**Experiment 1:**

**BOOK BANK**

**AIM**

To develop a project of Book bank management system using Rational Rose Software /StarUML

# PROBLEM ANALYSIS AND PROJECT DESIGN

The book bank management system is a software in which a member can register themselves and then he can borrow books from the book bank. It mainly concentrates on providing books for engineering students.

# PROBLEM STATEMENT

The process of members registering and purchasing books from the book bank are described sequentially through following steps:

a. First the member registers himself if he is new to the book bank.

b. Old members will directly select the old member button.

c. They select their corresponding year.

d. After selecting the year they fill the necessary details and select the book and he will be directed towards administrator

e. The administrator will verify the status and issue the book.

# INTRODUCTION

This system would be used by members who are students of any college to check the availability of the books and borrow the books, and then the databases are updated. The purpose of this document is to analyse and elaborate on the high- level needs and features of the book bank management system**.** It also tells us the usability, reliability defined in the use case specification.

# OBJECTIVE

The main objective of the system is to design an online book-bank monitoring system to enable a central monitoring mechanism of the book-bank to be faster and less error prone. Apart from this,

* 1. To help the students acquire the right books for the syllabus at the right time.
  2. To ensure availability of basic textbooks to students against limited funds and to develop students’ ability to handle property loaned to them

# OVERVIEW

The overview of this project is to design a tool for book banks so that it can be used by any book banks to lend their books as well as colleges.

# GLOSSARY

**TERMS DESCRIPTION**

MEMBER The one who registers himself and purchases books from the bank.

DATABASE Database is used to store the details of members and books.

ADMINISTRATOR The one who verifies the availability of book and issue them

USER Member

# SOFTWARE REQUIREMENT SPECIFICATION

This software specification documents the full set of features and functions for the online recruitment system that is performed on the company website.

# PURPOSE

The purpose of the book on bank management systems is to reduce manual intervention.

# SCOPE

The scope of this book bank management system is to act as a tool for book bank administrators for quick reference, and availability of the books.

# FUNCTIONALITY

Many members will be waiting to take the book from the book bank on a single day. To serve all the members

# USABILITY

User interface makes the Recruitment system to be efficient. That is the system will help the member to register easily and help them to get their books easily. The system should be user-friendly.

# PERFORMANCE

It describes the capability of the system to perform the recruitment process of the applicant without any error and perform it efficiently.

# RELIABILITY

The book bank management system should be able to serve the applicant with correct information and day-to-day update of information.

# FUNCTIONAL REQUIREMENTS

Functional requirements are those that refer to the functionality of the system. That is the services that are provided to the member who borrows books.

# EXTERNAL INTERFACE REQUIREMENTS SOFTWARE REQUIREMENTS

1. **Front end:** IBM rational rose enterprise edition.
2. **Back end:** visual basic 8.0.

# HARDWARE REQUIREMENTS

* 1. Processor: Pentium 4
  2. RAM: 256 **MB**
  3. **Operating system:** Microsoft Windows XP.
  4. Free disk space: 1GB UML DIAGRAMS

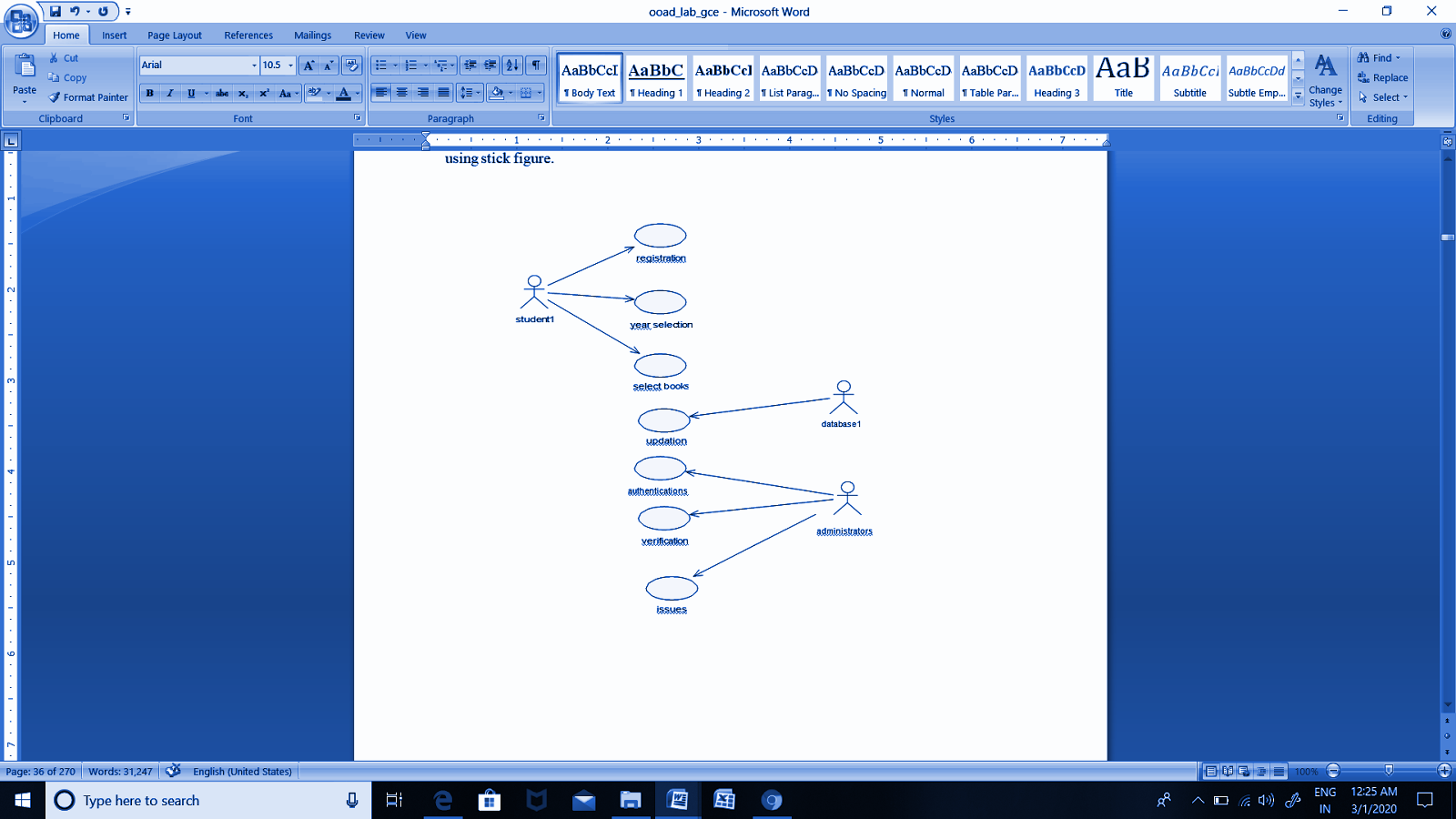
The following UML diagrams describe the process involved in the online recruitment system

1. **Use case diagram**
2. **Class diagram**
3. **Sequence diagram**
4. **Component diagram**

# USE CASE DIAGRAM

A use case is a methodology used in system analysis to identify, clarify, and organize system requirements. The use case is made up of a set of possible sequences of interactions between systems and users in a particular environment and related to a particular goal. It is represented using an ellipse.

The actor is an external entity that makes use of the system being modeled. It is represented using a stick figure.



