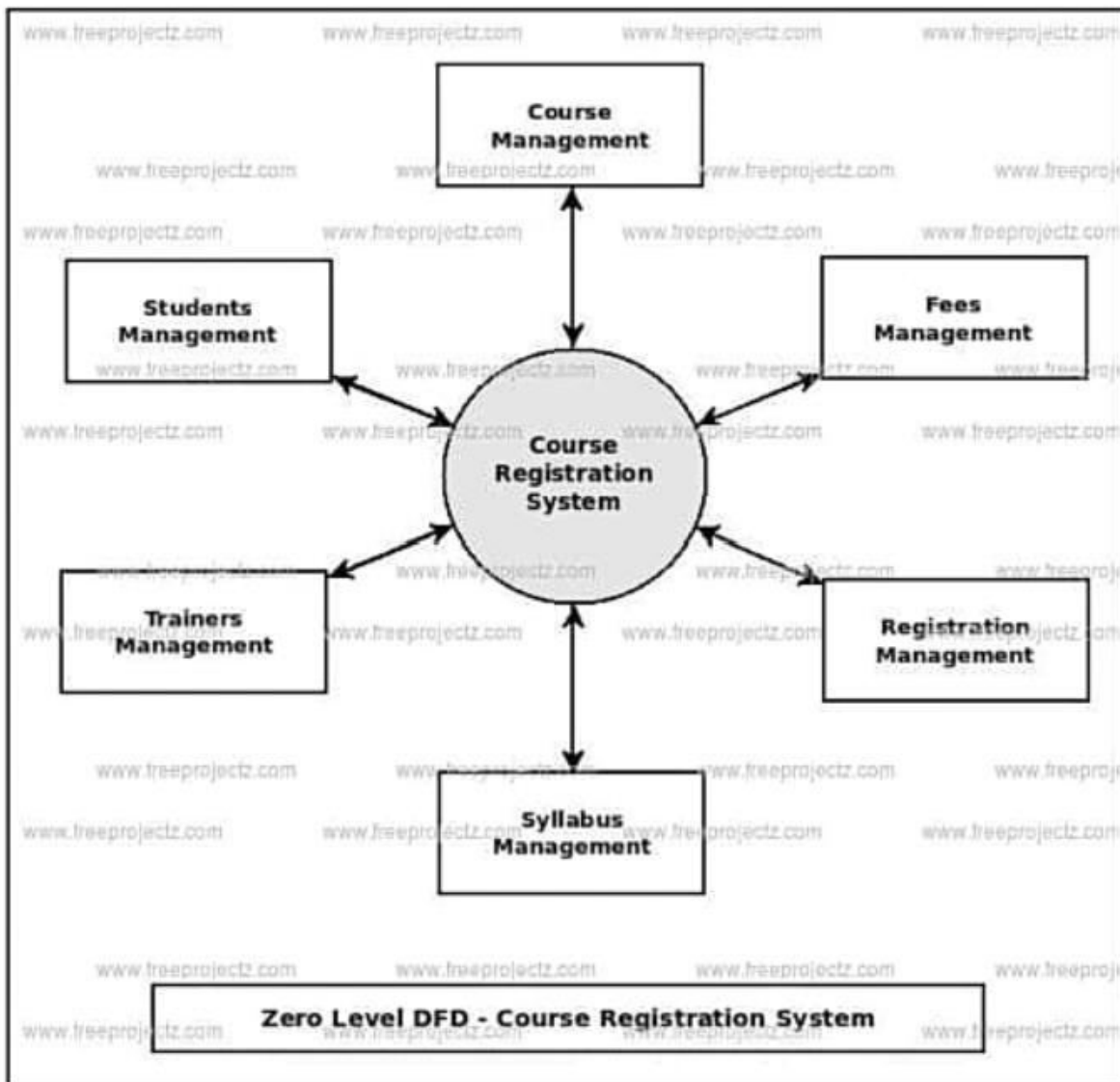
First Level Data Flow Diagram(1st Level DFD) Of Course Registration System :

First Level DFD (1st Level) of Course Registration System shows how the system is divided into sub-systems (processes), each of which deals with one or more of the data flows to or from an external agent, and which together provide all of the functionality of the Course Registration System system as a whole. It also identifies internal data stores of Login, Registration, Trainers, Students, Syllabus that must be present in order for the Course system to do its job, and shows the flow of data between the various parts of Course, Syllabus, Registration, Login, Trainers of the system. DFD Level 1 provides a more detailed breakout of pieces of the 1st level DFD. You will highlight the main functionalities of Course.

 Main entities and output of First Level DFD (1st Level DFD):

• Processing Course records and generate report of all

Course • Processing Fees records and generate report of all

Fees • Processing Syllabus records and generate report of all

Syllabus

Students • Processing Trainers records and generate report of all

• Processing Students records and generate report of all Trainers • Processing Login records and generate report of all Login

• Processing Registration records and generate report of all Registration
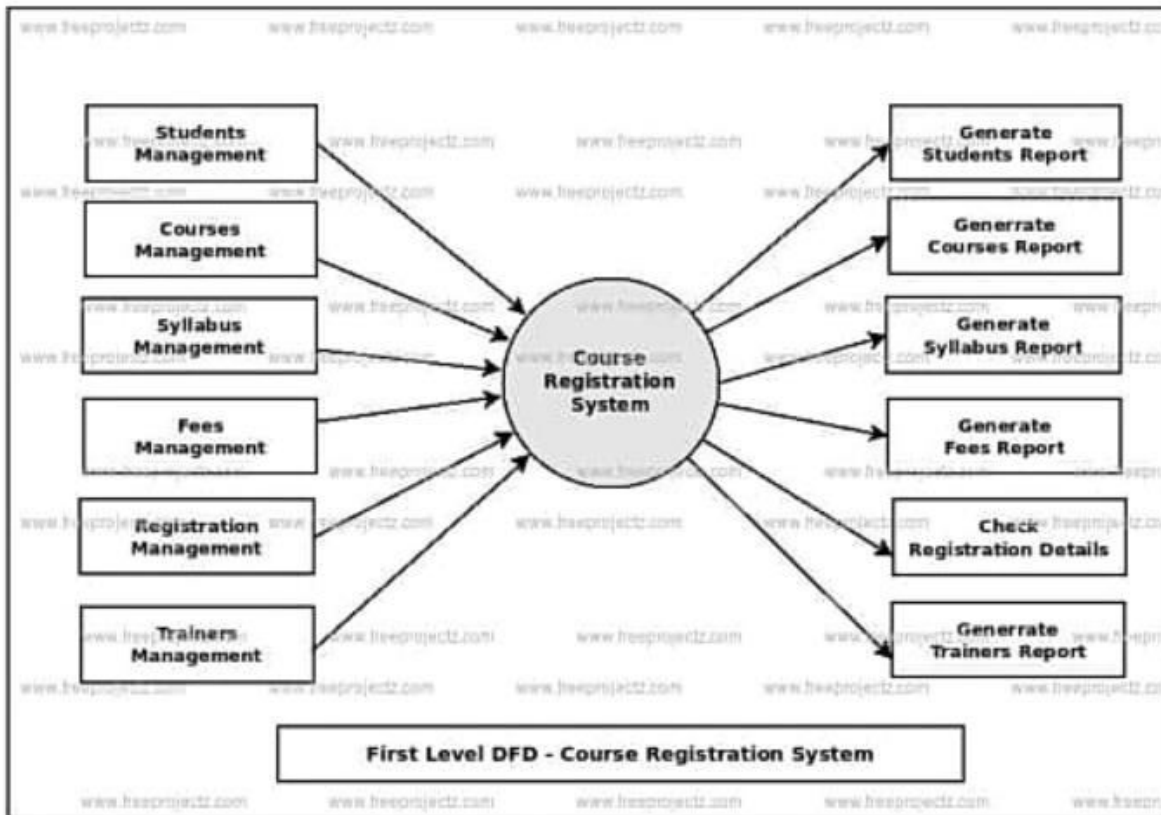
Second Level Data Flow Diagram(2nd Level DFD) Of Course Registration System:

DFD Level 2 then goes one step deeper into parts of Level 1 of Course. It may require more functionalities of Course to reach the necessary level of detail about the Course functioning. First Level DFD (1st Level) of Course Registration System shows how the system is divided into sub-systems (processes). The 2nd Level DFD contains more details of Login, Registration, Trainers, Students, Syllabus, Fees, Course.

Low level functionalities of Course Registration System

• Admin logins to the system and manage all the

functionalities of Course Registration System • Admin can add, edit, delete and view the records of

Course, Syllabus, Trainers, Login

• Admin can manage all the details of Fees, Students,

Registration • Admin can also generate reports of Course, Fees, Syllabus, Students, Trainers, Registration • Admin can search the details of Fees,

Trainers, Registration

• Admin can apply different level of filters on report of Course, Students, Trainers

• Admin can tracks the detailed information of Fees, Syllabus, Students,, Trainers

www.freeprojectz.com

Second Level DFD - Course Registration System
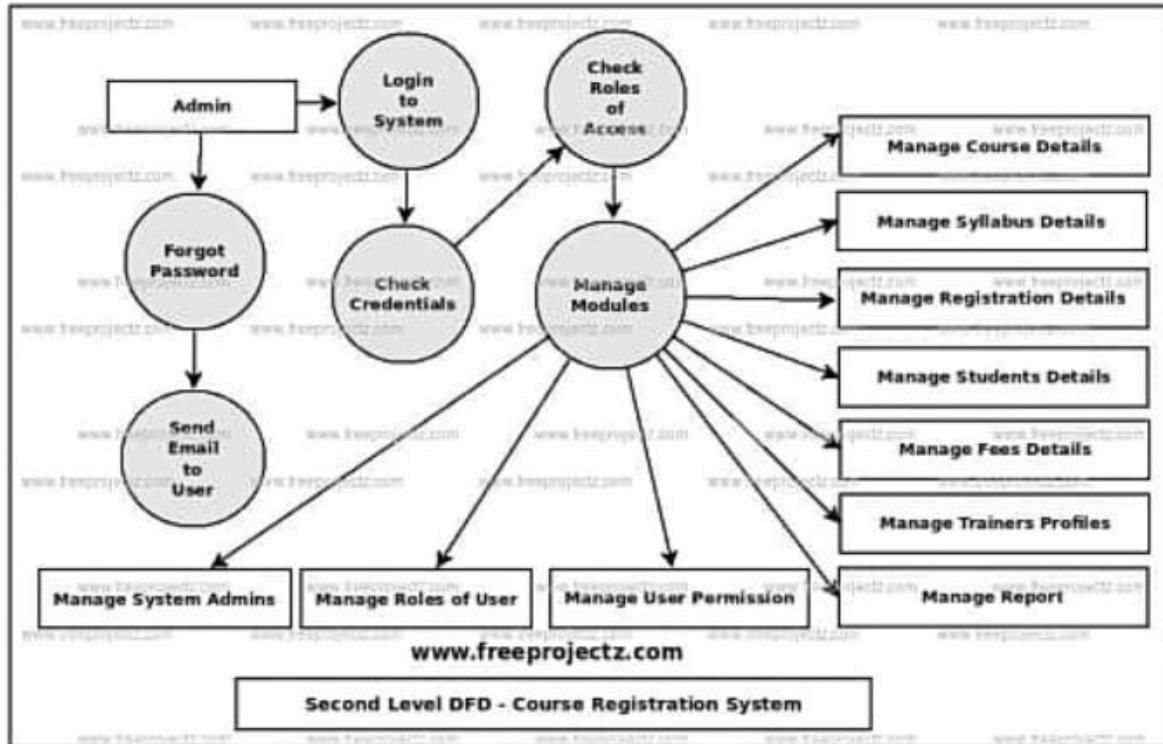
2.A)Draw E-R DIAGRAM FOR Students Marks Analyzing System.

This ER (Entity Relationship) Diagram represents the model of Result Management System Entity. The entity-re lationship diagram of Result Management System shows all the visual instrument of database tables and the rela tions between Class, Exam, Student, Teacher etc. It used structure data and to define the relationships between structured data groups of Result Management System functionalities. The main entities of the Result Manage ment System are Student, Class, Subject, Exam, Result and Teacher.

Result Management System entities and their attributes:

• Student Entity: Attributes of Student are student_id, student college_id, student_name, student_mobile, student email, student_username, student_password, student_address

• Class Entity: Attributes of Class are class_id, class_student_id, class_name, class_room, class_type, class_description

• Subject Entity : Attributes of Subject are subject_id, subject_course_id, subject_student_id, subject_name, subject type, subject_description

• Exam Entity: Attributes of Exam are exam_id, exam_student_id, exam_roll_number, exam_date exam_name, exam_type, exam_description • Result Entity: Attributes of Result are result_id,

result_student_id, result_name, result_description • Teacher Entity: Attributes of Teacher are teacher_id, teacher college_id, teacher_name, teacher_mobile, teacher_email, teacher_username, teacher_password, teacher_address

Description of Result Management System Database:

• The details of Student is store into the Student tables

respective with all tables • Each entity (Teacher, Subject, Result, Class, Student)

contains primary key and unique keys.

• The entity Subject, Result has binded with Student, Class entities with foreign key

• There is one-to-one and one-to-many relationships

available between Result, Exam, Teacher, Student • All the entities Student, Result, Subject, Teacher are

normalized and reduce duplicacy of records

• We have implemented indexing on each tables of Result Management System tables for fast query execution.

ER Diagram For Result Management System
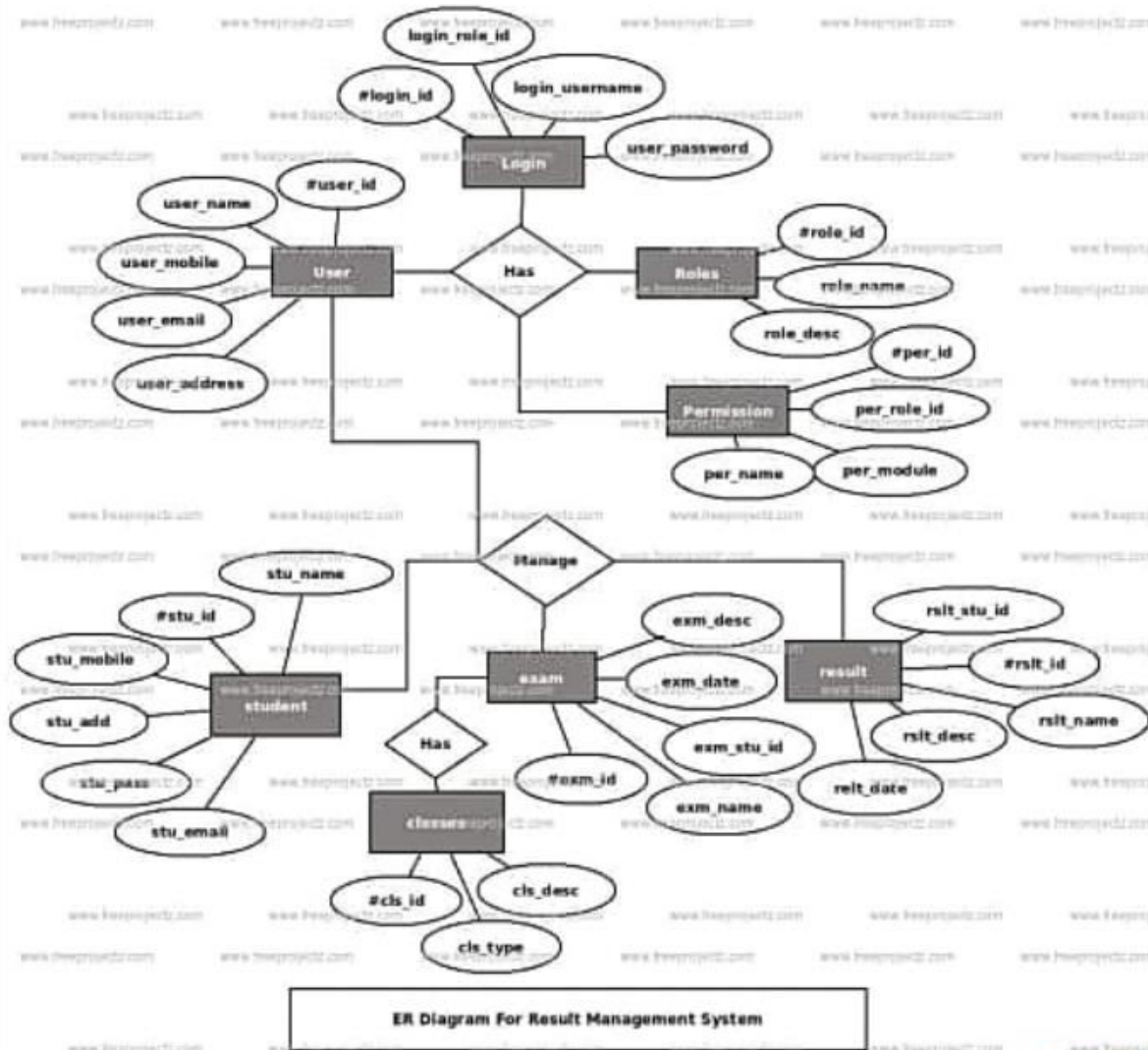
2.B) Draw DFD FOR Students Marks Analyzing System.

Result Management System Data flow diagram is often used as a preliminary step to create an overview of the Result Management without going into great detail, which can later be elaborated.it normally consists of overall ap plication dataflow and processes of the Result Manage ment process. It contains all of the userflow and their en tities such all the flow of Student, Exam, Class, Subject, Result, Teacher, Semester. All of the below diagrams has been used for the visualization of data processing and structured design of the Result Management process and working flow.

Zero Level Data Flow Diagram(0 Level DFD) Of Result Management System :

This is the Zero Level DFD of Result Management System, where we have eloborated the high level process of Result Management. It's a basic overview of the whole Result Management System or process being analyzed or modeled. It's designed to be an at-a-glance view of Result, Teacher and Semester showing the system as a single high-level process, with its relationship to external entities of Student,Exam and Class. It should be easily understood by a wide audience, including Student,

Class and Result In zero leve DFD of Result Management System, we have described the high level flow of the Result Management system.