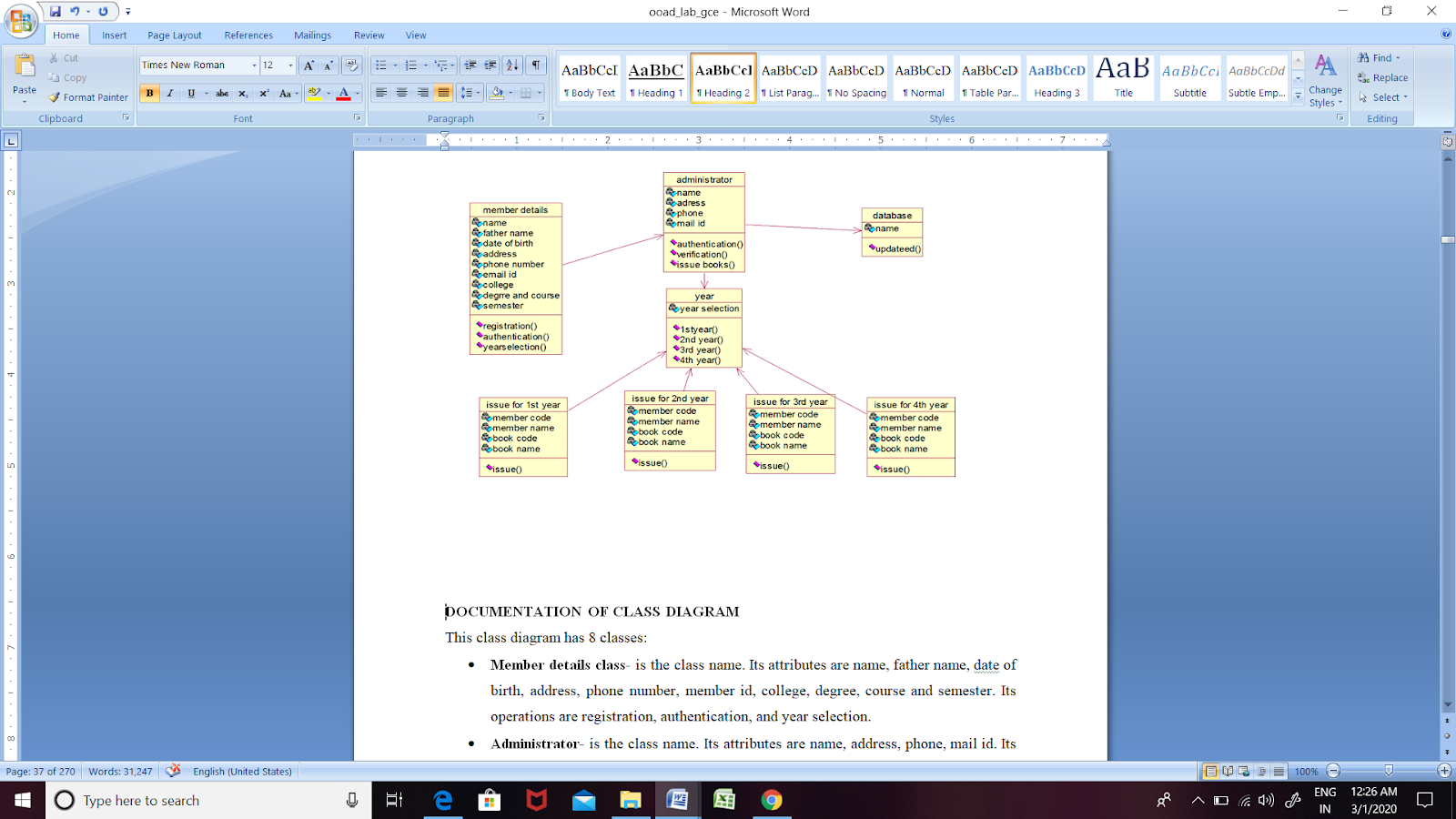
**DOCUMENTATION OF USE CASE DIAGRAM**

The actors in this use case diagram are members and databases. The use cases are the activities performed by actors.

1. The member will register himself in the book bank.
2. After registration, he will select the year to which he belongs
3. After selecting he will select books
4. The database will verify the status of the book and the books will be given.

# CLASS DIAGRAM

A class diagram in the unified modeling language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, and the relationships between the classes. It is represented using a rectangle with three compartments. The top compartment has the class name, the middle compartment has the attributes, and the bottom compartment with operations.

  
  
**DOCUMENTATION OF CLASS DIAGRAM**

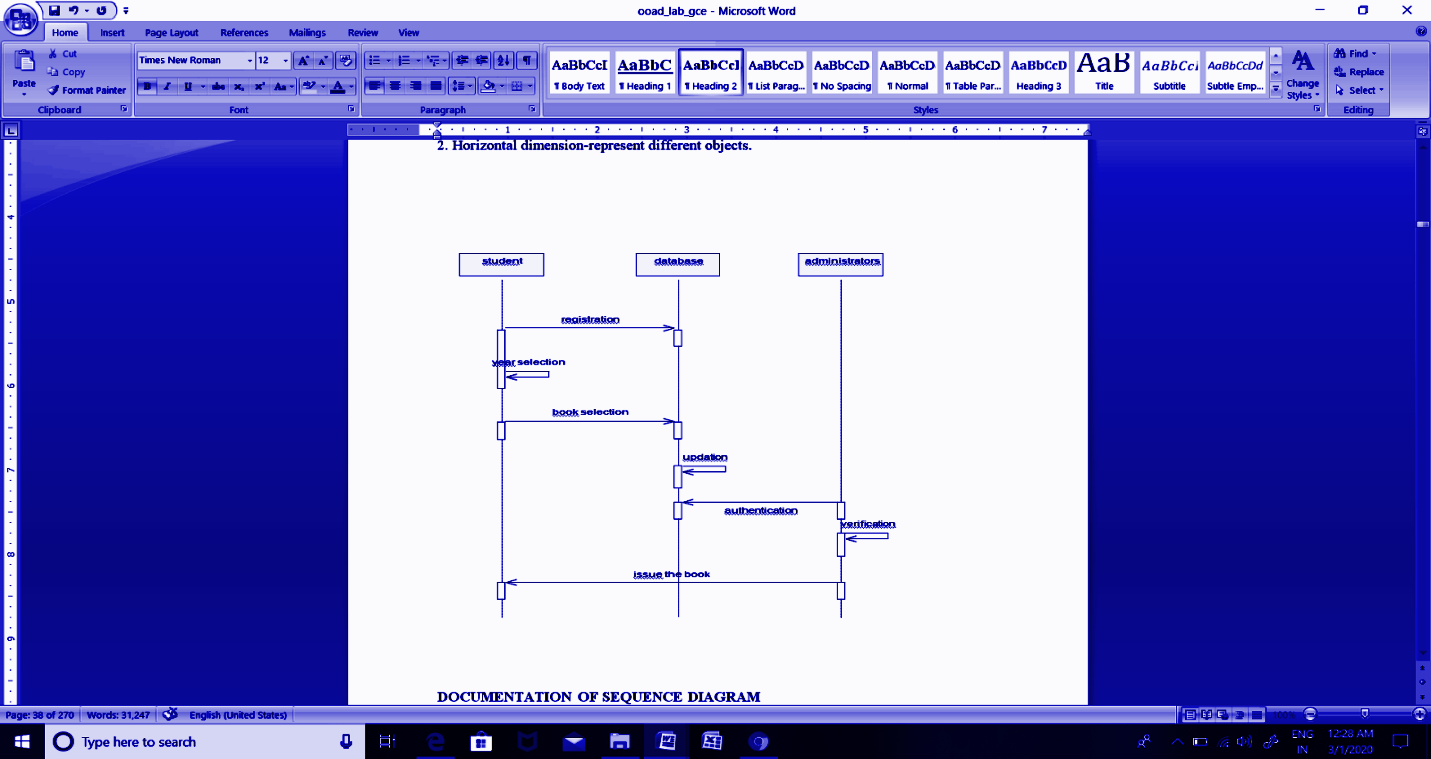
This class diagram has 8 classes:

* **Member details class-** is the class name. Its attributes are name, father
* **Administrator-** is the class name. Its attributes are name, address, phone, and mail id. Its operations are authentication, verification, and issuing books.
* **Year is** the class name. Its attribute is year selection. Its operations are 1st year,2nd year, 3rd year, and 4th year.
* **Issue for 1st year is** the class name. Its attributes are member code, member name, book code, book name, and quantity. Its operation is an issue
* **Issue for the 2nd year is** the class name. Its attributes are member code, member name, book code, book name, and quantity. Its operation is an issue
* **Issue for 3rd year is** the class name. Its attributes are member code, member name, book code, book name, and quantity. Its operation is issue

# SEQUENCE DIAGRAM

A sequence diagram in Unified Modelling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. There are two dimensions.