High Level Entities and proccess flow of Result Management System:
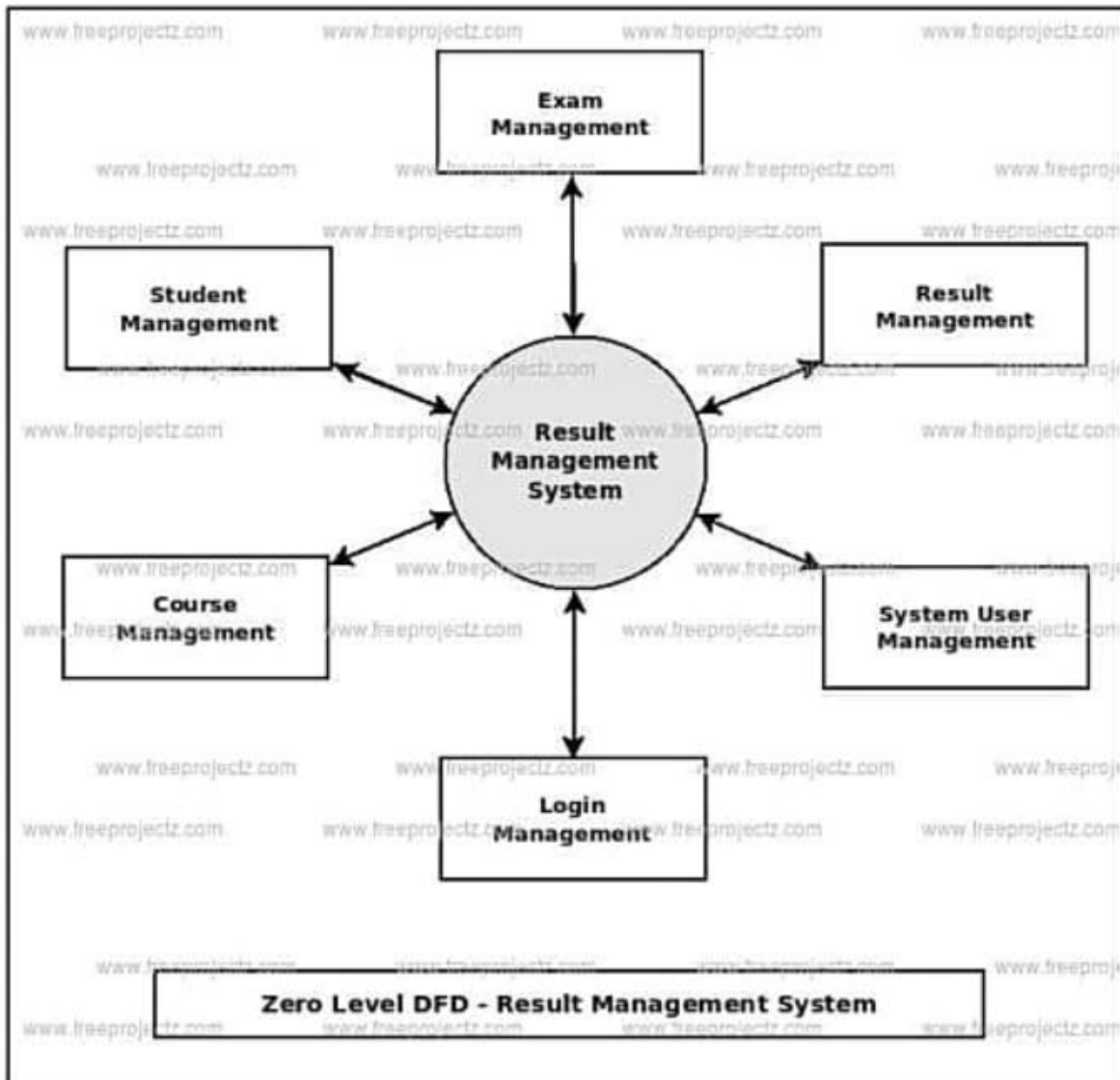
• Managing all the Student

• Managing all the Exam

• Managing all the Class

.Managing all the Subject

. Managing all the Result

. Managing all the Teacher

Managing all the Semester

Exam Management

Student Management

Result Management

Result Management System

Course Management

System User Management

Login Management

Zero Level DFD - Result Management System

First Level Data Flow Diagram(1st Level DFD) Of Result Management System :

First Level DFD (1st Level) of Result Management System shows how the system is divided into sub-systems (processes), each of which deals with one or more of the data flows to or from an external agent, and which together provide all of the functionality of the Result Management System system as a whole. It also identifies internal data stores of Semester, Teacher, Result, Subject, Class that must be present in order for the Result Management system to do its job, and shows the flow of data between the various parts of Student, Class, Teacher, Semester, Result of the system. DFD Level 1 provides a more detailed breakout of pieces of the 1st level DFD. You will highlight the main functionalities of Result Management.

Main entities and output of First Level DFD (1st Level DFD):

• Processing Student records and generate report of all Student

• Processing Exam records and generate report of all

Exam

• Processing Class records and generate report of all Class

• Processing Subject records and generate report of all

Subject

• Processing Result records and generate report of all Result

• Processing Teacher records and generate report of all

• Processing Semester records and generate report of all Semester

First Level DFD - Result Management System
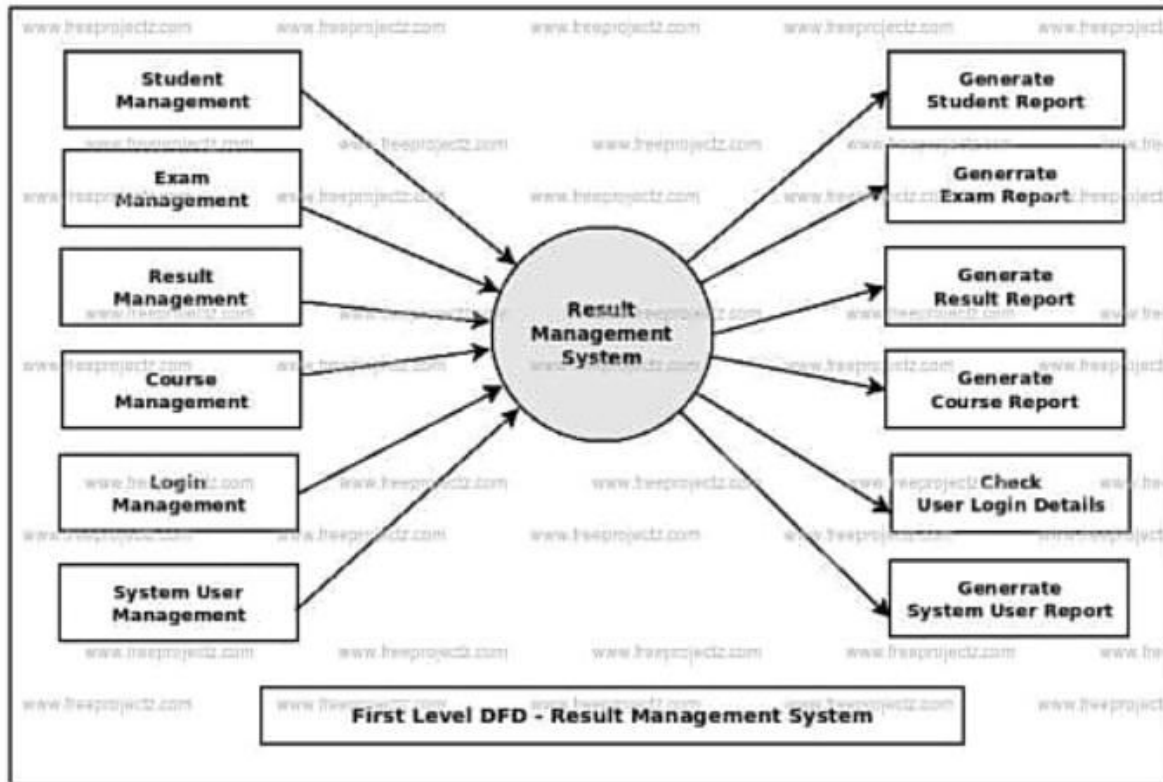
Second Level Data Flow Diagram(2nd Level DFD) Of Result Management System:

DFD Level 2 then goes one step deeper into parts of Level 1 of Result Management. It may require more functionalities of Result Management to reach the necessary level of detail about the Result Management functioning. First Level DFD (1st Level) of Result Management System shows how the system is divided into sub-systems (processes). The 2nd Level DFD contains more details of Semester, Teacher, Result, Subject, Class, Exam, Student.

Low level functionalities of Result Management System

• Admin logins to the system and manage all the functionalities of Result Management System • Admin can add, edit, delete and view the records of

Student, Class, Result, Semester • Admin can manage all the details of Exam, Subject, Teacher

• Admin can also generate reports of Student, Exam, Class, Subject, Result, Teacher

• Admin can search the details of Exam, Result, Teacher • Admin can apply different level of filters on report of

Student, Subject, Result

• Admin can tracks the detailed information of Exam, Class, Subject,, Result

Second Level DFD – Result Management System
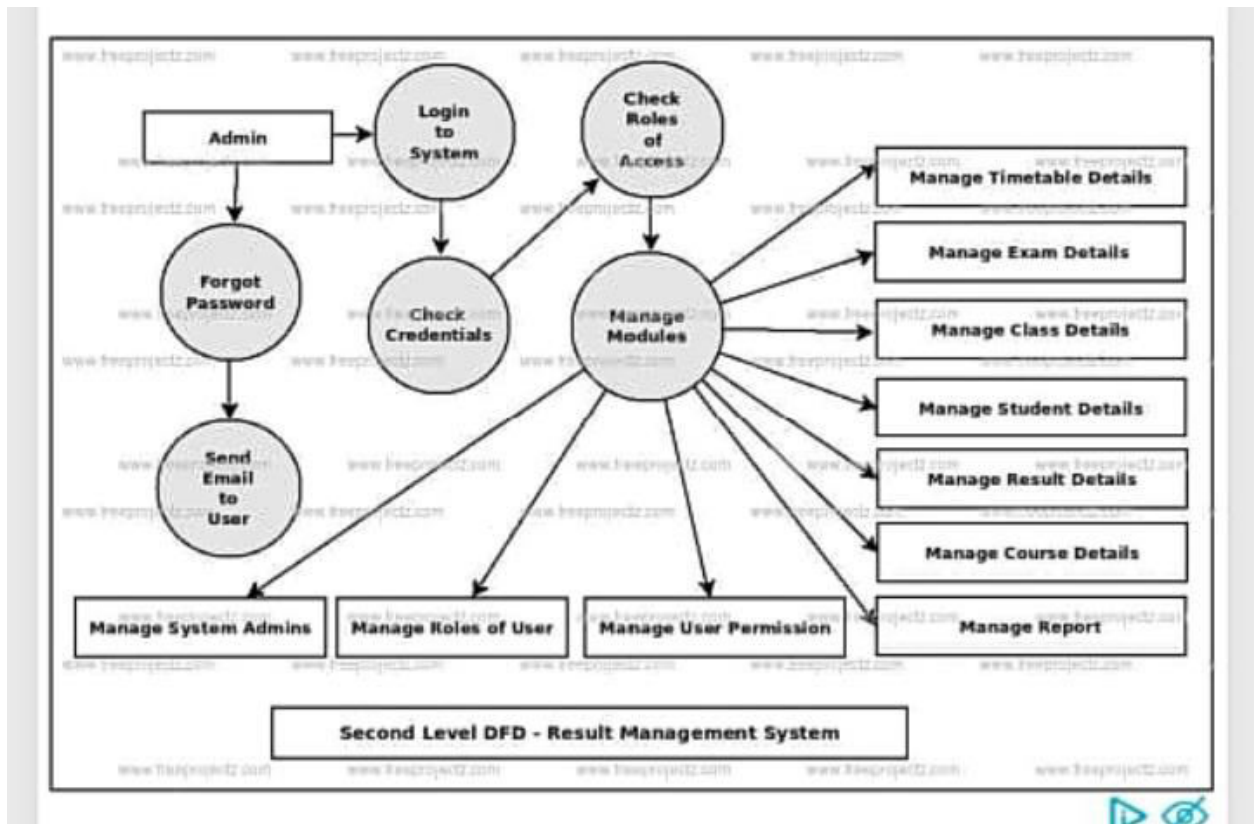
3.A) Draw E-R Diagram for OnlineTicket Reservation system.

This ER (Entity Relationship) Diagram represents the model of Ticket Reservation System Entity. The entity-rela tionship diagram of Ticket Reservation System shows all the visual instrument of database tables and the relations between Seats Availability, Stations, Trains, Passengers etc. It used structure data and to define the relationships between structured data groups of Ticket Reservation System functionalities. The main entities of the Ticket Re servation System are Trains, Seats Availability, Fare, Stations, Booking and Passengers.

Ticket Reservation System entities and their attributes:

• Trains Entity: Attributes of Trains are train_id, train_name, train_number, train_seat_number, train ticket, train_type, train_description

• Seats Availability Entity: Attributes of Seats Availability are seat_id, seat_train_id, seat customer_id, seat_number, seat_type, seat_description

• Fare Entity : Attributes of Fare are fare_id, fare_ticket_id, fare_title, fare_type, fare_description • Stations Entity: Attributes of Stations are station_id,

station_name, station_type, station_description

• Booking Entity: Attributes of Booking are booking_id, booking_ticket_id, booking_title, booking_type, booking_date, booking_description

• Passengers Entity: Attributes of Passengers are passenger_id, passenger_name, passenger_mobile, passenger_email, passenger_username, passenger_password, passenger_address

Description of Ticket Reservation System Database: • The details of Trains is store into the Trains tables

respective with all tables

• Each entity (Passengers, Fare, Booking, Seats

Availability, Trains) contains primary key and unique keys.

• The entity Fare, Booking has binded with Trains, Seats Availability entities with foreign key

• There is one-to-one and one-to-many relationships available between Booking, Stations, Passengers, Trains

. All the entities Trains, Booking, Fare, Passengers are normalized and reduce duplicacy of records

• We have implemented indexing on each tables of Ticket Reservation System tables for fast query execution.
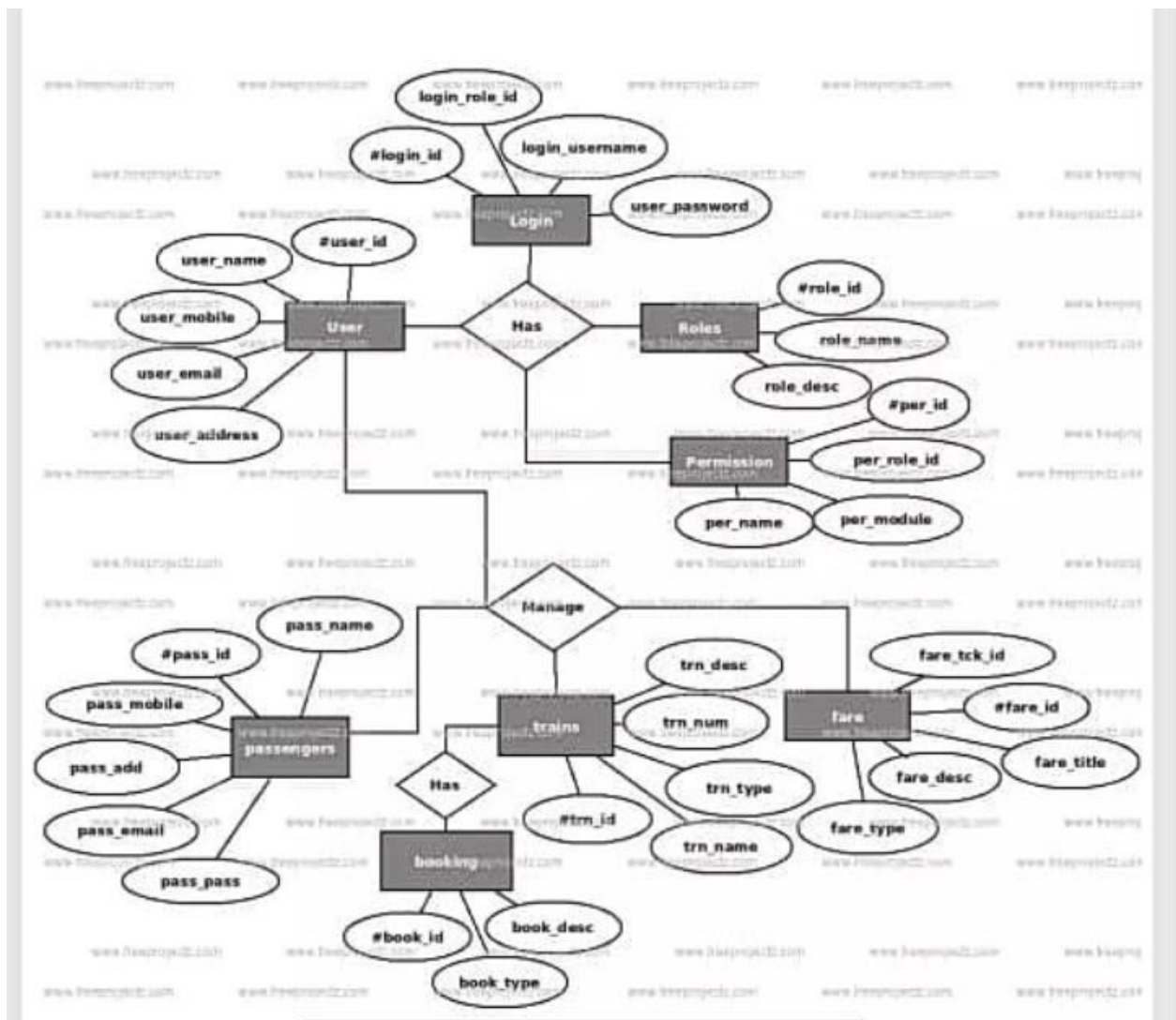
3.B) Draw DFD for OnlineTicket Reservation system.

Ticket Reservation System Data flow diagram is often used as a preliminary step to create an overview of the Ticket without going into great detail, which can later be elaborated.it normally consists of overall application data flow and processes of the Ticket process. It contains all of the userflow and their entities such all the flow of Trains, Seats, Fare, Stations, Booking, Passengers, Ticket. All of the below diagrams has been used for the visualiza tion of data processing and structured design of the Ticket process and working flow.

Zero Level Data Flow Diagram (0 Level DFD) Of Ticket Reservation System :

This is the Zero Level DFD of Ticket Reservation System, where we have eloborated the high level process of Ticket. It's a basic overview of the whole Ticket Reservation System or process being analyzed or modeled. It's designed to be an at-a-glance view of Booking,Passengers and Ticket showing the system as a single high-level process, with its relationship to external entities of Trains,Seats and Fare. It should be easily understood by a wide audience, including Trains,Fare and Booking In zero leve DFD of Ticket Reservation System, we have described the high level flow of the Ticket system.