1. Vertical dimension-represent time.
2. Horizontal dimension-represent different objects.

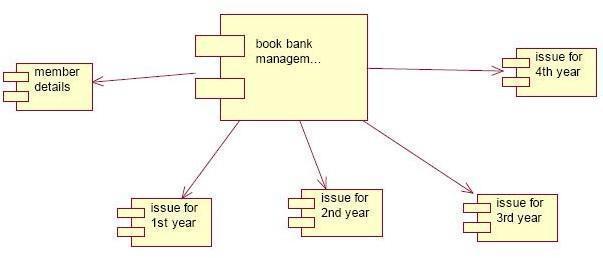
  
**DOCUMENTATION OF SEQUENCE DIAGRAM**

The sequence diagram describes the sequence of steps to show

1. The member registers himself in book bank
2. He will select the year
3. He selects the books given and the database will update the status of the book.
4. Then the administrator will log in and verify the status of the books.
5. If the book is available, he will issue the book.

**COMPONENT DIAGRAM**

The component diagram's main purpose is to show the structural relationships between the components of a system. It is represented by a boxed figure. Dependencies are represented by communication association



**DOCUMENTATION OF COMPONENT DIAGRAM**

The main component in this component diagram is the online book bank management systems. And member details, issue for the first year, the issue for the second-year issue for the third year, and issue for the fourth year are components that come under the main component

# RESULT

Thus, the project to develop book bank management system using Rational Rose/ StarUML Software.

**Experiment 2:**

**ONLINE COURSE RESERVATION SYSTEM**

**AIM**

To design an object-oriented model for the course reservation system.

# PROBLEM ANALYSIS AND PROJECT SCOPE:

The requirement from the customer is obtained and the requirements for the course registration are defined. The requirements are analyzed and defined so that it enables the student to efficiently select a course through the registration system. The project scope is identified and the problem statement is prepared.

# PROBLEM STATEMENT

1. Whenever the student comes to join the course he/she should be provided with the list of courses available in the college.
2. The system should maintain a list of professors who are teaching the course. At the end of the course, the student must be provided with a certificate for the completion of the course.

# SYSTEM REQUIREMENT SPECIFICATION GLOSSARY

Generally a glossary is performed to define the entire domain used in the problem. It defines the storage items that are familiar to the uses it provided these definitions and information about the attribute we are using in the particular project use,

# DEFINITIONS

The glossary contains the working definition for the key concept in the course registration system

# COURSE

The course which is offered by the institution

# COURSE CATALOG

The un bridge for all the courses offered by the institution.

# GRADE

The ranking of a particular student for a particular course offered

# PROFESSOR

A person who reaches the course

# CERTIFICATE

It is the proof of the completion of the course

# REGISTER

One who registers the course for the student

# OBJECTIVES

1. The main purpose of creating the document about the software is to know about the list of the requirements in the software project part of the project to be developed.

# SCOPE

* It specifies the requirement to develop a processing software part that completes the set of requirements.
* In this specification, we define the system requirements that are about the functionality of the system.
* It tells the users about the reliability defined in use case specification

# FUNCTIONALITY

Many members of the process line to check for its occurrences and transaction, we have to carry over at sometimes

# USABILITY

The user interface to make the transaction should be effectively

# PERFORMANCE

It is the capability about which it can perform function for many users at sometimes efficiently (ie) without any ever occurrences

# RELIABILITY

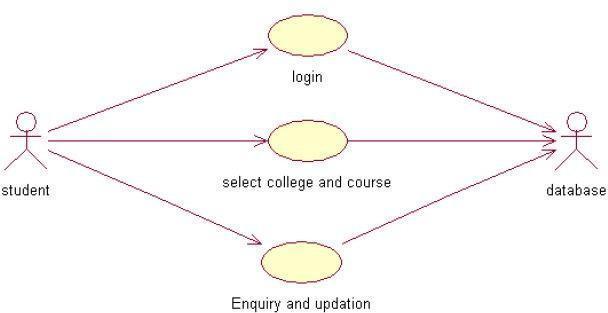
The system should be able to the user through the day to day transaction

**USE CASE DIAGRAM**

Use case is a sequence of transaction in a system whose task is to yield result of measurable value to individual author of the system

Use case is a set of scenarios together by a common user goal

A scenario is a sequence of step describing as interaction between a user and a system



# DOCUMENTATION FOR USE CASE DIAGRAM

The use case diagram in the course registration system illustrates the sequence of steps followed in the system related to the actions of the system

# LOGIN

This use case gives a entry to the student,professor and the register

# SELECT COLLEGE AND COURSE

This use case list out the various courses offered by the institution

# SUBMIT GRADES

This use case given the marks scored by the system

# MAINTAIN PROFESSOR INFORMATION

This use case maintain the information about professor in the system

# MAINTAIN STUDENT INFORMATION

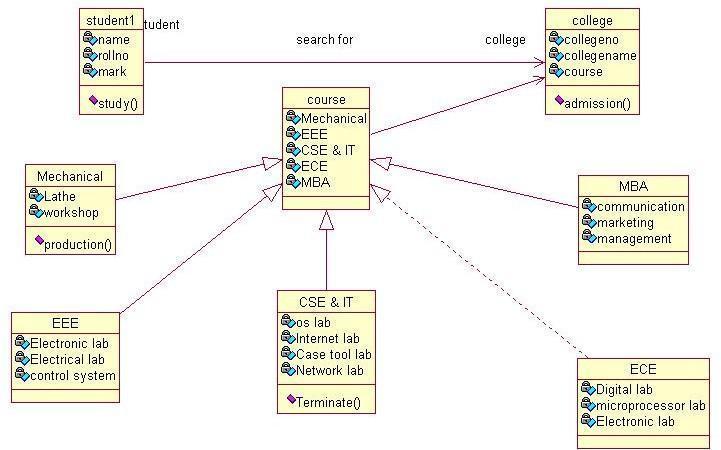
This use case maintain the information about the professor in the system

# CLOSE REGISTRATION

This use case describes the certification of the student when he/she finishes the course

**CLASS DIAGRAM:**

A class diagram describes the type of objectors in the system and the various kinds of static relationships that exist among them.

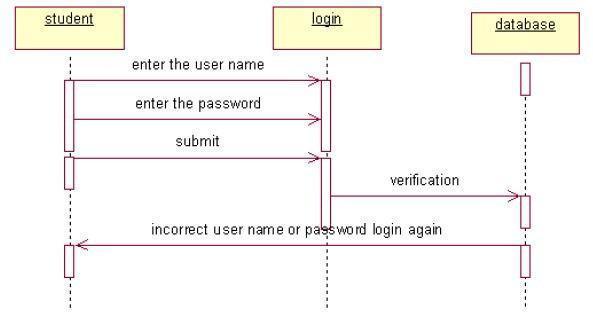
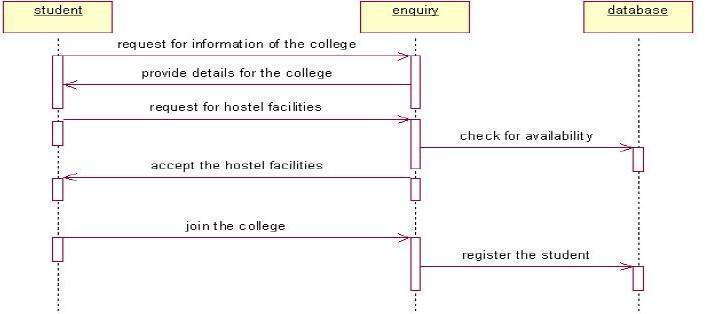


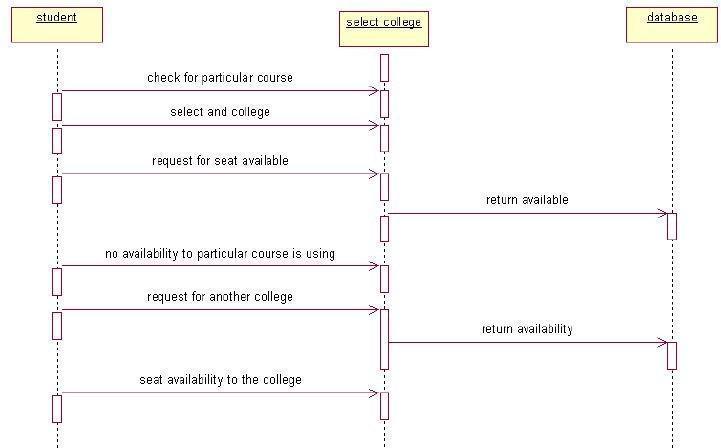
# DOCUMENTATION OF CLASS DIAGRAM

1. The various classes involved in the system are registered student record, professor record all administration grade, and close registration
2. The student register for the course
3. After the course gets over each student will be asked to write a test
4. Test marks are analyzed for the issue grade sheet after certification the registration of the student closes.