High Level Entities and proccess flow of Ticket Reservation System:

• Managing all the Trains • Managing all the Seats

• Managing all the Fare

Managing all the Stations

• Managing all the Booking

Managing all the Passengers

Managing all the Ticket

Zero Level DFD - Ticket Reservation System
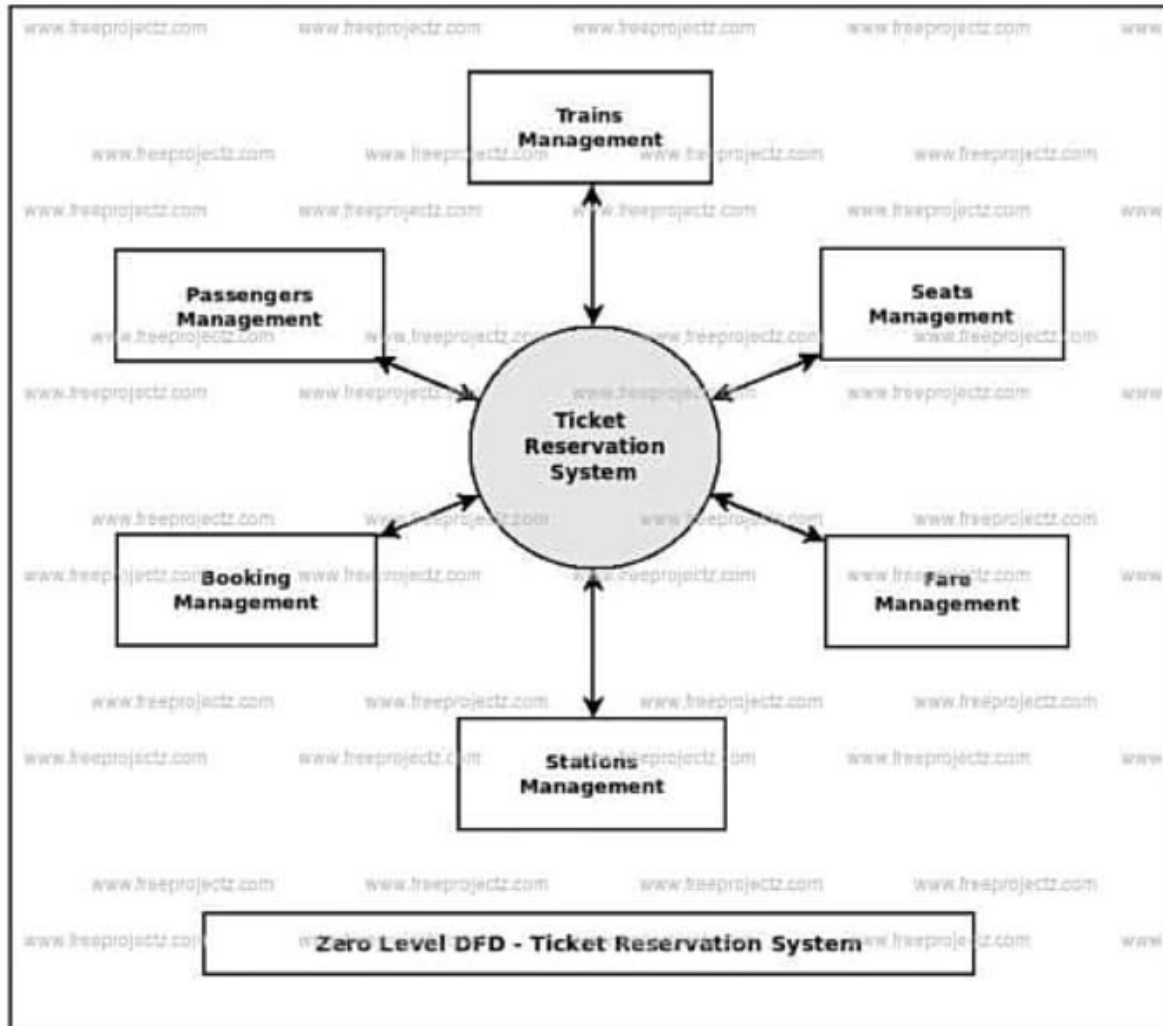
First Level Data Flow Diagram(1st Level DFD) Of Ticket Reservation System :

First Level DFD (1st Level) of Ticket Reservation System shows how the system is divided into sub-systems (processes), each of which deals with one or more of the data flows to or from an external agent, and which together provide all of the functionality of the Ticket Reservation System system as a whole. It also identifies internal data stores of Ticket, Passengers, Booking, Stations, Fare that must be present in order for the Ticket system to do its job, and shows the flow of data between the various parts of Trains, Fare, Passengers, Ticket, Booking of the system, DFD Level 1 provides a more detailed breakout of pieces of the 1st level DFD. You will highlight the main functionalities of Ticket.

Main entities and output of First Level DFD (1st Level DFD):

• Processing Trains records and generate report of all Trains

• Processing Seats records and generate report of all

Seats

• Processing Fare records and generate report of all

Fare

• Processing Stations records and generate report of all Stations • Processing Booking records and generate report of all

Booking

• Processing Passengers records and generate report of all Passengers

• Processing Ticket records and generate report of allTicket

First Level DFD - Ticket Reservation System
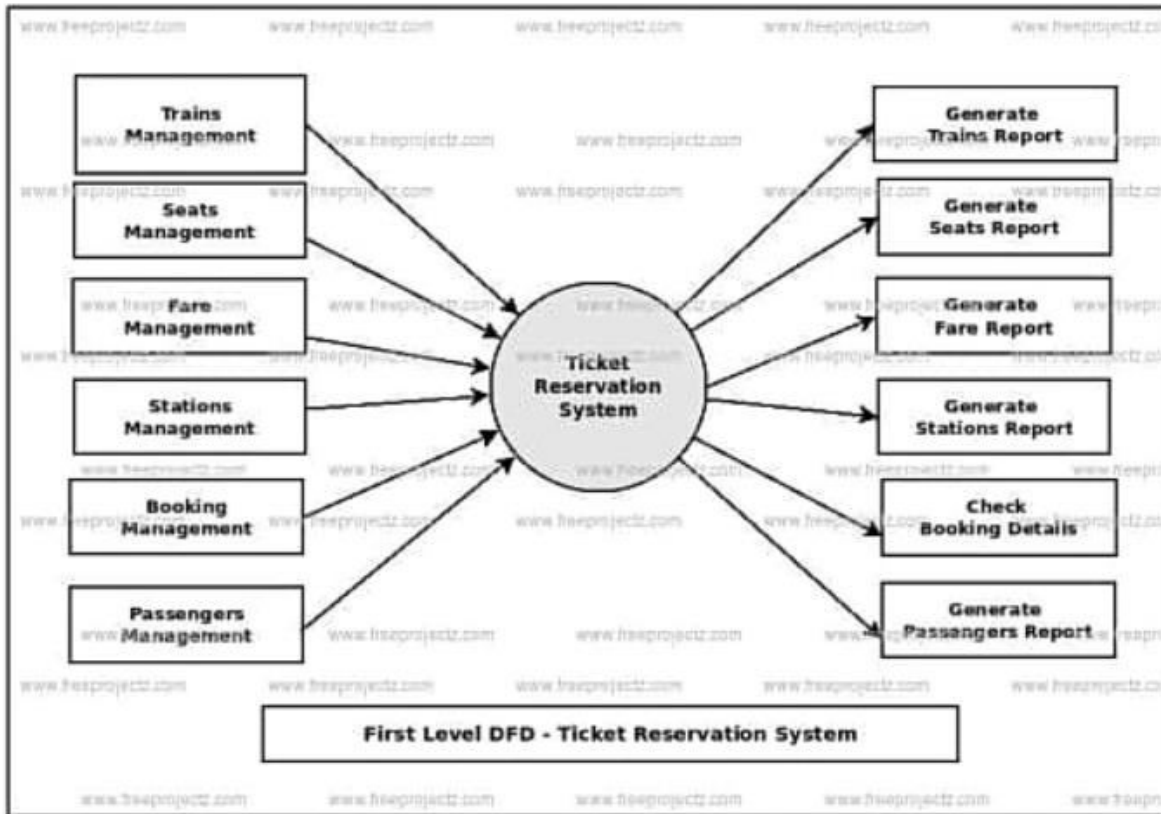
Second Level Data Flow Diagram (2nd Level DFD) Of Ticket ReservationSystem:

DFD Level 2 then goes one step deeper into parts of Level 1 of Ticket. It may require more functionalities of Ticket to reach the necessary level of detail about the Ticket functioning. First Level DFD (1st Level) of Ticket Reservation System shows how the system is divided into sub-systems (processes). The 2nd Level DFD contains more details of Ticket, Passengers, Booking, Stations, Fare, Seats, Trains.

Low level functionalities of Ticket Reservation System

• Admin logins to the system and manage all the

functionalities of Ticket Reservation System

• Admin can add, edit, delete and view the records of Trains, Fare, Booking, Ticket

• Admin can manage all the details of Seats, Stations,

Passengers • Admin can also generate reports of Trains, Seats, Fare,

Stations, Booking, Passengers

• Admin can search the details of Seats, Booking, Passengers

• Admin can apply different level of filters on report of

Trains, Stations, Booking • Admin can tracks the detailed information of Seats,

Fare, Stations,, Booking

www.freeprojectz.com

Second Level DFD - Ticket Reservation System
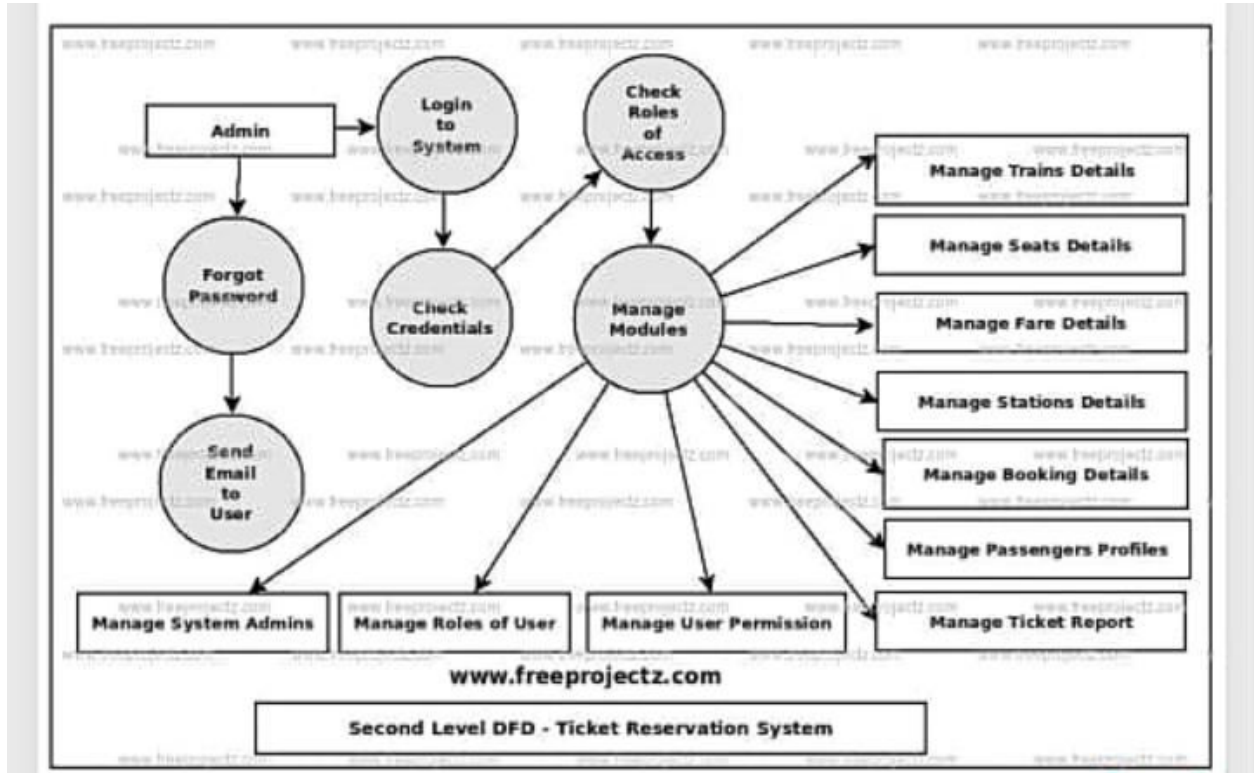
4.A) Draw E-R Diagram for Stock Maintenance system.

This ER (Entity Relationship) Diagram represents the model of Stock Management System Entity. The entity-re lationship diagram of Stock Management System shows all the visual instrument of database tables and the rela tions between Product, Bill, Stock, Store etc. It used struc ture data and to define the relationships between struc tured data groups of Stock Management System functionalities. The main entities of the Stock Manage ment System are Stock, Product, Product Quality, Bill, Cus tomer and Store.

Stock Management System entities and their attributes:

• Stock Entity: Attributes of Stock are stock_id, stock_items, stock_number, stock_type, stock_description

• Product Entity: Attributes of Product are product_id,

product_customer_id product_items, product_number,

product_type, product_description

• Product Quality Entity: Attributes of Product Quality are product_quality_id, product_quality_name, product_quality_type, product_quality_description

• Bill Entity: Attributes of Bill are bill_id, bill customer_id, bill_number, bill_type, bill_receipt, bill_description

• Customer Entity: Attributes of Customer are customer_id, customer_name, customer_mobile, customer_email, customer_username, customer_password, customer_address

• Store Entity: Attributes of Store are store_id, store_name, store_type, store_description

Description of Stock Management System Database:

• The details of Stock is store into the Stock tables respective with all tables

• Each entity (Store, Product Quality, Customer, Product, Stock) contains primary key and unique keys.

• The entity Product Quality, Customer has binded with Stock, Product entities with foreign key

• There is one-to-one and one-to-many relationships

available between Customer, Bill, Store, Stock

• All the entities Stock, Customer, Product Quality, Store are normalized and reduce duplicacy of records

• We have implemented indexing on each tables of Stock Management System tables for fast query execution.

ER Diagram For Stock Management System
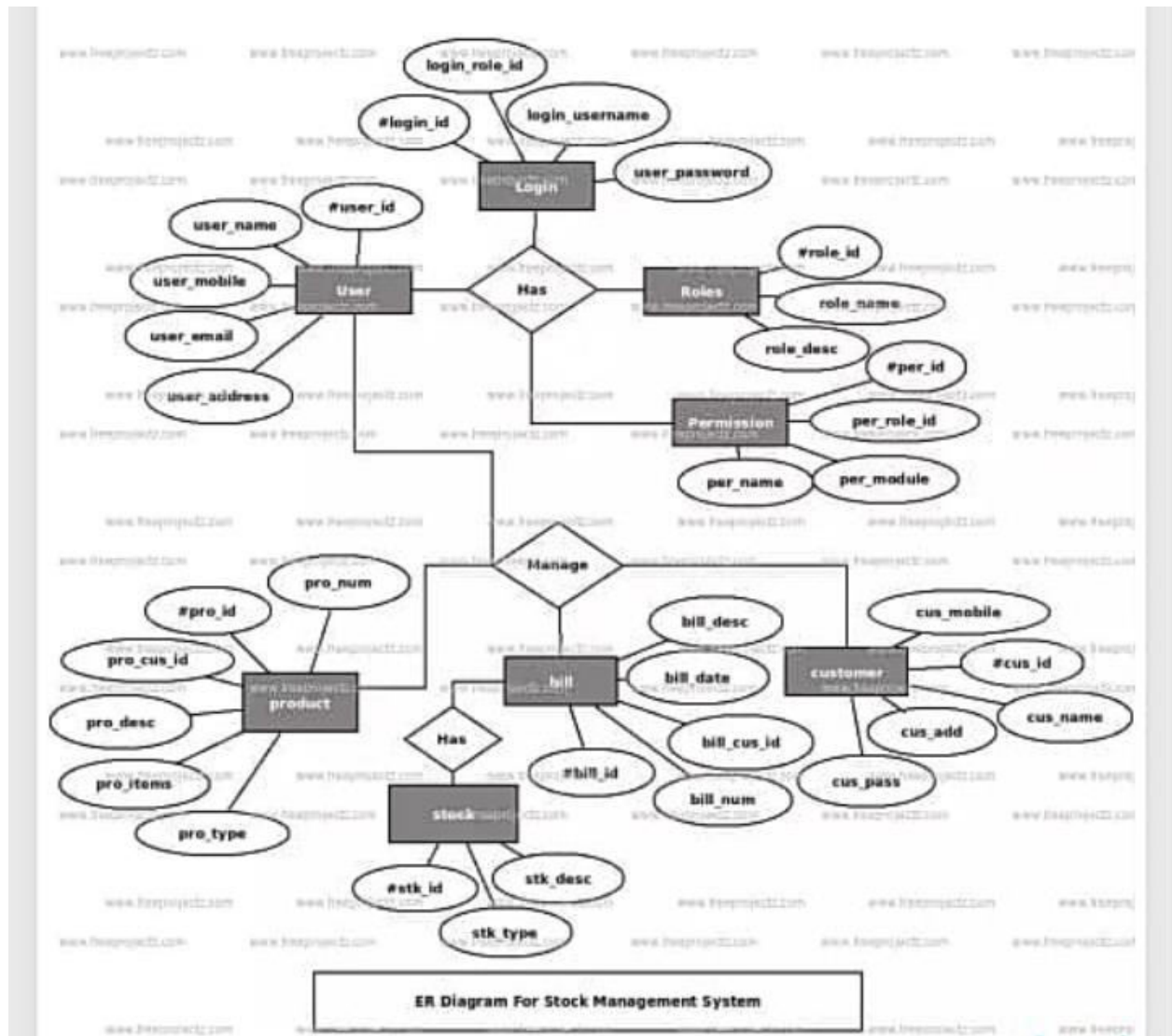
4.B) Draw DFD for Stock Maintenance system.

Stock Management System Data flow diagram is often used as a preliminary step to create an overview of the Stock without going into great detail, which can later be elaborated.it normally consists of overall application data flow and processes of the Stock process. It contains all of the userflow and their entities such all the flow of Stock, Product, Product Quality, Bill, Customer, Store, Login. All of the below diagrams has been used for the visualization of data processing and structured design of the Stock pro cess and working flow.

Zero Level Data Flow Diagram (0 Level DFD) Of Stock Management System:

This is the Zero Level DFD of Stock Management System, where we have eloborated the high level process of Stock. It's a basic overview of the whole Stock Management System or process being analyzed or modeled. It's designed to be an at-a-glance view of Customer,Store and Login showing the system as a single high-level process, with its relationship to external entities of Stock, Product and Product Quality. It should be easily understood by a wide audience, including Stock, Product Quality and Customer In zero leve DFD of Stock Management System, we have described the high level flow of the Stock system.

High Level Entities and proccess flow of Stock Management System:

• Managing all the Stock

Managing all the Product Managing all the Product Quality

• Managing all the Bills

• Managing all the Customer

Managing all the Store

High Level Entities and proccess flow of Stock Management System:

• Managing all the Stock

• Managing all the Product

• Managing all the Product Quality

• Managing all the Bill

• Managing all the Customer

• Managing all the Store

• Managing all the Login

Zero Level DFD - Stock Management System
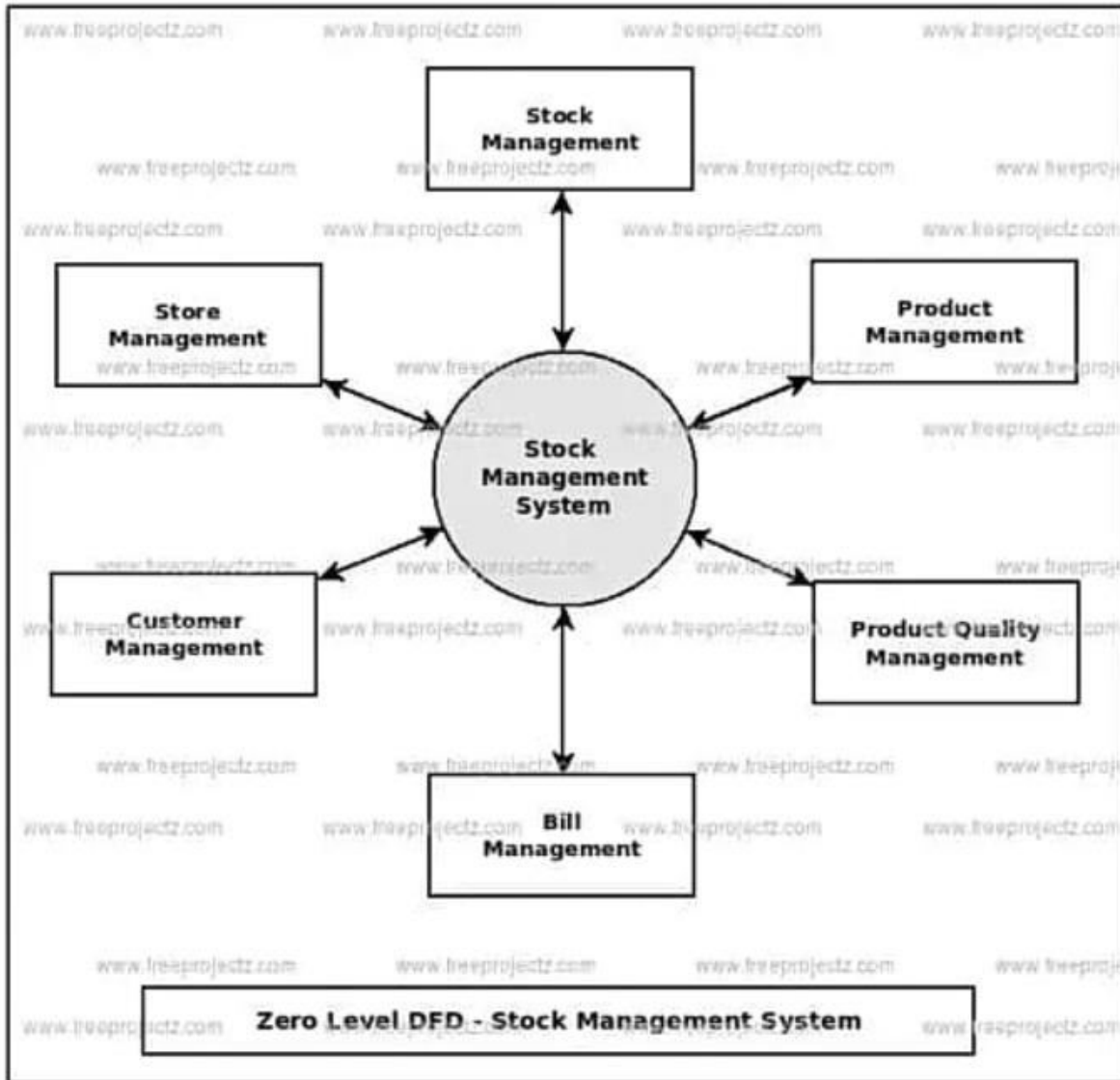
First Level Data Flow Diagram(1st Level DFD) Of Stock Management System :

First Level DFD (1st Level) of Stock Management System shows how the system is divided into sub-systems (processes), each of which deals with one or more of the data flows to or from an external agent, and which together provide all of the functionality of the Stock Management System system as a whole. It also identifies internal data stores of Login, Store, Customer, Bill, Product Quality that must be present in order for the Stock system to do its job, and shows the flow of data between the various parts of Stock, Product Quality, Store, Login, Customer of the system. DFD Level 1 provides a more detailed breakout of pieces of the 1st level DFD. You will highlight the main functionalities of Stock.

Main entities and output of First Level DFD (1st Level DFD):

• Processing Stock records and generate report of all Stock

• Processing Product records and generate report of all Product

• Processing Product Quality records and generate

report of all Product Quality

• Processing Bill records and generate report of all Bill

• Processing Customer records and generate report of all Customer

• Processing Store records and generate report of all Store

• Processing Login records and generate report of all Login

First Level DFD - Ticket Reservation System
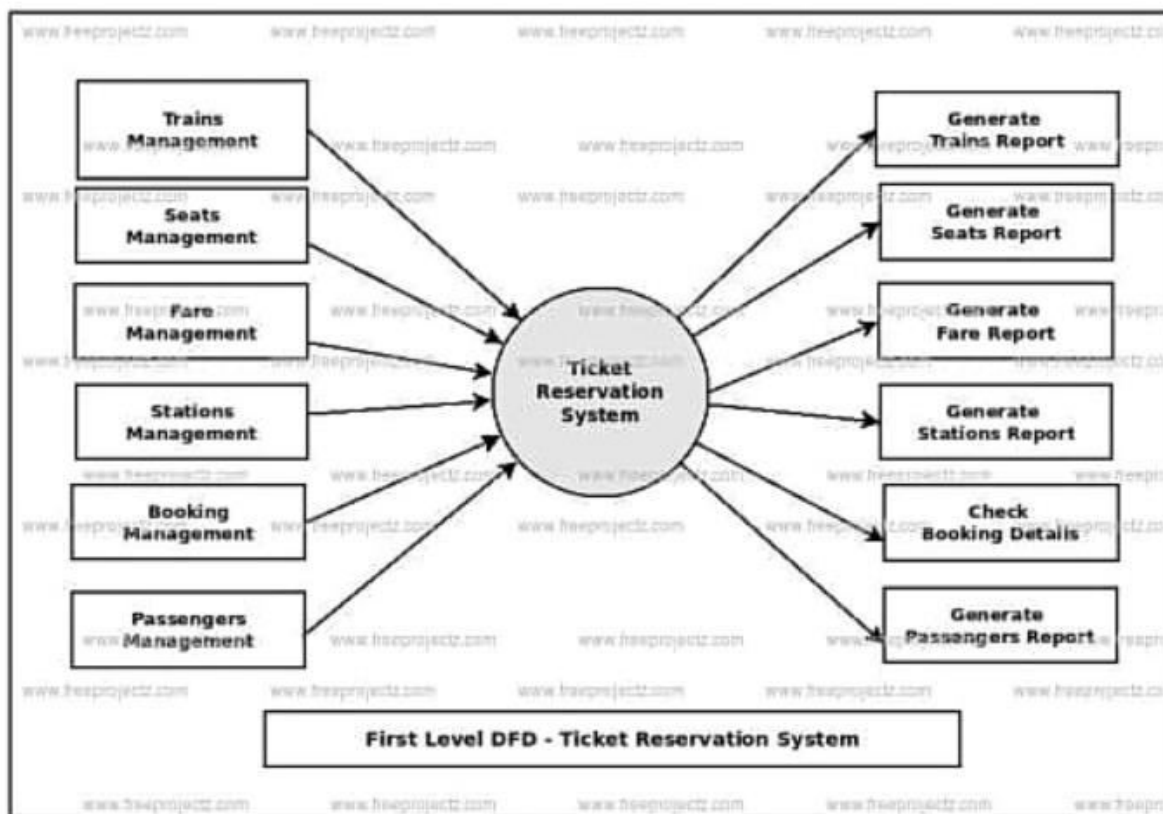
Second Level Data Flow Diagram (2nd Level DFD) Of Stock ManagementSystem:

DFD Level 2 then goes one step deeper into parts of Level 1 of Stock. It may require more functionalities of Stock to reach the necessary level of detail about the Stock functioning. First Level DFD (1st Level) of Stock Management System shows how the system is divided into sub-systems (processes). The 2nd Level DFD contains more details of Login, Store, Customer, Bill, Product Quality, Product, Stock.

Low level functionalities of Stock Management System

• Admin logins to the system and manage all the

functionalities of Stock Management System • Admin can add, edit, delete and view the records of

Stock, Product Quality, Customer, Login • Admin can manage all the details of Product, Bill, Store

• Admin can also generate reports of Stock, Product,

Product Quality, Bill, Customer, Store

• Admin can search the details of Product, Customer, Store

• Admin can apply different level of filters on report of

Stock, Bill, Customer

• Admin can tracks the detailed information of Product,

Product Quality, Bill,, Customer

Second Level DFD - Stock Management System

www.freeprojectz.com
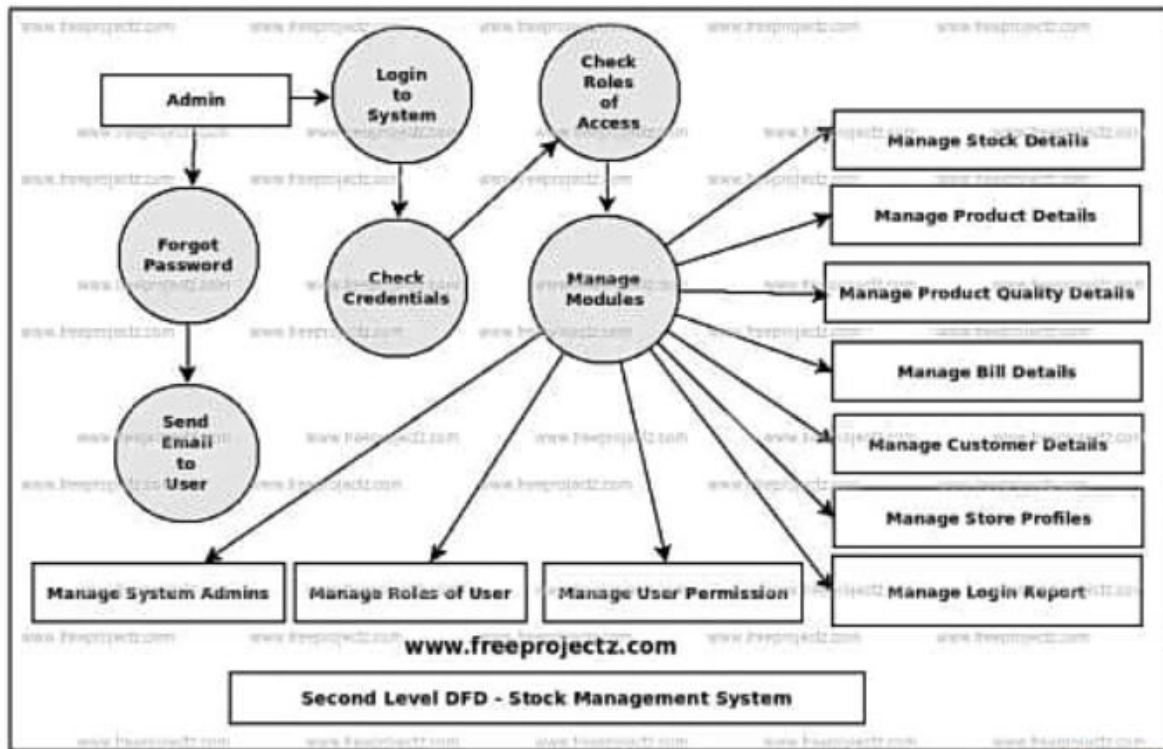
5.Attendance Management System Project using CONSTRUCTIVE COST ESTIMATION MODEL (COCOMO) Estimate the effort.

To estimate effort for developing Attendance Management System project using COCOMO model. The COCOMO model reflects your software development environment and produces more accurate estimates. It computes software development effort(cost) as a function of program size expressed in estimated lines of code(LOC).The main objective of basic COCOMO model gives an approximate estimate of the project parameters. Several COCOMO packages allow the user to estimate the tasks outside the scope using a percentage of the estimated software development effort.

Overview

The Constructive Cost Model (**COCOMO**) is an algorithmic software cost estimation model developed by Barry W. Boehm. The model uses a basic regression formula with parameters that are derived from historical project data and current as well as future project characteristics.

Boehm postulated that any software development project can be classified into one of the

following three categories based on the development complexity: organic, semidetached, and embedded. In order to classify a product into the identified categories, Boehm not only consideredthe characteristics of the product but also those of the development team and development environment. Roughly speaking, these three product classes correspond to application, utility and system programs, respectively. Normally, data processing programs are considered to be application programs. Compilers, linkers, etc., are utility programs. Operating systems and real- time system programs, etc. are system programs. System programs interact directly with the hardware and typically involve meeting timing constraints and concurrent processing.

Organic:

A development project can be considered of organic type, if the project deals with developing a well understood application program, the size of the development team is reasonably small, and the team members are experienced in developing similar types of projects.

Semi-detached:

A development project can be considered of semidetached type, if the development consists of a mixture of experienced and inexperienced staff. Team members may have limitedexperience on related systems but may be unfamiliar with some aspects of the system being developed.

Embedded:

A development project is considered to be of embedded type, if the software beingdeveloped is strongly coupled to complex hardware, or if the stringent regulations on the operational procedures exist.

The COCOMO cost estimation model is used by thousands of software project managers, and is based on a study of hundreds of software projects. Unlike other cost estimation models, COCOMO is an open model, so all of the details are published, including:

- The underlying cost estimation equations
- Every assumption made in the model (e.g. "the project will enjoy good management")
- Every definition (e.g. the precise definition of the Product Design phase of a project)
- The costs included in an estimate are explicitly stated (e.g. project managers are included, secretaries aren't)

The development time versus the product size in KLOC, it can be observed that the development time is a sub linear function of the size of the product, i. e. when the size of the product increases by two times, the time to develop the product does not double but rises moderately. This can be explained by the fact that for larger products, a larger number of activities which can be carried out concurrently can be identified. The parallel activities can be carried out simultaneously by the engineers. This reduces the time to complete the project. It canbe observed that the development time is roughly the same for all the three categories of products.For example, a 60 KLOC program can be developed in approximately 18 months,

regardless of whether it is of organic, semidetached, or embedded type.From the effort estimation, the project cost can be obtained by multiplying the required effort by the manpower cost per month. But, implicit in this project cost computation is the assumption that the entire project cost is incurred on account of the manpower cost alone. In addition to manpower cost, a project would incur costs due to hardware and software required for the project and the company overheads for administration, office space, etc.It is important to note that the effort and the duration estimations obtained using the COCOMO model are called as nominal effort estimate andnominal duration estimate. The term nominal implies that if anyone tries to complete the project in a time shorter than the estimated duration, then the cost will increase drastically. But, if anyone completes the project over a longer period of time than the estimated, then there is almost no decrease in the estimated cost value.

BASIC COCOMO MODEL: ☐

COCOMO consists of a hierarchy of three increasingly detailed and accurate forms.

1. The first level, Basic COCOMO is good for quick, early, rough order of magnitude estimatesof software costs, but its accuracy is limited due to its lack of factors to account for difference in project attributes (Cost Drivers).
2. Intermediate COCOMO takes these Cost Drivers into account and Detailed COCOMOadditionally accounts for the influence of individual project phases.
3. The Constructive Cost Model (COCOMO) is an algorithmic software cost estimation modeldeveloped by Barry Boehm. The model uses a basic regression formula, with parameters that are derived from historical project data and current project characteristics.

The basic COCOMO model gives an approximate estimate of the project parameters. The basic COCOMO estimation model is given by the following expressions: [

$$\text{Effort} = a1\text{x (KLOC)}^{\wedge}a2 \text{ pm}$$
$$\text{Tdev} = b1\text{x (Effort)}^{\wedge}b2$$
$$\text{MonthsP} = \text{Effort/ Tdev}$$

where

- KLOC is the estimated size of the software product expressed in Kilo Lines of Code.
- P is the no of persons required to complete the work .
- a1, a2 , b1, b2 are constants for each category of software products.
- Tdev is the estimated time to develop the software, expressed in months.
- Effort is the total effort required to develop the software product, expressed in person-months (PMs).

The coefficients a1, a2 , b1, b2 for various types of software projects

Software Projects	a1	a2	b1	b2
Organic	2.4	1.05	2.5	0.35
Semi-detached	3.0	1.12	2.5	0.32
Embedded	3.6	1.20	2.5	0.38

Estimation of development effort :-for the three classes of software products, the formulas for estimating the effort based on the code size are shown below:

Organic : Effort = 2.4(KLOC)^1.05PM

Semi-detached: Effort =

3.0(KLOC)^1.12PMEmbedded : Effort =

3.6(KLOC)^1.20PM

For the three classes of software products, the formulas for estimating the development timebased on the effort are given below:

Organic: Tdev = 2.5(Effort)^0.38Months

Semidetached: Tdev = 2.5(Effort)^0.35Months

Embedded: Tdev = 2.5(Effort)^0.32Months.

Example:-

Effort Calculation for Attendance Maintenance System.

Consider Lines of Code = 10000

   i.e value of KLOC is 10

Organic : Effort = $2.4(KLOC)^{1.05}$ PM

$$= 2.4*(10)^{1.05}$$

$$= 2.4 * 11.220$$

$$= 26.92 \text{ pm}$$

Semi-detached: Effort $= 3.0(10)^{1.12}$ PM

$$=3.0*13.18$$

$$=39.5 \text{ pm}$$

Embedded : Effort $= 3.6(10)^{1.20}$ PM

$$= 3.6*15.84$$

$$= 57.02 \text{PM}$$

6).Calculating effort for Attendance Management System using FP Function Point Oriented estimation model.

Objective

        Calculating effort for Attendance Management System using Function Point oriented estimation model. It is a method to break systems into smaller components, so they can be better understood and analyzed. It is used to express the amount of business functionality, an information system (as a product) provides to a user. Fps measure software size. They are widely accepted as an industry standard for functional sizing. Function points are used to compute a functional size measurement (FSM) of software. The cost (in dollars or hours) of a single unit is calculated from past projects.Function Point Analysis can provide a mechanism to track and monitor scope creep. Function Point Counts at the end of requirements, analysis, design, code, testing and implementation can be compared. The function point count at the end of requirementsand/or designs can be compared to function points actually delivered. The amount of growth is an indication of how well requirements were gathered by and/or communicated to the project team. If the amount of growth of projects declines over time it is a natural assumption that communication with the user has improved.

Overview

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 the software is counted. Inputs should be distinguished from inquiries, which are counted separately.

Number of user outputs: Each user output that provides application oriented information to the 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 storage media) that are used to transmit information 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, orcomplex. Nonetheless, the determination of complexity is somewhat subjective.

FPA provides different estimation mechanism within it for development and maintenance projects. (having different multiplication factors).This approach computes the total function points (FP) value for the project, by totaling the number of external user inputs, inquiries, outputs, and master files, and then applying the following weights: inputs , outputs , inquiries, and master files.

FP POINTS COMPUTATION

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

FP = count total $[0.65 + 0.01 \Sigma (Fi)]$

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 followingquestions :

1. Does the system require reliable 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 screensor 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.

Effort = FP/ Person-Month

Count Total can be obtained using the following table

Domain characteristics	Count		Weighting factor			Count
			Simple	Average	Complex	
No of user input		*	3	4	6	
No of user output		*	4	5	7	
No of user queries		*	3	4	6	
No of files		*	7	10	15	
No of external interfaces		*	5	7	10	
Count total:						

Example:

Assume that....

Number of user input : 5

Number of user output : 5

Number of user enquires : 6

Number of files :5

Number of external interfaces : 5

Apply these assumptions on a **simple project** and calculate the Count Total

Domain characteristics	Count		Weighting factor			Count
			Simple	Average	Complex	
No of user input	5	*	3	4	6	15
No of user output	5	*	4	5	7	2
No of user queries	6	*	3	4	6	18
No of files	5	*	7	10	15	35
No of external interfaces	5	*	5	7	10	25
Count total:						113

Therefore Count Total =113

Now calculate the Functional Points using FP = count total *0.65 + 0.01 Σ (Fi)]

FP = count total *0.65 + 0.01 Σ (Fi)]

= 113 * (0.65 +0.01 * 25) where Σ (Fi)

=25

i.e the questions answered using a scale that ranges from 0 (not important or applicable) to 5 (absolutely essential) in total 14 questions

=113 *(0.65 + 0.25)

=113 * 0.9

= 101.7

FP = 101.7

Effort = FP / person-month

ADVANTAGES:

1. This method is independent of programming languages.

2. It is based on the data which can be obtained in early stage of project

3. Function Points are easily understood by the non technical user. This helps communicatesizing information to a user or customer.

DISADVANTAGES:

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

2. Many aspects of this method are not validated

3. The functional point has no significant ant meaning, it's just a numerical value.

7.A) AIM:

To model the "**Course Registration System**" using the software Rational Rose with various UML (Unified Modeling Language) diagrams.

ABSTRACT:

The course registration project helps the user to know the procedures followed in universities to enroll students for a particular course.

The limitation of the existing system are :the system is not efficient to meet the needs, the system is very slow and time consuming, not more than one choice can be given by a student, the number of seats left out and the course details cannot be known.

The problems that are overcome are the existing system uses spreadsheet as backend now the proposed system uses access as backend, the basic requirement of the system has being improved, its has become user friendly.

The proposed system is very efficient. Now a student can know the number of seats that are being allotted in a day. The person can even get the course detail. The report displays the number of students per course and the number of students allotted seat on a particular day.