# SEQUENCE DIAGRAM

A sequence diagram is one that includes the object of the projects and tells the lifetimes and also various actions performed between objects.



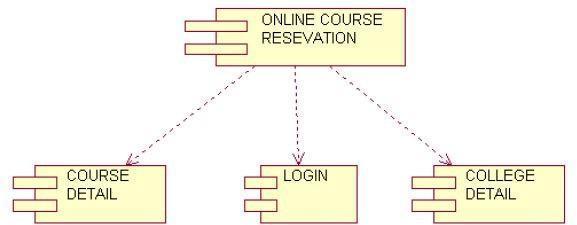


# DOCUMENTATION OF SEQUENCE DIAGRAM

1. The single use case in the course registration is taken and the sequence of operations is followed in the use case
2. In the registration for the course usecase diagram illustration of the process of registering and selecting a course
3. The student enters the institution and gets a catalog about the list of courses offered by the system
4. The student can select a particular use case and register for the course
5. In the record use case submit a grade at the end of each course each student will be asked to write a test. The result will be evaluated for the issue of the grade sheet and the grade are submitted

# COMPONENT DIAGRAM

The component diagram is represented by figure dependency and it is a graph of the design of figure dependency. The component diagram's main purpose is to show the structural relationships between the components of a system. It is represented by a boxed figure. Dependencies are represented by communication association

  
  
  
  
**DOCUMENTATION OF COMPONENT DIAGRAM**

1. The components of the online course reservation are course details, log-in, and college details
2. The course details, log in and college details are dependent on the online course reservation and are shown by the dotted arrows

# RESULT

Thus the UML diagrams to develop an online course reservation system were developed using StarUML.

**Experiment 3:**

**E-TICKETING**

# AIM

To develop the E-Ticketing System using Rational Rose Software and to implement the software in visual basics.

# PROBLEM ANALYSIS AND PROJECT PLANNING

In the E-Ticketing system the main process is a applicant have to login the database then the database verifies that particular username and password then the user must fill the details about their personal details then selecting the flight and the database books the ticket then send it to the applicant then searching the flight or else cancelling the process.

# PROBLEM STATEMENT

The E-Ticketing system is the initial requirement to develop the project about the mechanism of the E-ticketing system and what the process does at all.

1. The requirements are analyzed and refined which enables the end users to efficiently use the E-ticketing system.
2. The complete project is developed after the whole project analysis explaining about scope and project statement is prepared.
3. The main scope for this project is that the applicant should reserve a flight ticket.
4. First the applicant wants to login to the database after that the person wants to fill their details.
5. Then the database will search for a ticket or else the person will cancel the ticket if he/she has no need.

**INTRODUCTION**

# 1.1Purpose

The applicant should login to the database for reserving the ticket. In the specification we define the system requirements that are part of the functionality of the system. It tells us the usability, reliability defined in the use case specification.

# Technology To Be Used

# Microsoft Visual Basic 6.0

* Rational Rose tool or StarUML (for developing UML Patterns)

# Overview

SRS includes two sections: overall description and specific requirements - Overall description will describe the major role of the system components and inter- connections. Specific requirements will describe roles & functions of the actors.

# OVERALL DESCRIPTION

**Functionality**

The database should act as a main role of the e-ticketing system; it can be booking the ticket in an easy way.

# Usability

The User interface makes the Credit Card Processing System to be efficient.

# Performance

It is of the capacities about which it can perform functions for many users at the same time efficiently that are without any error occurrence.

# Reliability

The system should be able to process the user for their corresponding request.

# UML DIAGRAMS

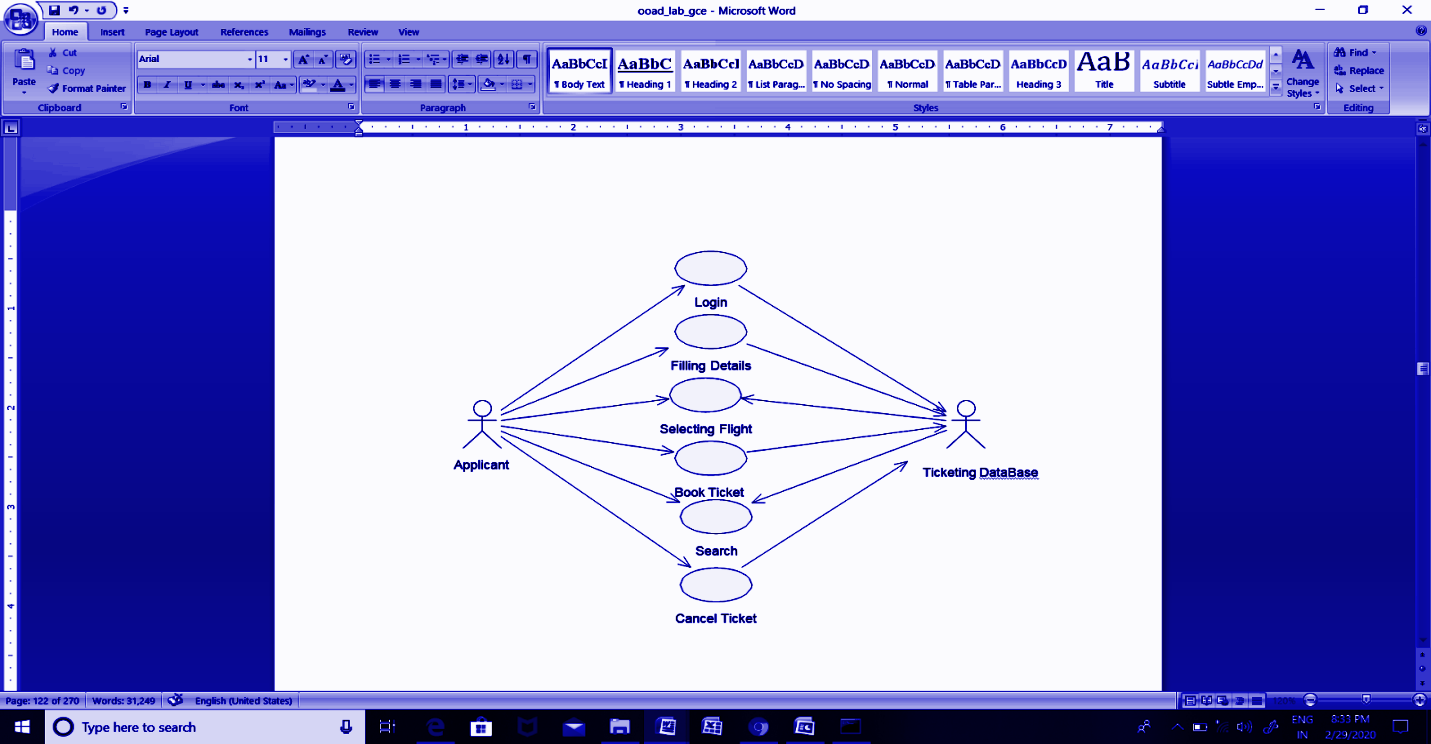
The project can be explained diagrammatically using the following diagrams.

1. Use case diagram
2. Class diagram
3. Sequence diagram
4. Component diagram

# USE CASE DIAGRAM

A use case is a methodology used in system analysis to identify, clarify, and organize system requirements. The use case is made up of a set of possible sequences of interactions between systems and users in a particular environment and related to a particular goal. It is represented using an ellipse.

Actor is any external entity that makes use of the system being modelled. It is represented using stick figure



**DOCUMENTATION OF USE CASE DIAGRAM**

The actors in this use case diagram are applicant, and E-ticketing DataBase. The use cases are the activities performed by actors.

The actors in this use case diagram are

1. **Applicant** - logins the E-Ticketing and filling the required data fields.
2. **E-Ticketing DataBase-**verify the login and filling the details and selected applicant details are stored in it.