SYSTEM ANALYSIS PHASE:

The proposed system offers a student to select course. The students can check the number of seats available. The students can know the details of the person who have been allotted seat on a day. Every time a seat is allotted the number of seats in the database gets reduced. The proposed system is being designed for counseling of students in single window system.
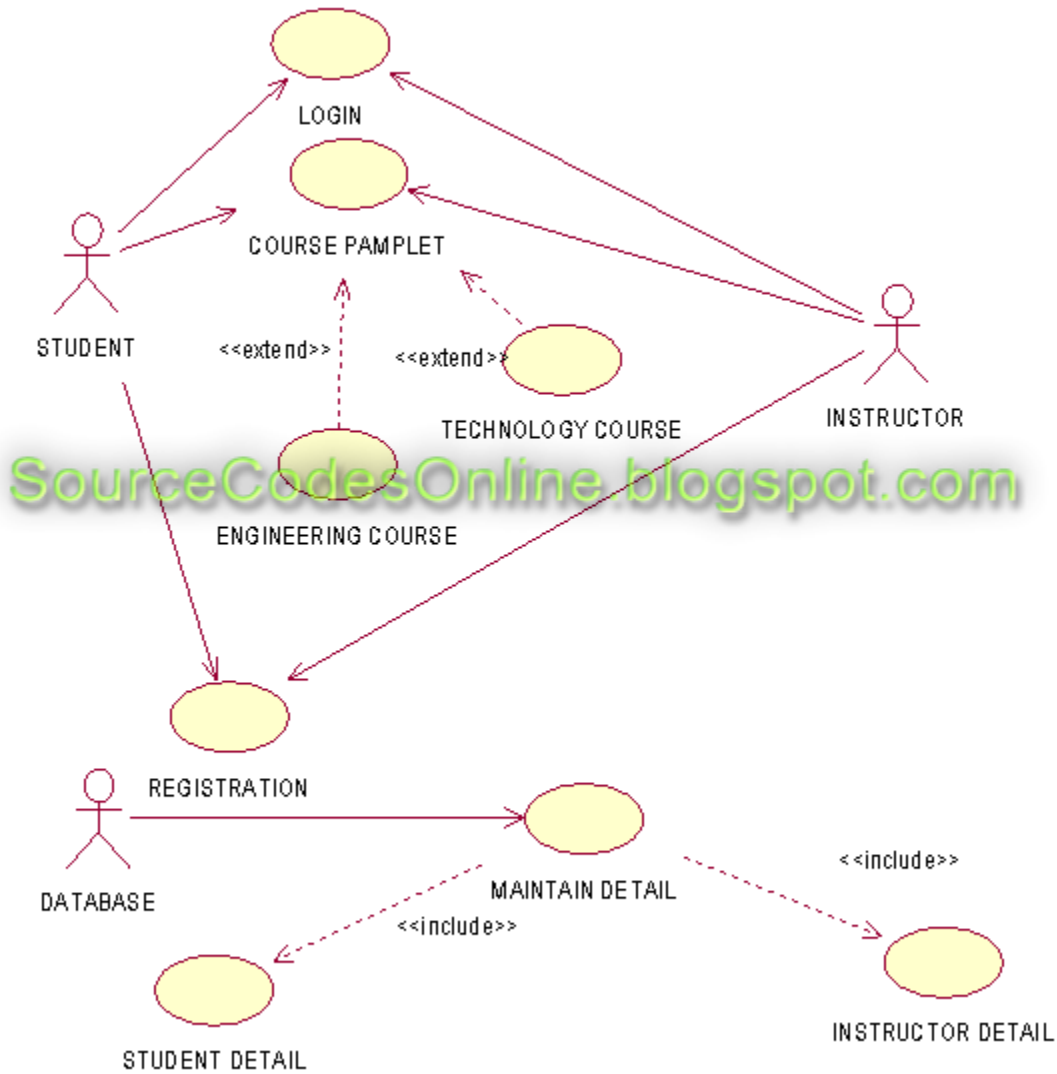
PRESENT SYSTEM:

The present system is inefficient and time consuming. The students can choose only one course if seat is available the student are allotted, else the student are not allotted and can also not give second choice. The student cannot know the number of seats left out. The backend used is an older version and the processing time is too slow.
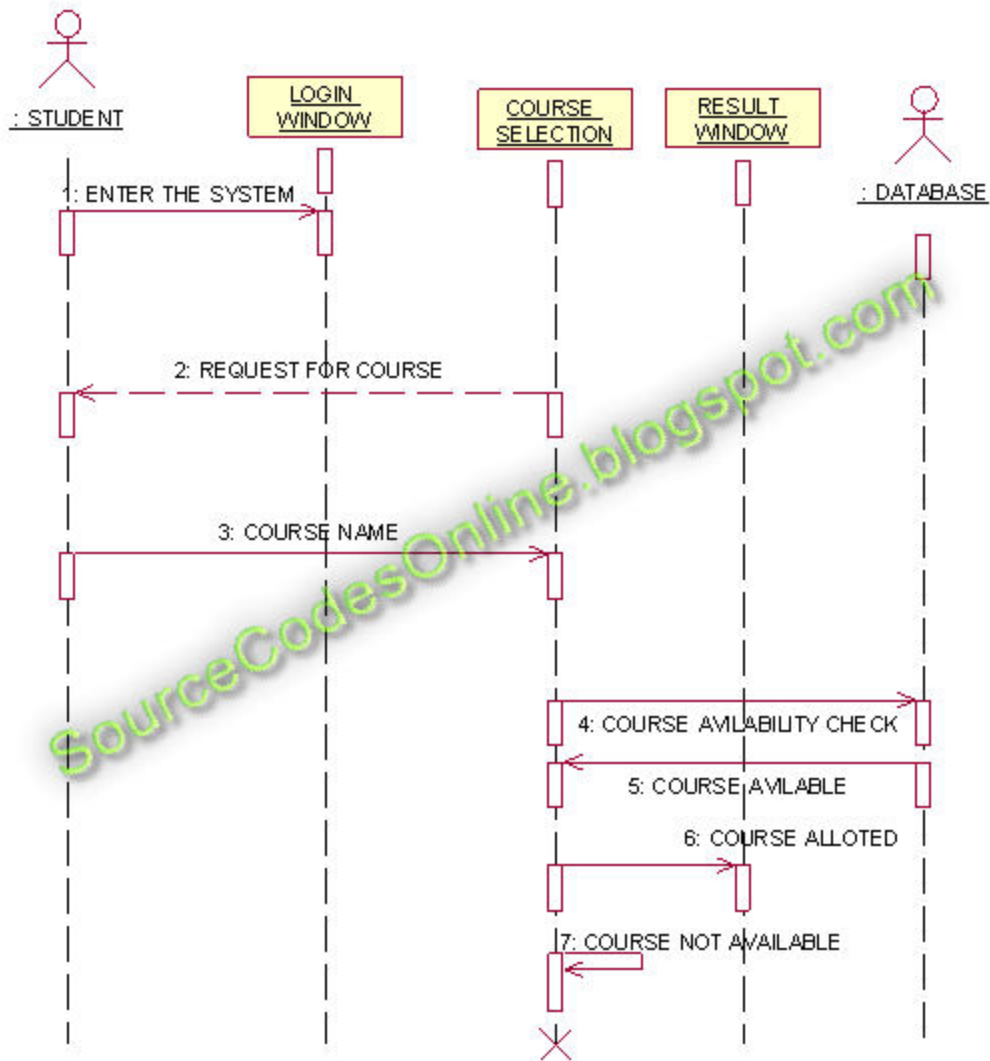
PROPOSED SYSTEM:

The system that is being developed is a user friendly system. The processing speed is very high when compared to the existing system. The space occupied by the system in the memory is also very less. The course details can be known. Second choice can be given if the course is not available. Date wise report and course wise report are generated.
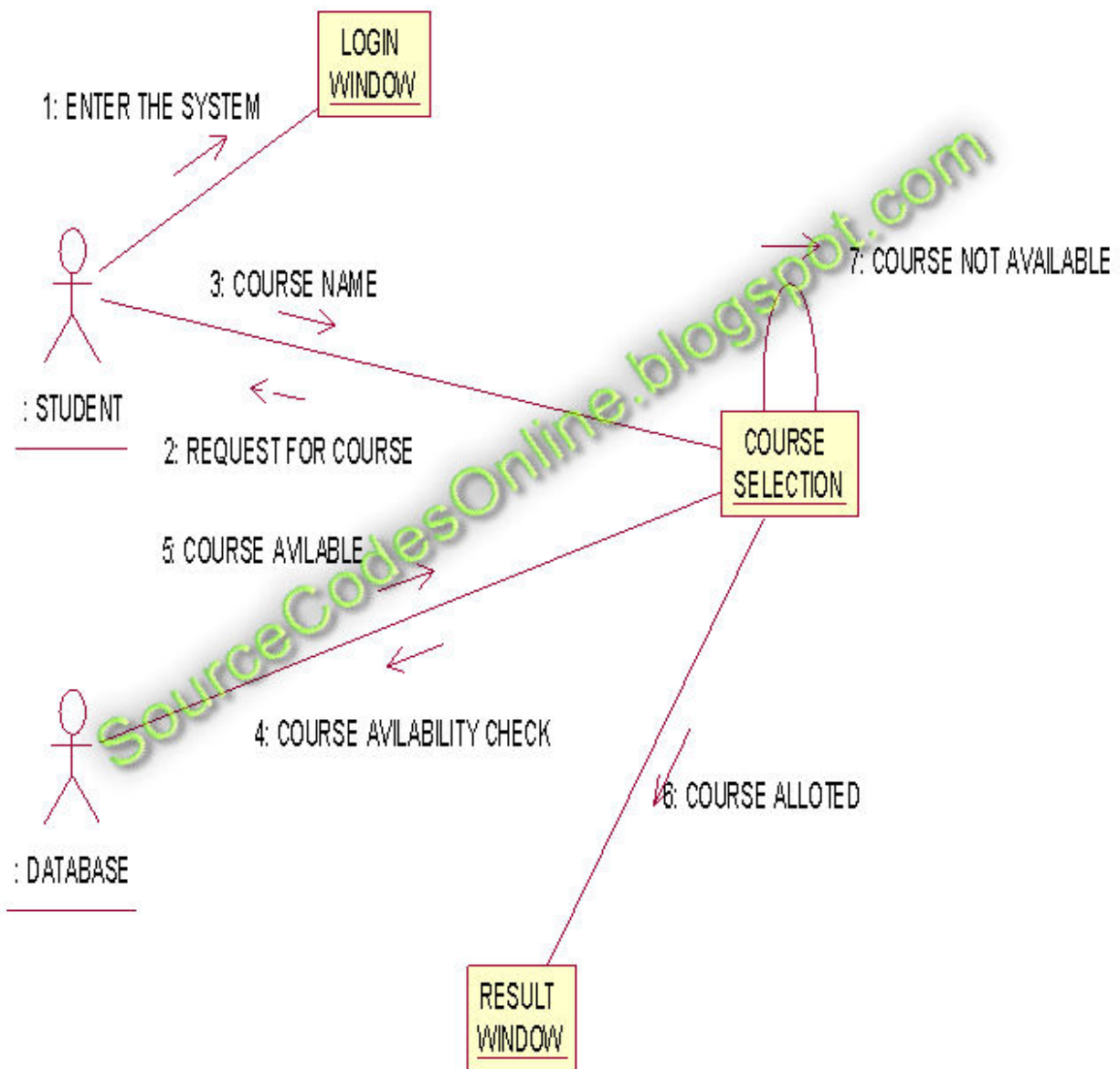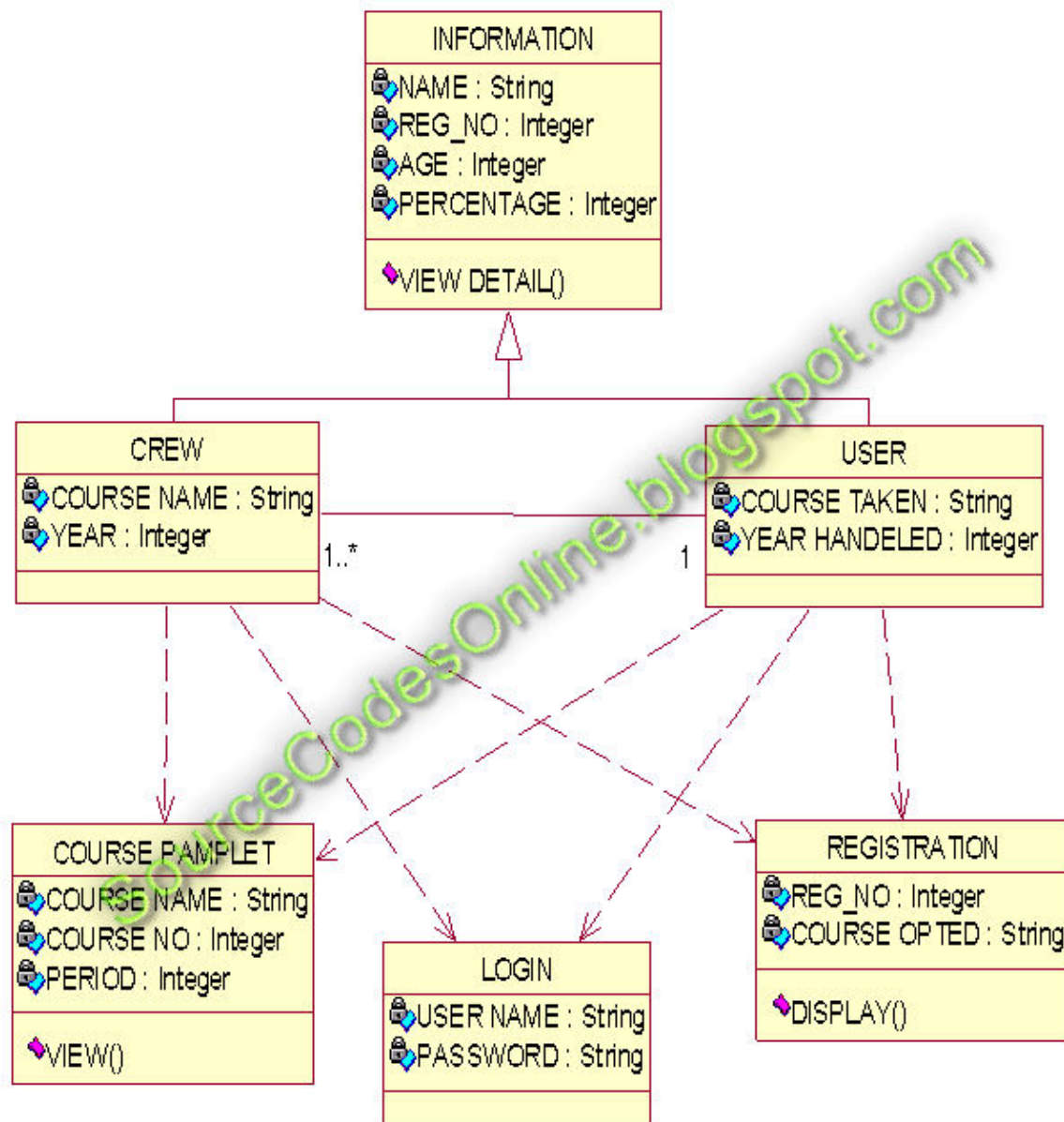
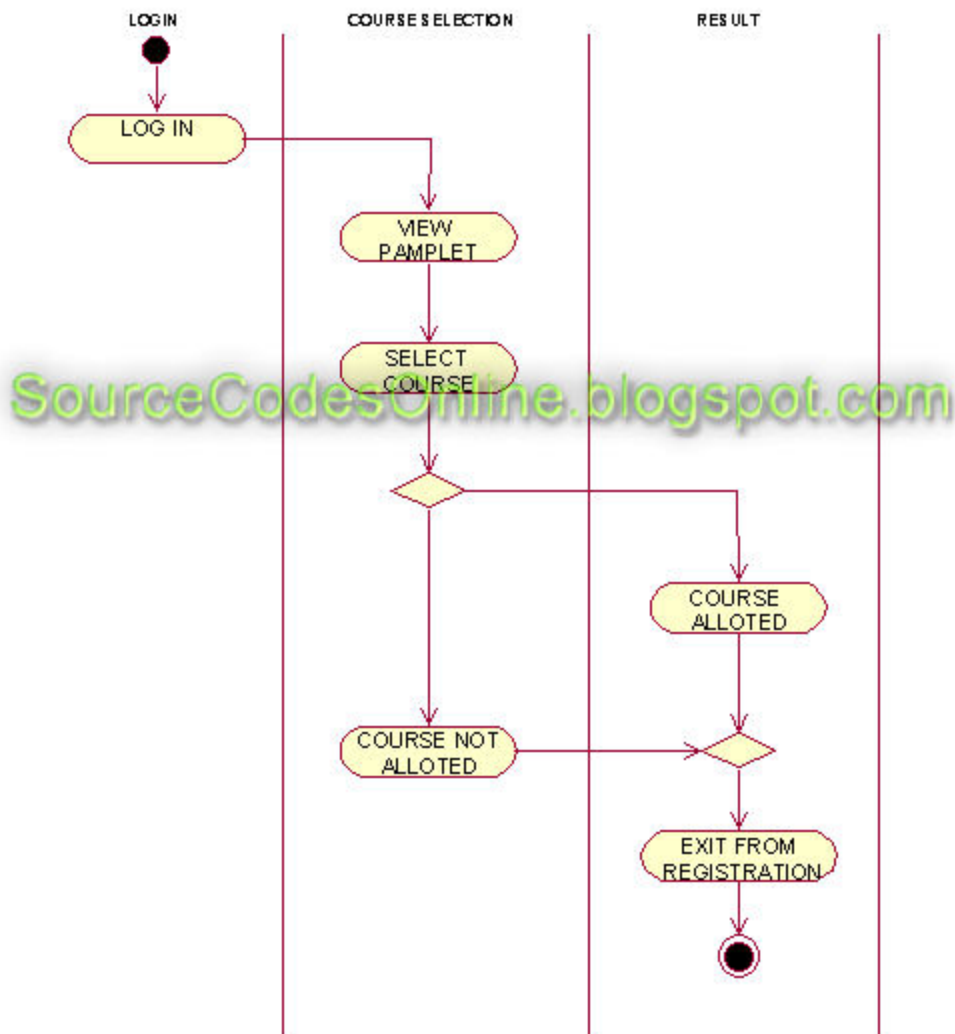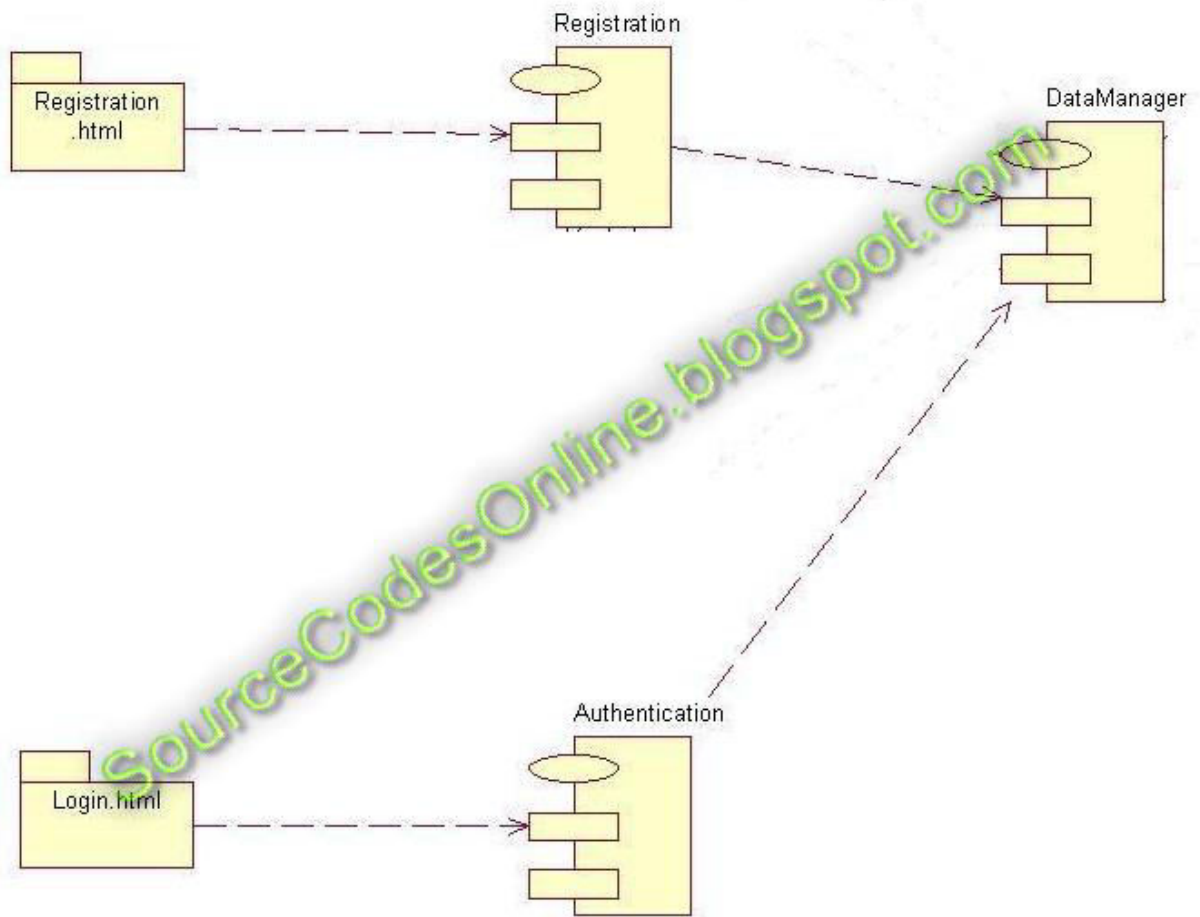**UML DIAGRAMS:**

USE CASE DIAGRAM:

SEQUENCE DIAGRAM:

COLLABORATION DIAGRAM:

LOGIN WINDOW

1: ENTER THE SYSTEM

7: COURSE NOT AVAILABLE

3: COURSE NAME

: STUDENT

2: REQUEST FOR COURSE

COURSE SELECTION

5: COURSE AVILABLE

4: COURSE AVILABILITY CHECK

6: COURSE ALLOTED

: DATABASE

RESULT WINDOW

CLASS DIAGRAM:

ACTIVITY DIAGRAM:

LOGIN	COURSE SELECTION	RESULT

LOG IN

VIEW PAMPLET

SELECT COURSE

COURSE ALLOTED

COURSE NOT ALLOTED

EXIT FROM REGISTRATION

COMPONENT DIAGRAM:

DEPLOYMENT DIAGRAM:
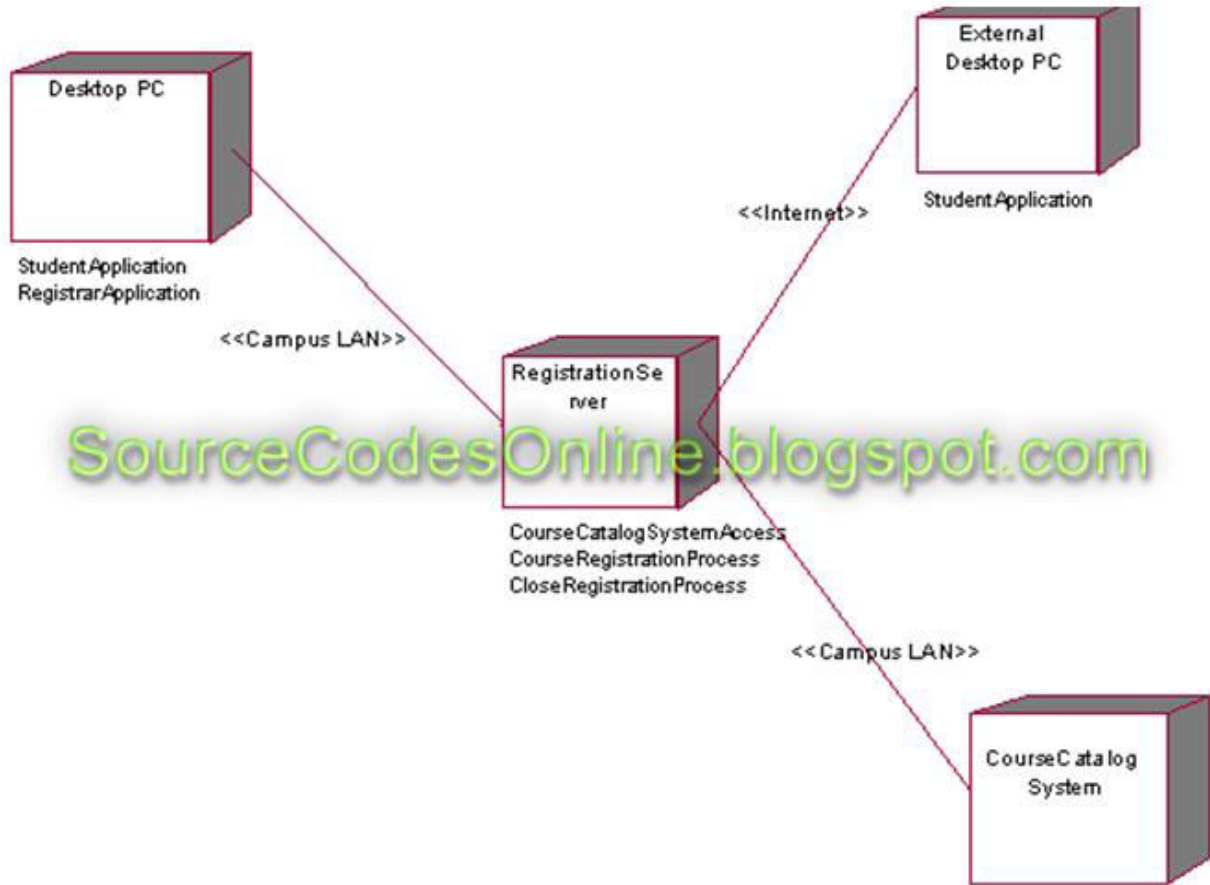
RESULT:
Thus the **Course Registration System** application was successfully designed and the output was verified.
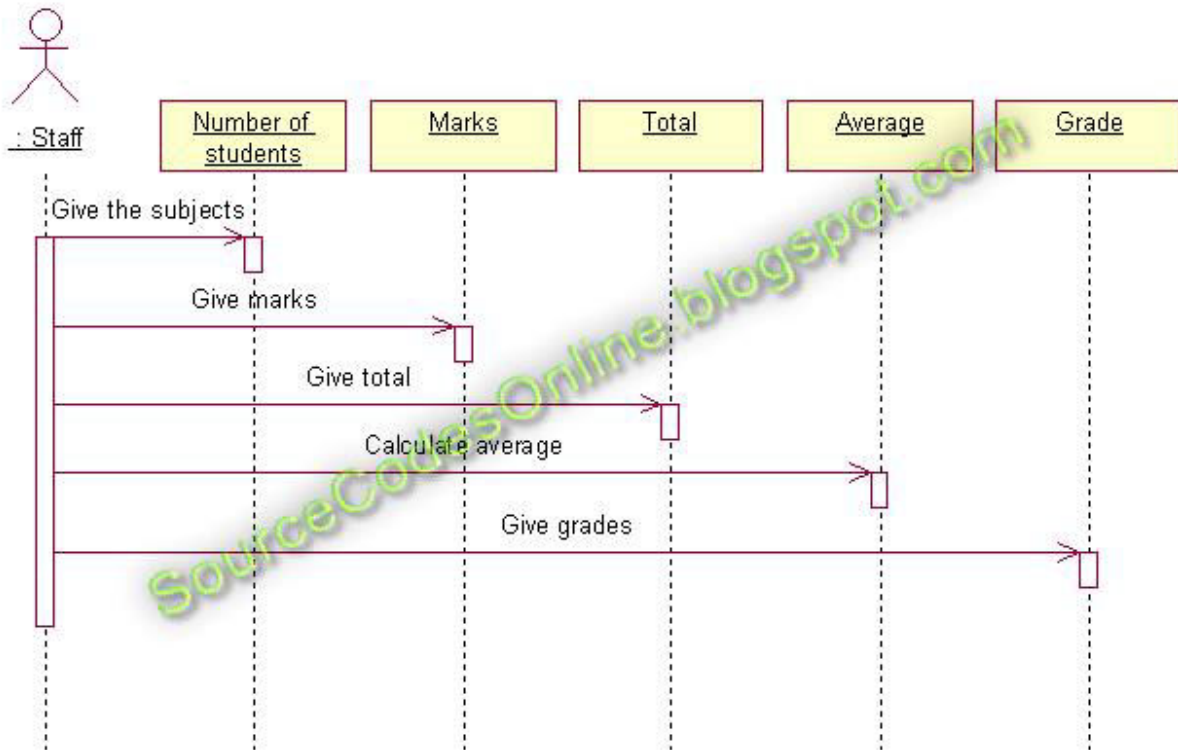
7.B) AIM:
    To model the "Student Marks Analysis System" using the software Rational Rose with various UML (Unified Modeling Language) diagrams.
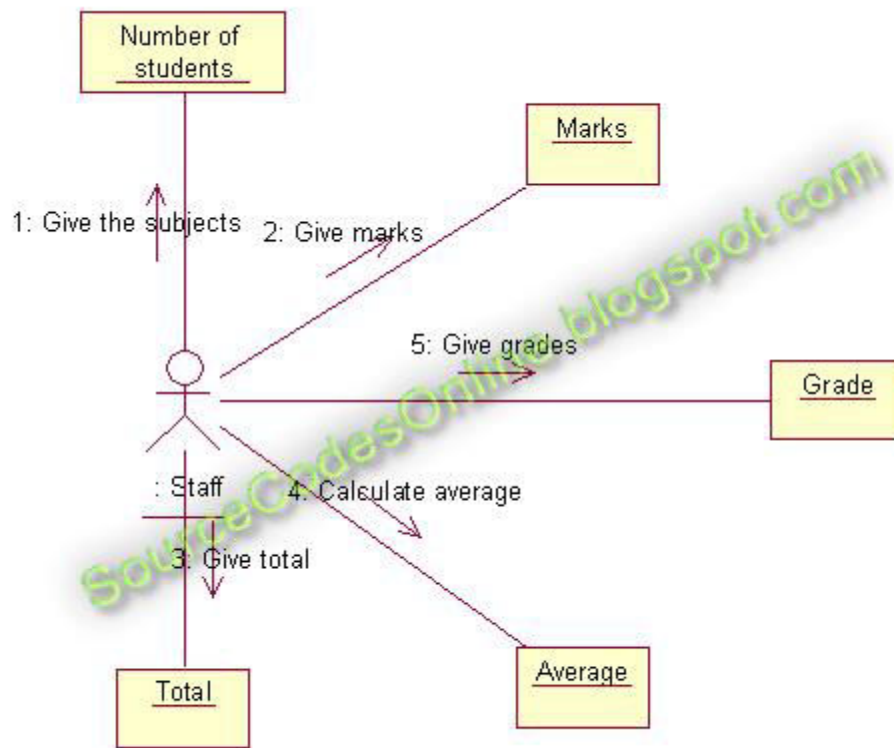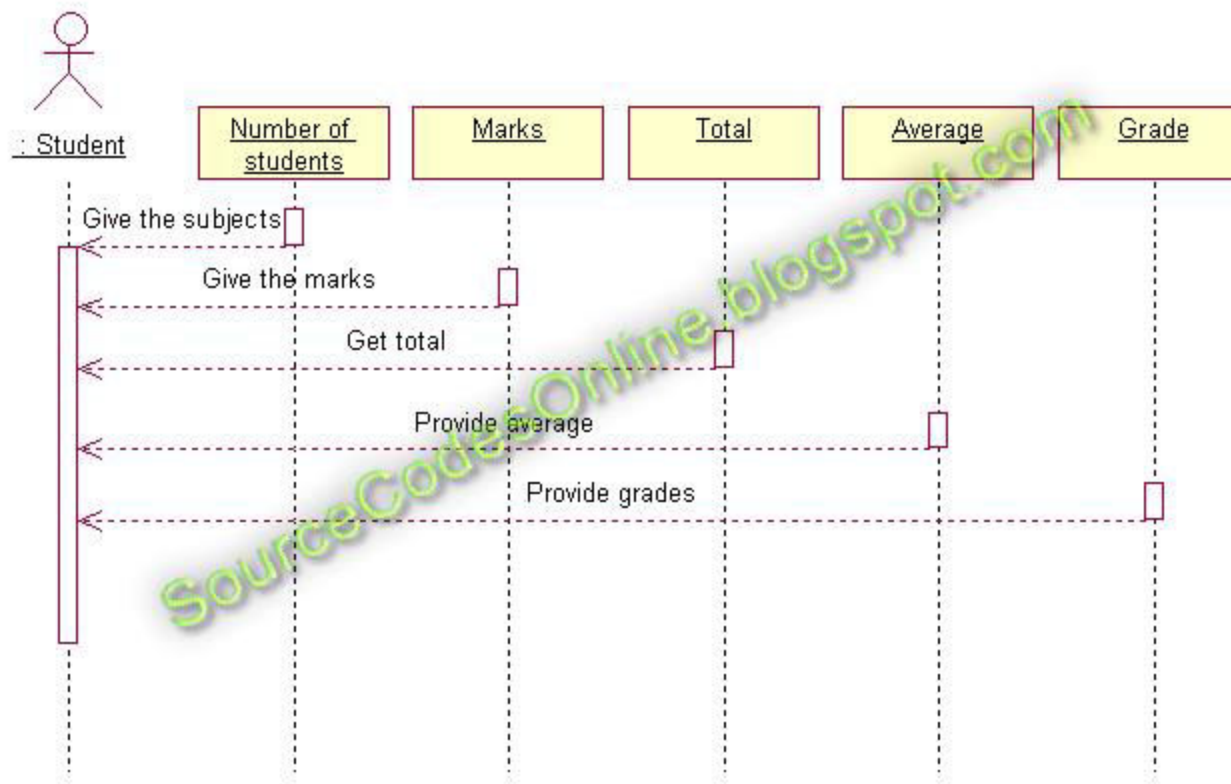
UML DIAGRAMS:

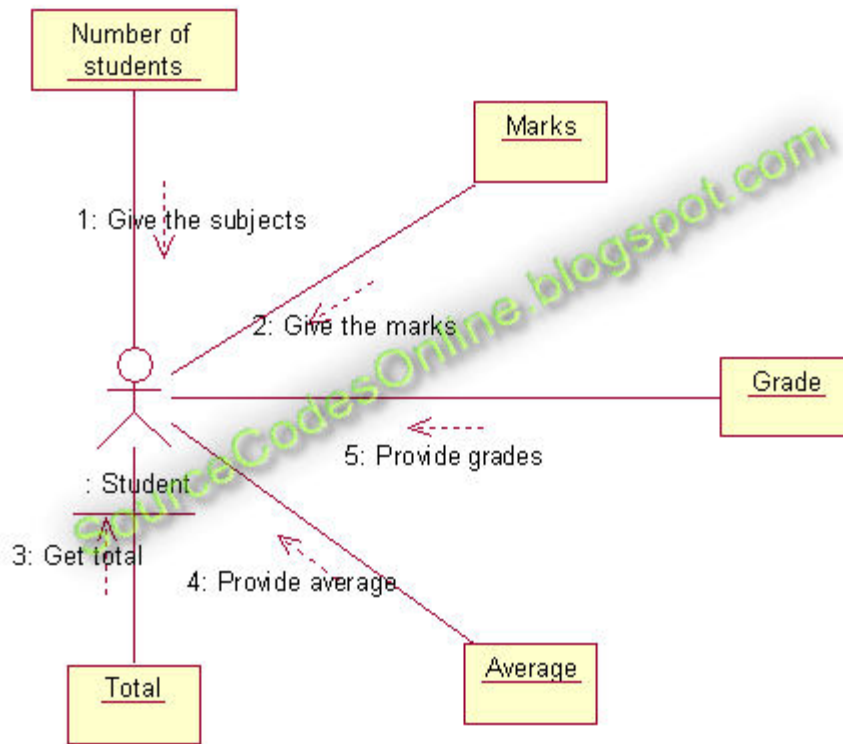**USE CASE DIAGRAM:**
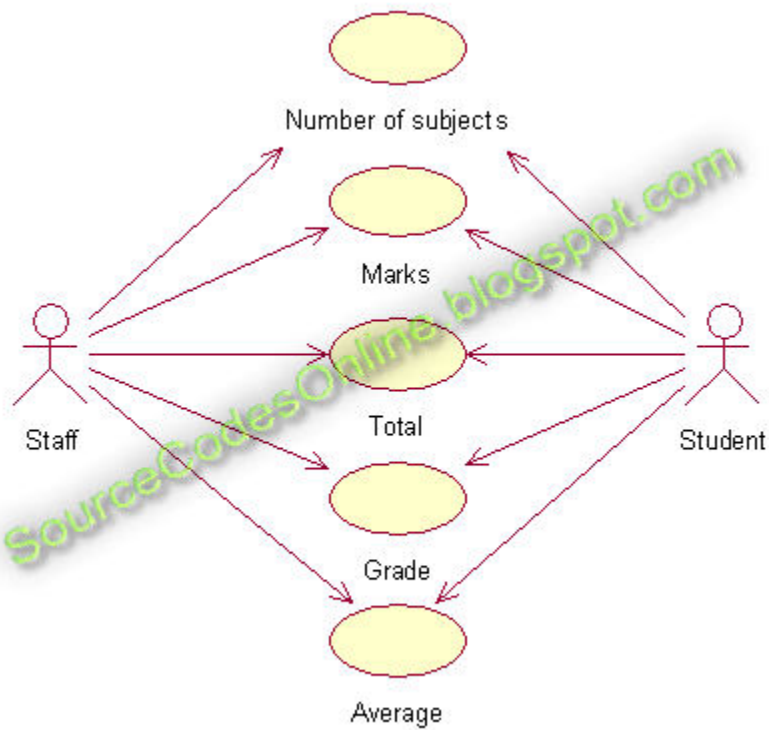
SEQUENCE DIAGRAM FOR STAFF:

COLLABORATION DIAGRAM FOR STAFF:

SEQUENCE DIAGRAM FOR STUDENT:
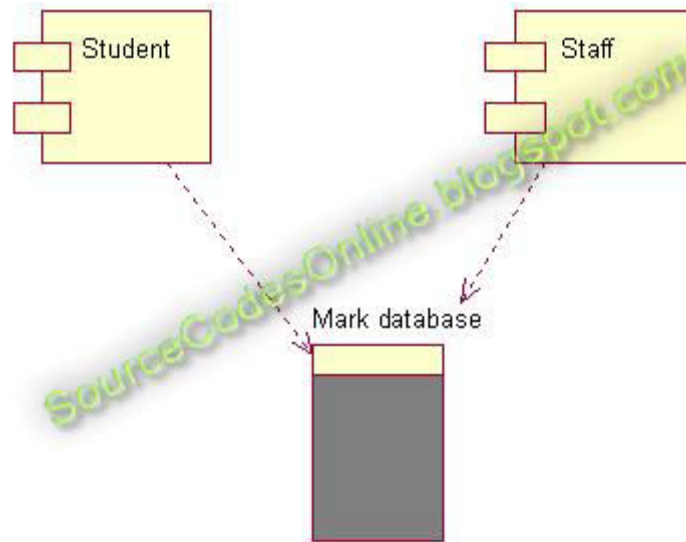
COLLABORATION DIAGRAM FOR STUDENT:

CLASS DIAGRAM:

ACTIVITY DIAGRAM:

Start

Number of students

Analyse Marks

Fail

Pass

Total marks

Calculate average

Give grade

Stop
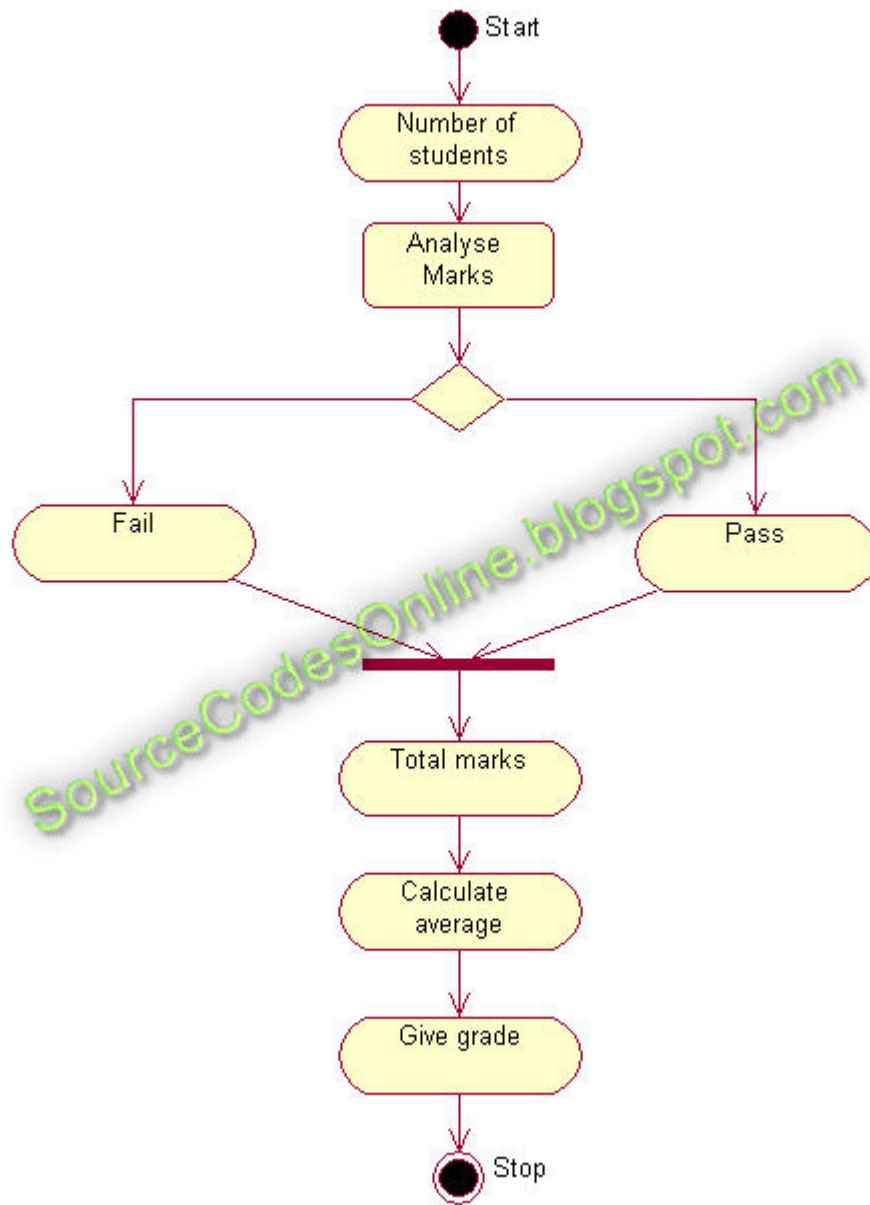
COMPONENT DIAGRAM:

RESULT:

Thus the Student Marks Analysis System application was successfully designed and the output was verified.

7.C) AIM:

To model the **"Online Flight Ticket Reservation System"** using the software **Rational Rose** with various **UML** (Unified Modeling Language) diagrams.

PROBLEM STATEMENTS:

Online ticket reservation application is to maintain flight details, flight status, reservation, cancellation process. The flight status which maintain flight ID, flight name, arrival time, departure time and also it contain details about the seats such business class seats, economic class seats. The flight detail contains the details about needed flight name as well as the details about the seats. Flight reservation which contains the flight ID, ticket number, passenger name, destination, flight name, business and economic class seats, travel charge, passport number, date of travel are reserved. Then the cancellation process is also carried out.

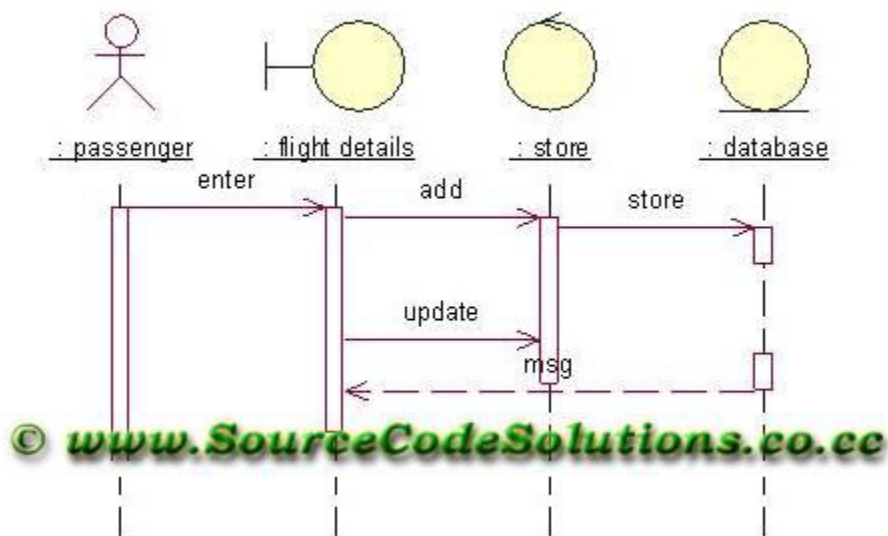Finally a repot is generated about flight details, flight status, and reservation and cancellation tickets.

UML DIAGRAMS:

USE CASE DIAGRAM
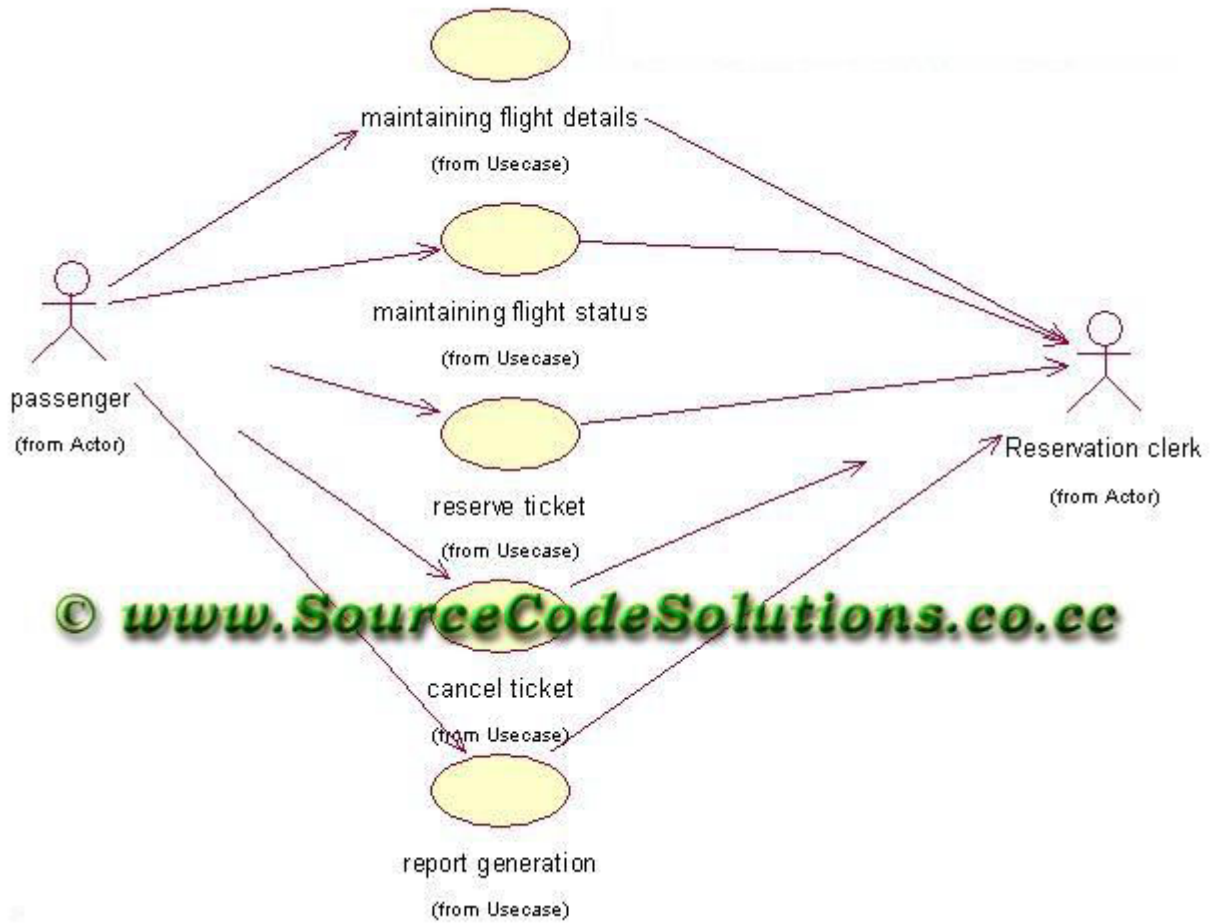
SEQUENCE DIAGRAM:

COLLABORATION DIAGRAM:

ACTIVITY DIAGRAM:
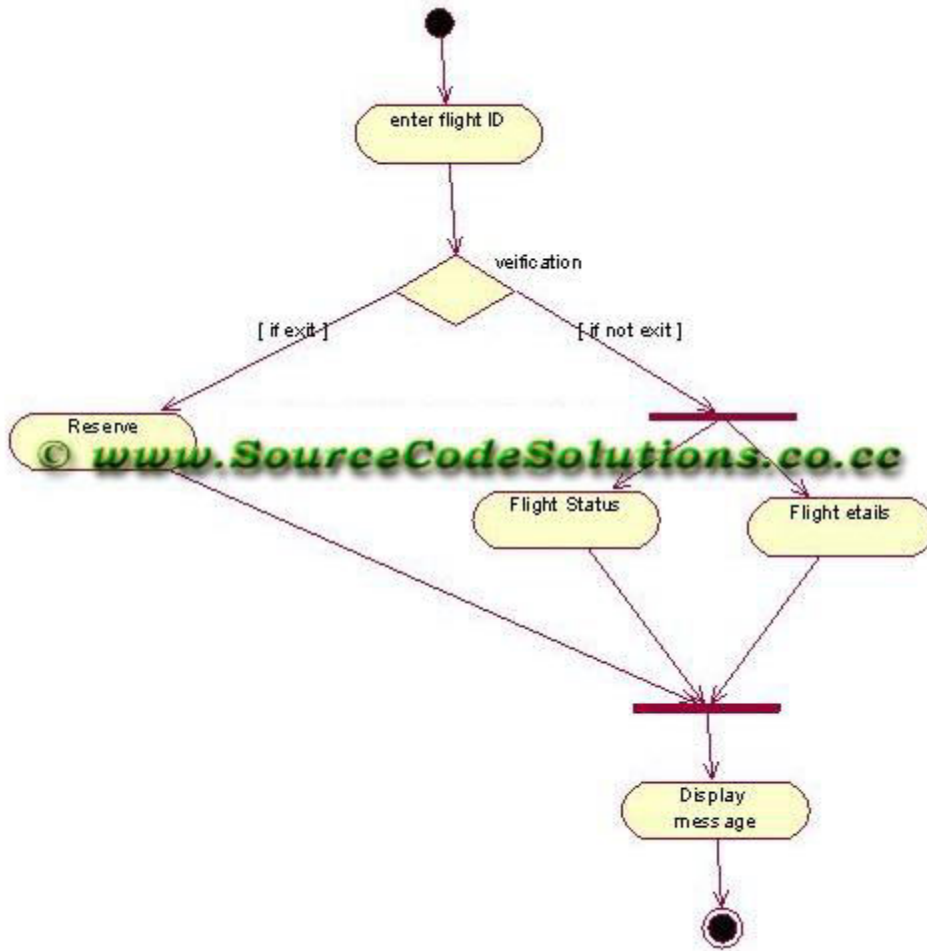
COMPONENT DIAGRAM:
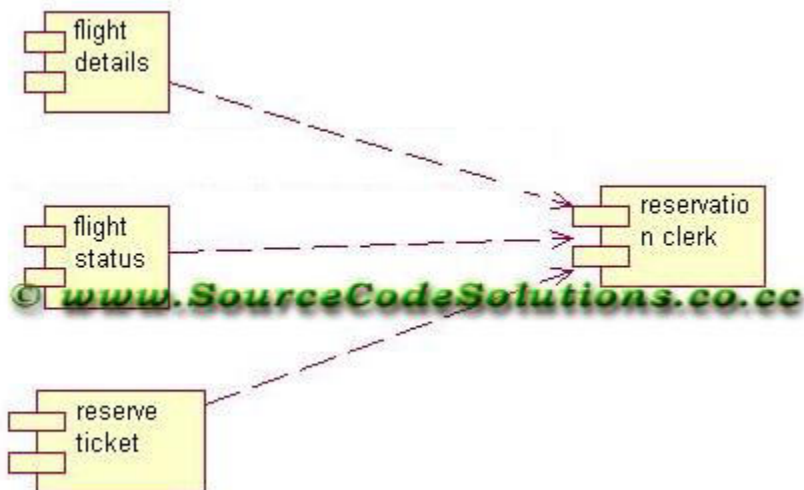
DEPLOYMENT DIAGRAM:
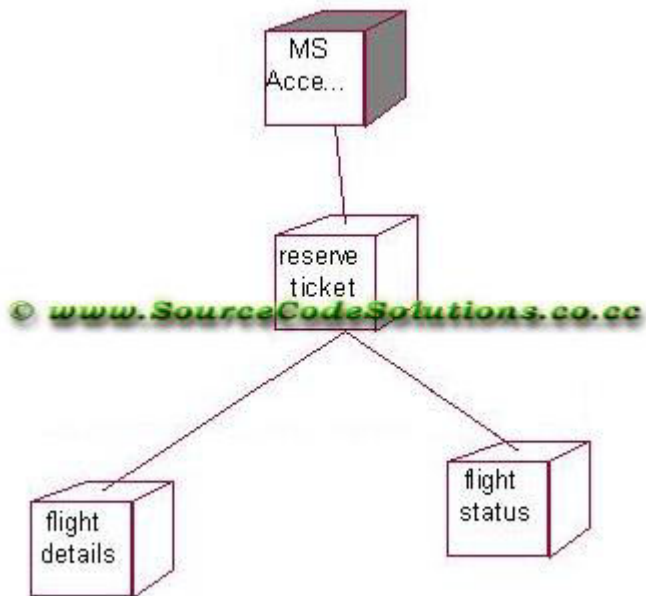
RESULT:

Thus the **Online Flight Ticket Reservation System** application was successfully designed and the output was verified.

7.D) AIM:

To model the "**Stock Maintenance System**" using the software Rational Rose with various UML (Unified Modeling Language) diagrams.
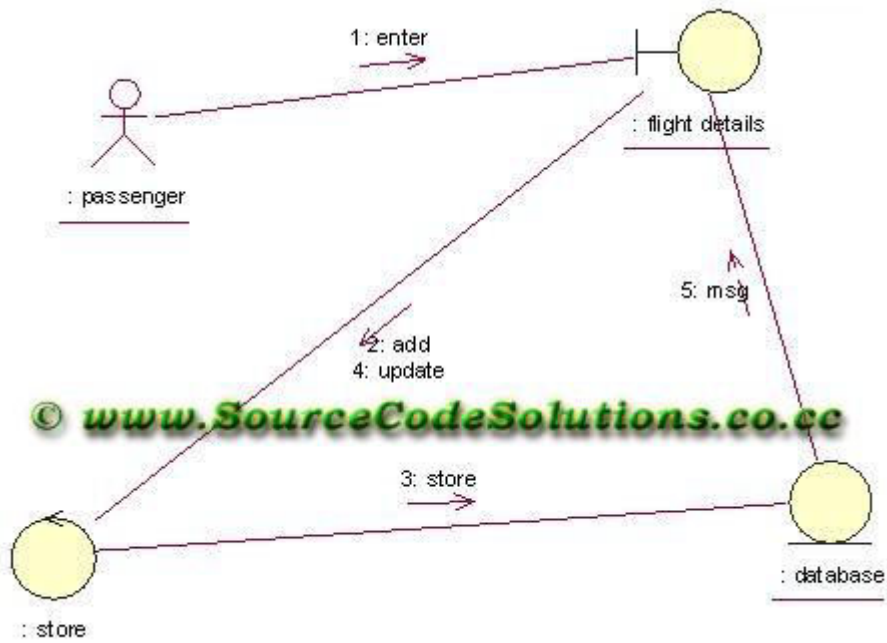
UML DIAGRAMS:

USE CASE DIAGRAM:

SEQUENCE DIAGRAM:

COLLABORATION DIAGRAM:

1: Make order
7: Initiate packing order
6: Advice customer
8: Receive packing order to retrieve stock

: Customer
: Agent
: S

4: Verify product number & stock on hold
3: Check the client in the existing list & create a new record
2: Collect company's customer information
5: Update building
13: Place new s
9: Query location retrieve product add it to

: Agent db

10: Present order for shippig
12: Place r

: Customer db
: Stock db

11: Prepare customer price

14: Cut charge against purchase order for new entr

: Shipping Agent
: Accountant db
: Pu

CLASS DIAGRAM:

ACTIVITY DIAGRAM:

Start

Make order

Check for old client

Verify product no.,description and stock on hold

Create new record

Inform stock person for packing

Query location

Pay for service

Inform

Order to shipping delivery

Inform accountant to prepare customer invoice

COMPONENT DIAGRAM:

RESULT:

Thus the "**Stock Maintenance System**" application was successfully designed and the output was verified.

8) .Design the Testcases for e-commerce application(Flipcart,Amazon).

Test Case ID	Module Name	Test Scenario	Test Case
TC001	Home Page	Verify the details on Home page	Verify that home page is displayed after login or not.
TC002	Home Page	Verify the details on Home page	Verify that user name is displayed on home page or not
TC003	Home Page	Verify the details on Home page	Verify that featured products are displayed on home page

TC004	Home Page	Verify the details on Home page	Verify that Search Functionality is present on home page.
TC005	Home Page	Verify the details on Home page	Verify the home page of application on different browsers.
TC006	Home Page	Verify the details on Home page	Verify the alignment on the home page
TC007	Home Page	Verify the details on Home page	Verify that products displayed on home page are clickable.
TC008	Home Page	Verify the details on Home page	Verify that when user clicks on a product, user should be redirected to product specification page.
TC009	Home Page	Verify the details on Home page	Verify that user profile section is present on home page.