The use cases in this use case diagram are

**Login** - applicant enters their username and password to enter into the E-Ticketing form.

**Filling Details –**applicants are used to enter the details in the requiredForm.

**Selecting Flight –**it is used to select the flight for the applicants.

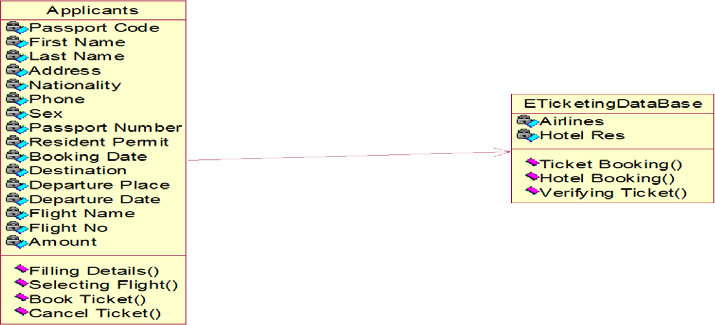
**Book Ticket –** it is used to book the ticket through the E-Ticketing database.

**Search –**it is used to search the flight details.

**Cancel Ticket-** it is used to cancel the ticket through the E-Ticketing DataBase.

# CLASS DIAGRAM

A class diagram in the unified modeling language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, and the relationships between the classes. It is represented using a rectangle with three compartments. Top compartment has the classname, the middle compartment the attributes and the bottom compartment with operations.



# DOCUMENTATION OF CLASS DIAGRAM

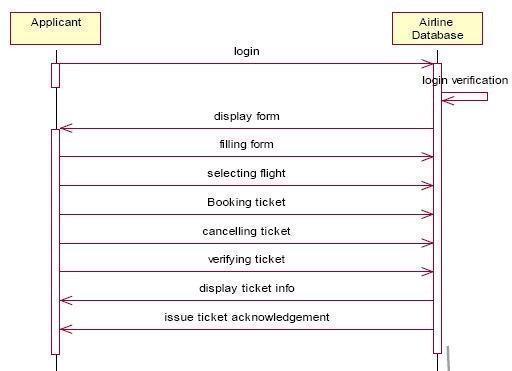
This class diagram has two classes applicant, E-Ticketing DataBase.

1. **Applicant** - logins the E-Ticketing and filling the required data fields
2. **E-Ticketing DataBase-**verify the login and filling the details and selected applicant details are stored in it.

# SEQUENCE DIAGRAM

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. There are two dimensions.

1. Vertical dimension-represent time.
2. Horizontal dimension-represent different objects.



# DOCUMENTATION OF SEQUENCE DIAGRAM

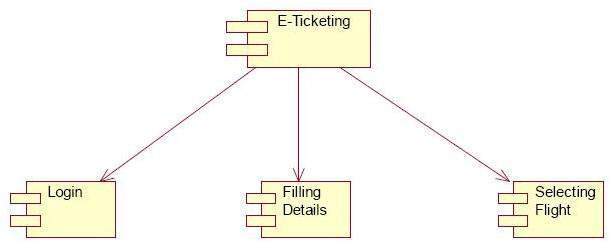
This sequence diagram describes the sequence of steps to show

1. Applicants are used to login the form.And then it verifies the username and password.
2. If the password and username are correct then applicants are used to login the filling details.
3. Applicants are used to selecting the flights and booking the tickets.
4. Now the E-Ticketing DataBase verifies the filling Details.
5. And then the E-Ticketing DataBase displays the ticket information.
6. Incase of any sudden change of the plan,the applicant can cancel the ticket.

# COMPONENT DIAGRAM

The component diagram's main purpose is to show the structural relationships between the components of a system. It is represented by a boxed figure. Dependencies are represented by a communication association.

# DOCUMENTATION OF COMPONENT DIAGRAM



The main component in this component diagram is E-Ticketing systems. And Login, Filling Details and selecting flights applicants are the components that come under the main component.

# RESULT

Thus the UML diagrams to develop an E-Ticketing system using STARUML has been done successfully.

**Experiment 4:**

**RECRUITMENT SYSTEM**

**AIM**

To develop a project on an online recruitment system using Rational Rose Software and to implement the project in Visual Basic.

# PROBLEM ANALYSIS AND PROJECT PLANNING

The Online Recruitment System is an online website in which applicants can register themselves and then attend the exam. Examination will be conducted at some venue. The details of the examination, venue & Date of the examination will be made available to them through the website. Based on the outcome of the exam the applicant will be short listed and the best applicant is selected for the job.

# PROBLEM STATEMENT

The process of applicants is to login to the recruitment system and register for the job online. The resume is processed by the company and the required applicant is called for the test. On the basis of the test marks, they are called for the next level of interview. Finally the best applicant is selected for the job. This process of online recruitment system are described sequentially through following steps,

* The applicant login to the online recruitment system.
* They register with the company for the job.
* They appear for examinations.
* Based on the outcome of the exam, the best applicant is selected.
* The recruiter informs the applicant about their selection.

**INTRODUCTION**

This software specification documents the full set of features and functions for the online recruitment system that is performed on the company website. In this we give specifications about the system requirements that are apart from the functionality of the system to perform the recruitment of the jobseekers. It tells the usability, reliability defined in use case specification.

# OBJECTIVE

The main objective of the Online Recruitment System is to make applicants register themselves online and apply for jobs and attend the exam. Online Recruitment System provides online help to the users all over the world.

# OVERVIEW

The overview of the project is to design an online tool for the recruitment process which eases the work for the applicant as well as the companies. Companies can create their company forms according to their wish in which the applicant can register.

**GLOSSARY TERMS DESCRIPTION**

APPLICANT Applicants can register themselves. After registration, he will be directed to his homepage. Here he can update his profile, change password and see the examination details and all.

RECRUITER Recruiter verify applicant details and conduct examination, approve or disapprove applicant attending examination and providing results about the selected applicant.

DATABASE Database is used to verify login and store the details of selected applicants.

READER Anyone visiting the site to read about the online recruitment system.

USER Applicant and the reader

**SOFTWARE REQUIREMENT SPECIFICATION**

This software specification documents the full set of features and functions for the online recruitment system that is performed on the company website.

The main functionality of the recruitment system is to recruit the applicant for the job in their company.

# USABILITY

User interface makes the Recruitment system to be efficient. That is the system will help the applicant to register easily and help the companies to recruit the applicant effectively. The system should be user friendly.

**PERFORMANCE**

It describes the capability of the system to perform the recruitment process of the applicant without any error and performing it efficiently.

# RELIABILITY

The online recruitment system should be able to serve the applicant with correct information and day-to-day update of information.

**FUNCTIONAL REQUIREMENTS**

Functional requirements are those that refer to the functionality of the system. That is the services that are provided to the applicant who applies for the job.

# UML DIAGRAMS

The following UML diagrams describe the process involved in the online recruitment system

Use case diagram

Class diagram

Sequence diagram

Collaboration diagram

State chart diagram

Activity diagram

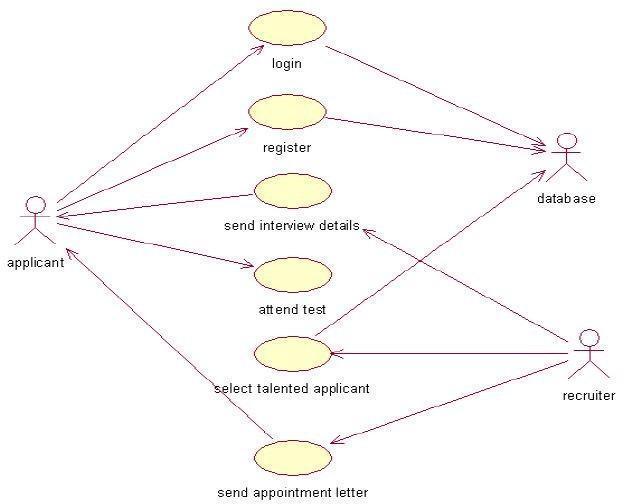
Component diagram

Deployment diagram

Package diagram

# USE CASE DIAGRAM

A use case is a methodology used in system analysis to identify, clarify, and organize system requirements. The use case is made up of a set of possible sequences of interactions between systems and users in a particular environment and related to a particular goal. It is represented using ellipse.Actor is any external entity that makes use of the system being modelled. It is represented using a stick figure.



# DOCUMENTATION OF USE CASE DIAGRAM

The actors in this use case diagram are applicant, recruiter and database. The use cases are the activities performed by actors.

The actors in this use case diagram are

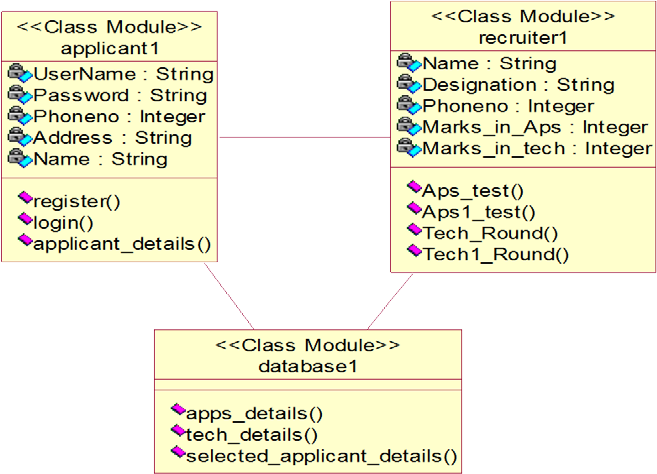
* Applicant - logins the recruitment system and register for the job and attend the test conducted at some venue.
* Recruiter - send the interview details, select talented applicants and send appointment letters to them.
* Databases - verify the login and register details and selected applicant details are stored in it.

The use cases in this use case diagram are

* Login - applicant enter their username and password to enter in to the recruitment system
* Register – applicants register into the recruitment system for jobs.
* Send interview details – recruiter sends interview details to the applicant.
* Attend test – applicant appears for the test.
* Select talented applicants – based on the outcome of the test talented applicants are selected.
* Send an appointment letter – an appointment letter is sent to the selected applicant by the recruiter.

**CLASS DIAGRAM**

A class diagram in the unified modeling language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, and the relationships between the classes. It is represented using a rectangle with three compartments. Top compartment has the classname, the middle compartment the attributes and the bottom compartment with operations.



# DOCUMENTATION OF CLASS DIAGRAM

This class diagram has three classes: applicant, recruiter and database.

* Applicant – is the class name. Its attributes are username, password, name, phone no and address. The operations performed in the applicant class are login, register and giving applicant details.
* Recruiter – is the class name. Its attributes are name, designation, phone no, marks in apps and marks in technical. The operations performed are selecting applicants based on apps and technical.
* Database – is the class name. The operations performed are storing applicant details, verifying login and storing selected applicant details.

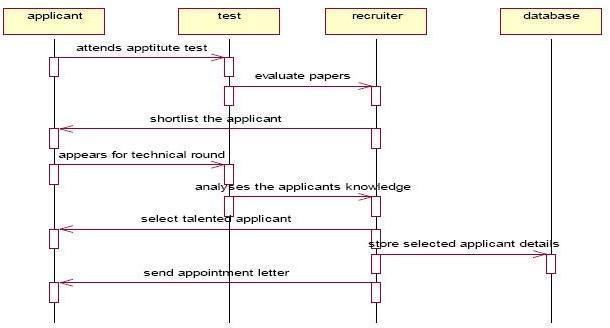
# SEQUENCE DIAGRAM

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. There are two dimensions.

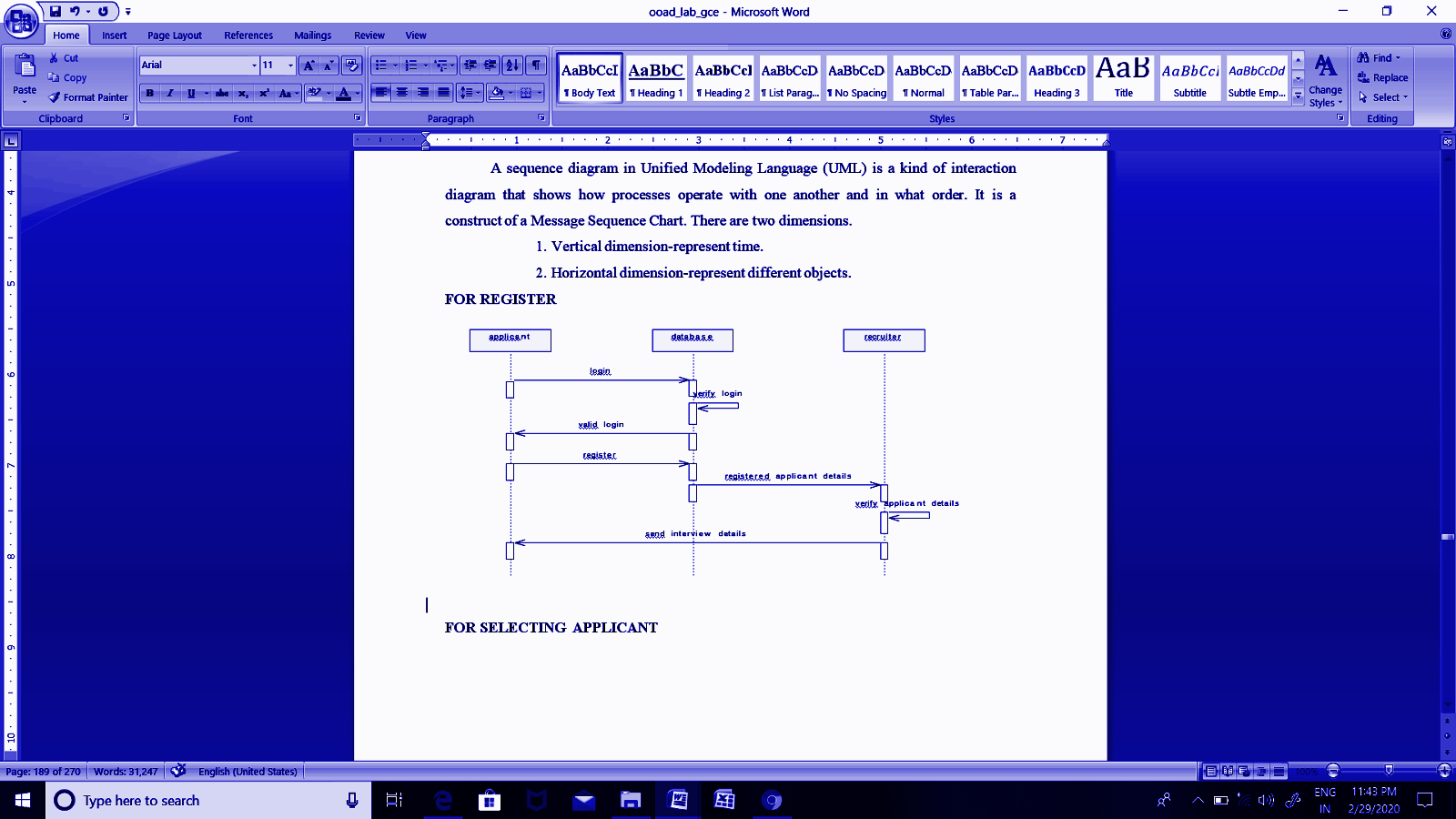
Vertical dimension-represent time.

Horizontal dimension-represent different objects.

# FOR SELECTING APPLICANT



**FOR REGISTER**



**DOCUMENTATION OF SEQUENCE DIAGRAM**

**REGISTER**

This sequence diagram describes the sequence of steps to show

The applicant login into the recruitment system and register for a job.

The verification done in the database and recruiter

The interview details are sent to the applicant by the recruiter.

# SELECTING APPLICANT

This sequence diagram shows steps to show

The applicant attend aptitude test and they are shortlisted based on evaluation

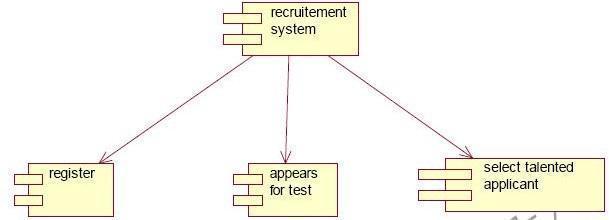
The applicant appear for technical round

The talented applicant is selected.