TC0010	Home Page	Verify the details on Home page	Verify that products displayed on home page are categorised.

Pre-requisites	Test Data	Step No.	Test Step
Browser should be installed Internet connection should be present	Username-Username Password-password	1	Open browser
		2	Open url
		3	Enter username and password
		4	Click on login button
Browser should be installed Internet connection should be present User should be logged in		1	Open browser
		2	Open url
		3	Enter username and password
		4	Click on login button
Browser should be installed Internet connection should be present User should be logged in		1	Open browser
		2	Open url
		3	Enter username and password
		4	Click on login button
Browser should be installed Internet connection should be present User should be logged in		1	Open browser
		2	Open url
		3	Enter username and password
		4	Click on login button
Browser should be installed Internet connection should be present User should be logged in		1	Open browser
		2	Open url
		3	Enter username and password

		4	Click on login button
Browser should be installed Internet connection should be present User should be logged in		1	Open browser
		2	Open url
		3	Enter username and password
		4	Click on login button
Browser should be installed Internet connection should be present User should be logged in		1	Open browser
		2	Open url
		3	Enter username and password
		4	Click on login button
		5	Click on any product displayed on home page
Browser should be installed Internet connection should be present User should be logged in		1	Open browser
		2	Open url
		3	Enter username and password
		4	Click on login button
		5	Click on any product displayed on home page
Browser should be installed Internet connection should be present User should be logged in		1	Open browser
		2	Open url
		3	Enter username and password
		4	Click on login button
		5	Click on user name displayed on home page.
Browser should be installed Internet connection should be present User should be logged in		1	Open browser
		2	Open url
		3	Enter username and password
		4	Click on login button

Expected Result	Actual Result	Status	Comments	Defect ID

Browser should be opened				
Ecommerce website should be opened				123
user should be able to input username and password				
Home page should be displayed after login				
Browser should be opened				
Ecommerce website should be opened				
user should be able to input username and password				
Home page should be displayed after login and user name should be displayed on home page				
Browser should be opened				
Ecommerce website should be opened				
user should be able to input username and password				
Home page should be displayed after login and user name should be displayed on home page				
Browser should be opened				
Ecommerce website should be opened				
user should be able to input username and password				
User should be logged in and Search functionality is present on home page.				
Browser should be opened				
Ecommerce website should be opened				
user should be able to input username and password				
User should be logged in and home page should be same on different browsers.				
Browser should be opened				
Ecommerce website should be opened				
user should be able to input username and password				
User should be logged in and alignment of products on home page should be proper.				
Browser should be opened				
Ecommerce website should be opened				
user should be able to input username and password				
Home page should be displayed after login and user name should be displayed on home page				
User should be redirected to product specification page.				

Browser should be opened			
Ecommerce website should be opened			
user should be able to input username and password			
Home page should be displayed after login and user name should be displayed on home page			
User should be redirected to product specification page.			
Browser should be opened			
Ecommerce website should be opened			
user should be able to input username and password			
Home page should be displayed after login and user name should be displayed on home page			
User profile should be displayed.			
Browser should be opened			
Ecommerce website should be opened			
user should be able to input username and password			
Used should be logged in and home page should be displayed, products should be categorised.			

9.design the Test cases for a mobile application (phone pay app)
Test Cases:

CASE 1: SELECT PAY THROUGH PHONEPE WALLET

Pre-Condition : Amount should be available in logged in PhonePe user account

Steps:

1. From merchant app select PhonePe

2. Login and select wallet

3. Click on pay

Expected Result:

- If the transaction status response code is PAYMENT_SUCCESS user should be redirected to order success page of your app.
- If the transaction status response code is not PAYMENT_SUCCESS user should be redirected to order failure page of your app.

CASE 2: SELECT PAY THROUGH VPA AND PAY/DECLINE THE COLLECT CALL REQUEST

Pre-Condition : PhonePe UAT app should be installed and logged in.

Steps:

1. From merchant app select PhonePe
2. Login and select VPA option to pay
3. Enter valid VPA details (Enter the VPA details configured before). Click on verify and pay
4. Navigate to PhonePe UAT app and in notification section view the collect call request
5. Click on Pay/Decline
6. Navigate back to merchant app and see the transaction status

Expected Result:

- User should be displayed with transaction failure message if the collect call request is declined and success message if its paid and the transaction status response code is PAYMENT_SUCCESS

CASE 3: VALIDATE PHONEPE NAME

Steps:

1. From the merchant app navigate to Payment screen

Expected Result:

- In merchant payment page name should be displayed as
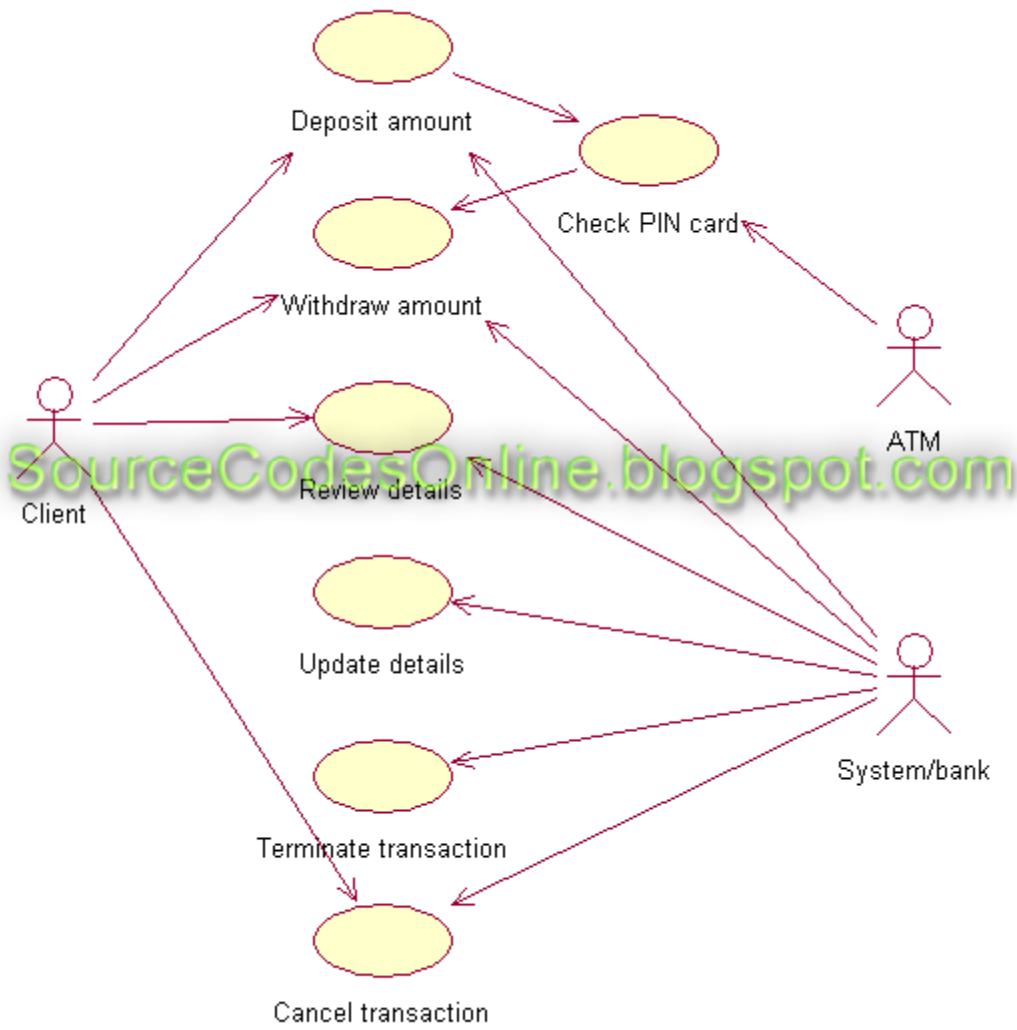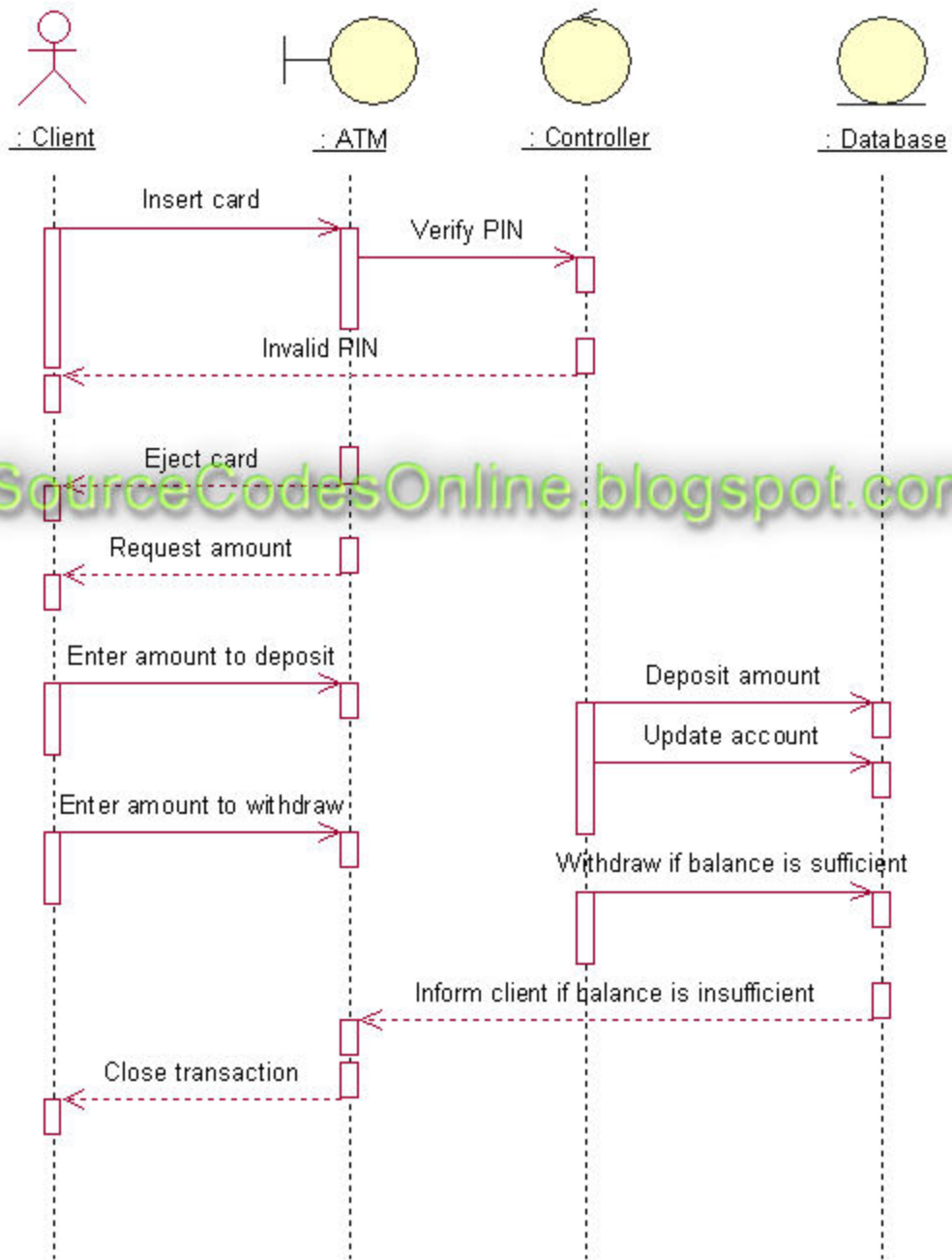
PhonePe / BHIM UPI

or

PhonePe / BHIM UPI

10.AIM:

To model the "**ATM System**" using the software Rational Rose with various UML (Unified Modeling Language) diagrams.
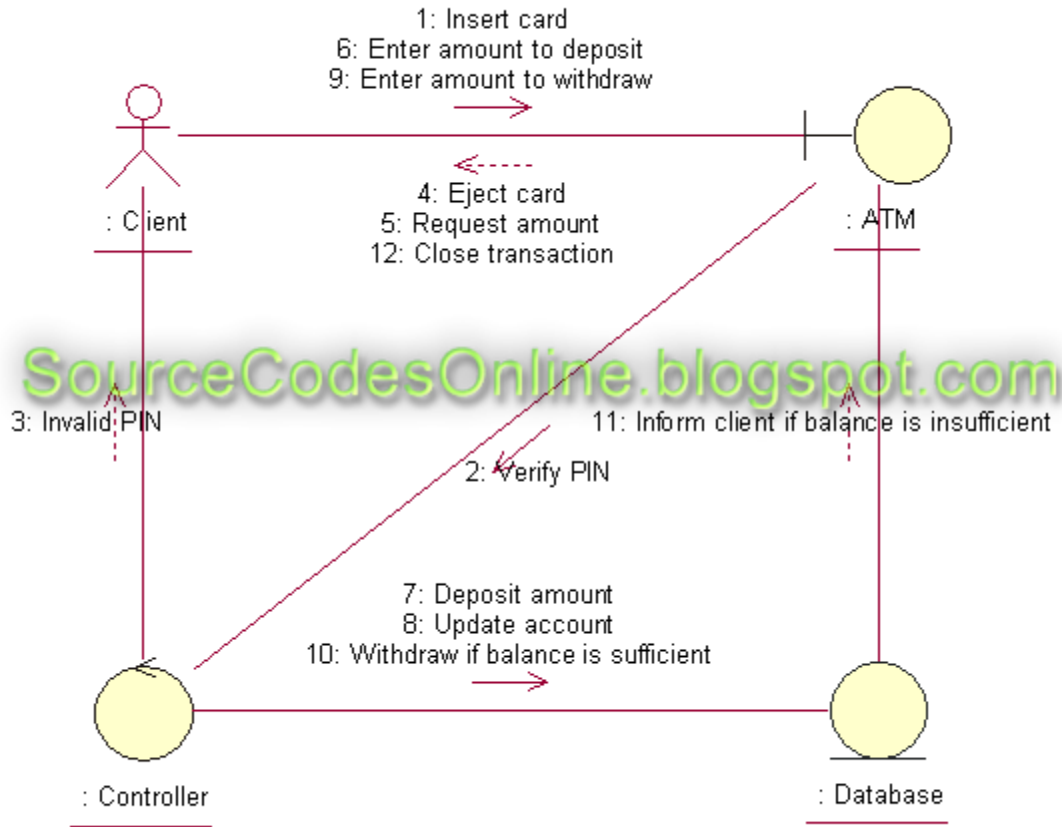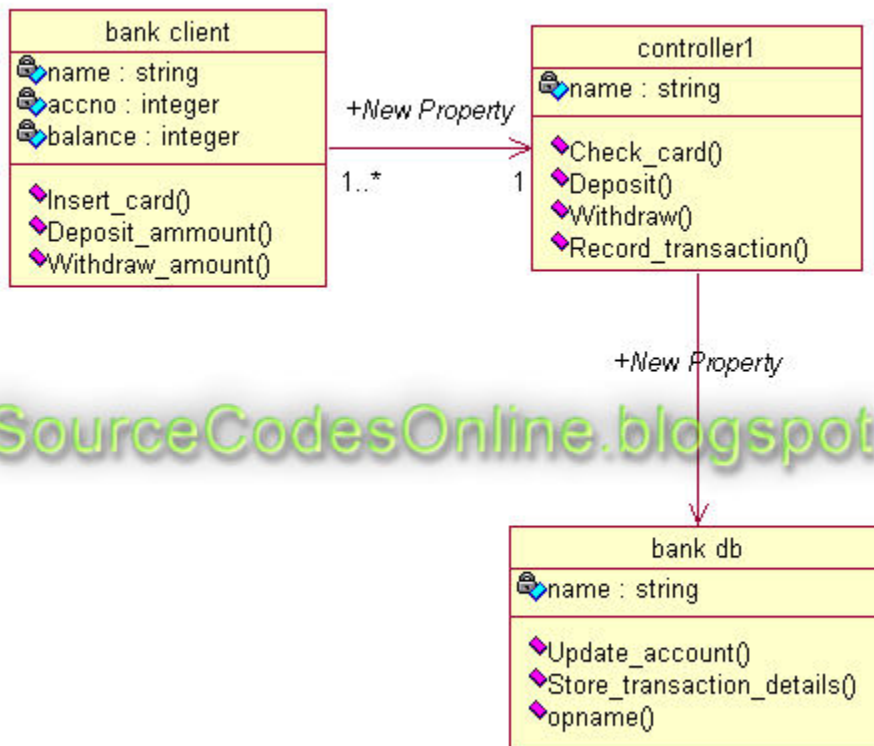
UML DIAGRAMS:

USE CASE DIAGRAM:

Deposit amount

Check PIN card

Withdraw amount

ATM

Client

Review details

Update details

System/bank

Terminate transaction

Cancel transaction

SEQUENCE DIAGRAM:

COLLABORATION DIAGRAM:

1: Insert card
6: Enter amount to deposit
9: Enter amount to withdraw

4: Eject card
5: Request amount
12: Close transaction

: Client

: ATM

3: Invalid PIN

11: Inform client if balance is insufficient

2: Verify PIN

7: Deposit amount
8: Update account
10: Withdraw if balance is sufficient

: Controller

: Database

CLASS DIAGRAM:

bank client

🔒◆name : string
🔒◆accno : integer
🔒◆balance : integer

◆Insert_card()
◆Deposit_ammount()
◆Withdraw_amount()

+*New Property*

1..*   1

controller1

🔒◆name : string

◆Check_card()
◆Deposit()
◆Withdraw()
◆Record_transaction()

+*New Property*

bank db
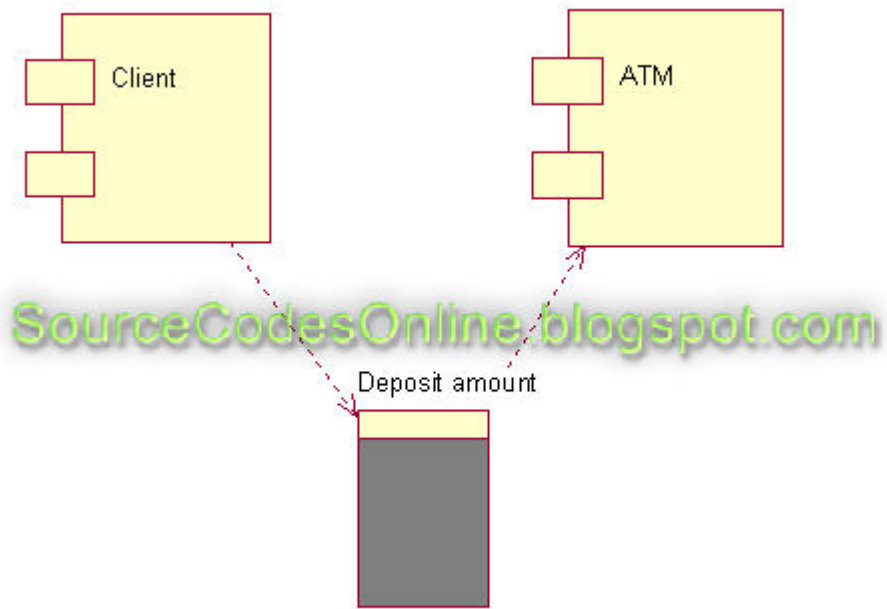
🔒◆name : string

◆Update_account()
◆Store_transaction_details()
◆opname()

ACTIVITY DIAGRAM:

COMPONENT DIAGRAM:

RESULT:

Thus the **ATM System** application was successfully designed and the output was verified.