This detail is stored in the database.

# COMPONENT DIAGRAM

The component diagram's main purpose is to show the structural relationships between the components of a system. It is represented by a boxed figure. Dependencies are represented by a communication association.



# DOCUMENTATION OF COMPONENT DIAGRAM

The main component in this component diagram is online recruitment systems. And registering, attending tests and selecting talented applicants are the components that come under the main component.

# RESULT

Thus the project to develop an online recruitment system using Rational Rose Software is done successfully.

**Experiment 5:**

**HOSPITAL MANAGEMENT SYSTEM**

**AIM**

Hospital Management System is an organized computerized system designed and programmed to deal with day to day operations and management of the hospital activities. The program can look after inpatients, outpatients, records, database treatments, status illness, billings in the pharmacy and labs. It also maintains hospital information such as ward id, doctors in charge and department administering. The major problem for the patient nowadays is to get a report after consultation. Many hospitals manage reports in their system but it's not available to the patient when he / she is outside. In this project we are going to provide the extra facility to store the report in the database and make it available from anywhere in the world.

### INTRODUCTION:

The project Hospital Management system includes registration of patients, storing their details into the system, and also computerized billing in the pharmacy, and labs. The software has the facility to give a unique id for every patient and stores the details of every patient and the staff automatically. It includes a search facility to know the current status of each room. Users can search for the availability of a doctor and the details of a patient using the id.

The Hospital Management System can be entered using a username and password. It is accessible either by an administrator or receptionist. Only they can add data into the database. The data can be retrieved easily. The interface is very user-friendly. The data is well protected for personal use and makes the data processing very fast.

The Hospital Management System is powerful, flexible, and easy to use and is designed and developed to deliver real conceivable benefits to hospitals.

Hospital Management System is designed for multispeciality hospitals, to cover a wide range of hospital administration and management processes. It is an integrated end-to- end Hospital Management System that provides relevant information across the hospital to support effective decision making for patient care, hospital administration and critical financial accounting, in a seamless flow.

Hospital Management System is a software product suite designed to improve the quality and management of hospital management in the areas of clinical process analysis and activity-based costing. Hospital Management System enables you to develop your organization and improve its effectiveness and quality of work. Managing the key processes efficiently is critical to the success of the hospital helps you manage your processes

### PROBLEM STATEMENT:

**Lack of immediate retrievals: -**

The information is very difficult to retrieve and to find particular information like- E.g. - To find out about the patient’s history, the user has to go through various registers. This results in convenience and wastage of time.

#### Lack of immediate information storage: -

The information generated by various transactions takes time and effort to be stored at the right place.

#### Lack of prompt updating: -

Various changes to information like patient details or immunization details of a child are difficult to make as paperwork is involved.

#### Error prone manual calculation: -

Manual calculations are error prone and take a lot of time this may result in incorrect information. For example, calculation of patient’s bills based on various treatments.

#### Preparation of accurate and prompt reports: -

This becomes a difficult task as information is difficult to collect from various registers.

### Goals

1-User friendly 2-Simple fast

3-Low cost and effective

4-It deals with the collection of patient’s information

5- Diagnosis

### Objective:-

1) Define hospital

2) Recording information about the Patients that come.

3) Generating bills.

4) Recording information related to diagnosis given to Patients.

5) Keeping record of the Immunization provided to children/patients.

6) Keeping information about various diseases and medicines available to cure them.

These are the various jobs that need to be done in a Hospital by the operational staff and Doctors. All these works are done on paper.

### Scope of the Project:-

1) Information about Patients is done by just writing the Patient's name, age and gender. Whenever the Patient comes up his information is stored freshly.

2) Bills are generated by recording the price for each facility provided to Patient on a separate sheet and at last they all are summed up.

3) Diagnosis information to patients is generally recorded on the document, which contains Patient information. It is destroyed after some time period to decrease the paper load in the office.

4) Immunization records of children are maintained in pre-formatted sheets, which are kept in a file.

5) Information about various diseases is not kept in any document. Doctors themselves do this job by remembering various medicines.

All this work is done manually by the receptionist and other operational staff and a lot of papers are needed to be handled and taken care of. Doctors have to remember various medicines available for diagnosis and sometimes miss better alternatives as they can’t remember them at that time.

### MODULES:

The entire project mainly consists of 7 modules, which are

1. Admin module
2. User module (patient)
3. Doctor module
4. Nurse module
5. Pharmacist module
6. Laboratorist module
7. Accountant module

Admin module:

1. manage department of hospitals, user, doctor, nurse, pharmacist, laboratorist accounts.

2. watch appointment of doctors

3. watch transaction reports of patient payment

4. Bed ,ward, cabin status

5. watch blood bank report

6. watch medicine status of hospital stock

7. watch operation report

1. watch birth report
2. watch diagnosis report
3. watch death report

**User module(patient):**

1. View appointment list and status with doctors
2. View prescription details
3. View medication from doctor
4. View doctor list
5. View blood bank status
6. View operation history
7. Views admit history. like bed, ward icu etc
8. Manage own profile

**Doctor module:**

1. Manage patients. account opening and updating
2. Create, manage appointment with patient
3. Create prescription for patient
4. Provide medication for patients
5. Issue for operation of patients and creates operation report
6. Manage own profile

**Nurse module:**

1. Manage patients. account opening and updating
2. Allot bed, ward, cabin for patients
3. Provide medication according to patient prescription
4. Manage blood bank and update status
5. Keep record of patient operation, baby born and death of patient
6. Manage own profile

**Pharmacist module:**

1. Maintain medicine
2. Keep records of hospitals stock medicines and status
3. Manage medicine categories
4. Watch prescription of patient
5. Provide medication to prescriptions

### L aboratorist module:

1. Watch prescription list
2. Upload diagnostic report
3. Preview of report files. like xray images, ct scan, mri reports
4. Manage own profile

### Accountant module:

1. Create invoice for payment
2. Order invoice to patient
3. Take cash payment
4. Watch payment history of patients
5. Manage own profile

### SYSTEM DESIGN:

### INTRODUCTION TO UML:

UML Design

The Unified Modeling Language (UML) is a standard language for specifying, visualizing, constructing, and documenting the software system and its components. It is a graphical language , which provides a vocabulary and set of semantics and rules. The UML focuses on the conceptual and physical representation of the system. It captures the decisions and understandings about systems that must be constructed. It is used to understand, design, configure, maintain, and control information about the systems.

The UML is a language for:

1. Visualizing
2. Specifying
3. Constructing
4. Documenting

**Visualizing**

Through UML we see or visualize an existing system and ultimately we visualize how the system is going to be after implementation. Unless we think, we cannot implement. UML helps to visualize how the components of the system communicate and interact with each other.

**Specifying**

Specifying means building models that are precise, unambiguous and complete UML addresses the specification of all the important analysis design, implementation decisions that must be made in developing and deploying a software system.

**Constructing**

UML models can be directly connected to a variety of programming language through mapping a model from UML to a programming language like JAVA or C++ or VB. Forward Engineering and Reverse Engineering is possible through UML.

**Documenting**

The Deliverables of a project apart from coding are some Artifacts, which are critical in controlling, measuring and communicating about a system during its developing requirements, architecture, desire, source code, project plans, tests, prototypes, releases, etc...

**UML Approach**

**UML Diagram**

A diagram is the graphical presentation of a set of elements, most often rendered as a connected graph of vertices and arcs . you draw a diagram to visualize a system from a different perspective, so a diagram is a projection into a system. For all but most trivial systems, a diagram represents an elided view of the elements that make up a system. The same element may appear in all diagrams, only a few diagrams , or in no diagrams at all. In theory, a diagram may contain any combination of things and relationships. In practice, however, a small number of common combinations arise, which are consistent with the five most useful views that comprise the architecture of a software-intensive system. For this reason, the UML includes nine such diagrams:

1. Class diagram

2. Object diagram

3. Use case diagram

4. Sequence diagram

5. Collaboration diagram

6. State chart diagram

7. Activity diagram

8. Component diagram

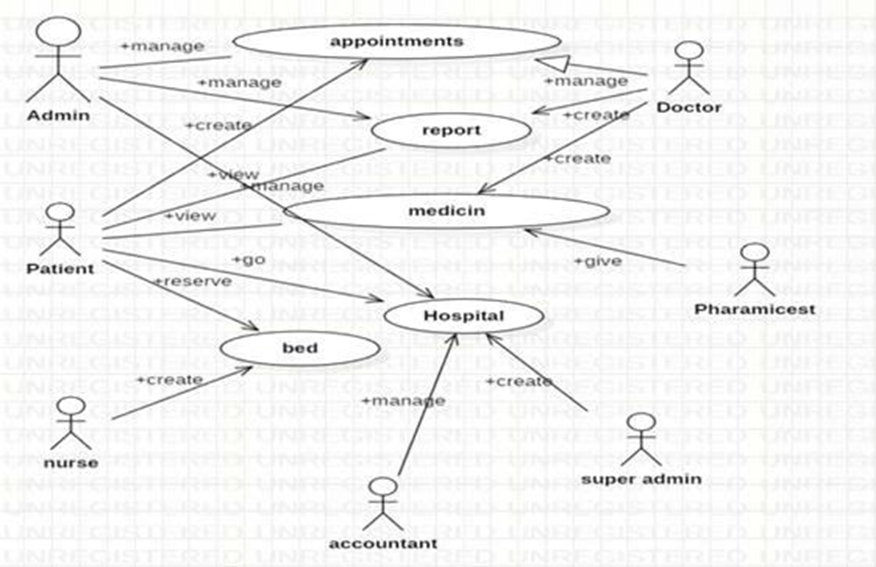
9. Deployment diagram

**USE CASE DIAGRAM:**

A use case diagram in the Unified Modeling Language(UML) is a type of behavioral diagram defined by and created from a use-case analysis.its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals(represented as use cases),and any dependencies between those use cases.

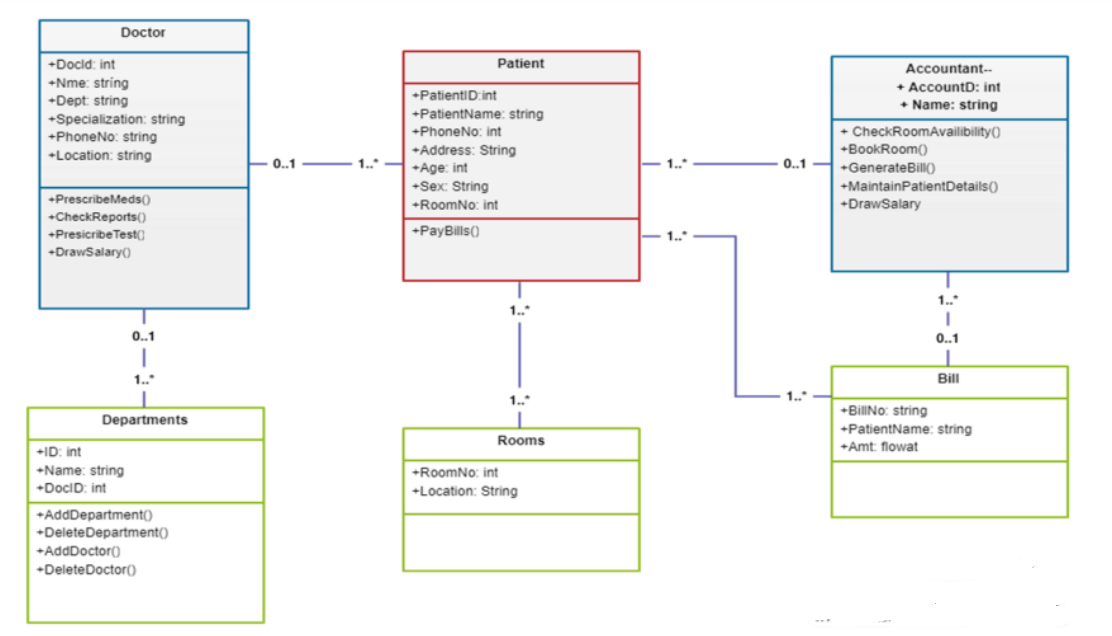
Use case diagrams are formally included in two modeling languages defined by the OMG:unified modeling language(UML) and the systems modeling language(sysML)

Use case diagram of the project:



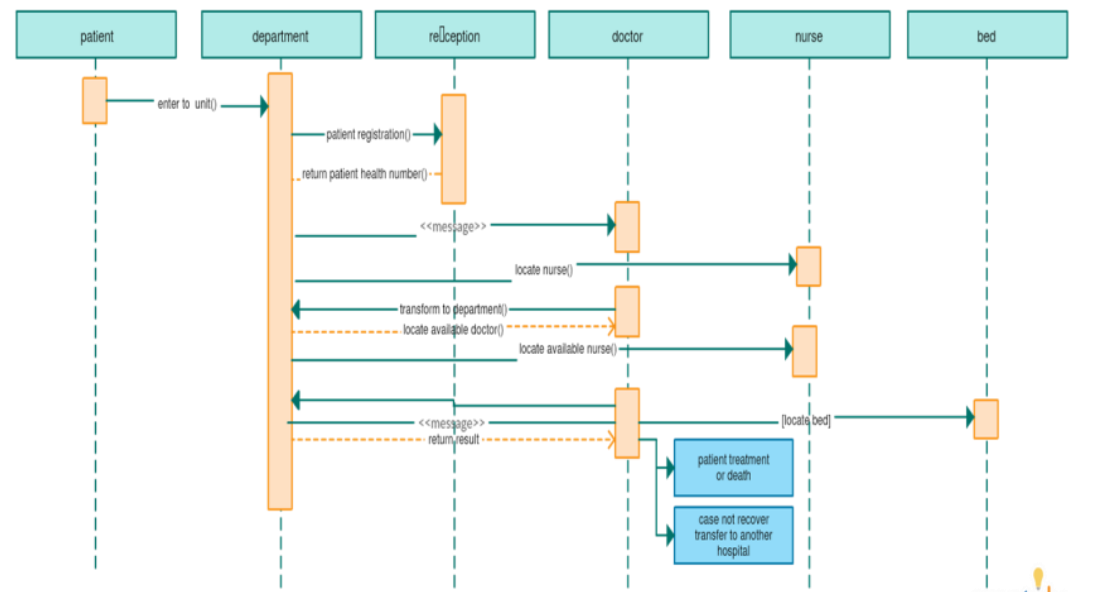
**Class Diagram:**

A Class is a category or group of things that has similar attributes and common behavior. A Rectangle is the icon that represents the class it is divided into three areas. The upper most area contains the name, the middle; area contains the attributes and the lowest areas show the operations. Class diagrams provide the representation that developers work from. Class diagrams help on the analysis side, too.



**Sequence diagram:**

A Sequence Diagram is an interaction diagram that emphasizes the time ordering of messages; a collaboration diagram is an interaction diagram that emphasizes the structural organization of the objects that send and receive messages. Sequence diagrams and collaboration diagrams are isomorphic, meaning that you can take one and transform it into the other.

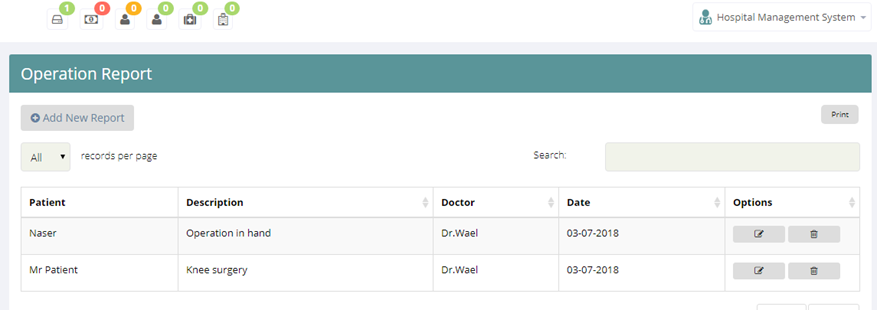


### EXISTING SYSTEM:

Hospitals currently use a manual system for the management and maintenance of critical information. The current system requires numerous paper forms, with data stores spread throughout the hospital management infrastructure. Often information is incomplete or does not follow management standards. Forms are often lost in transit between departments requiring a comprehensive auditing process to ensure that no vital information is lost. Multiple copies of the same information exist in the hospital and may lead to inconsistencies in data in various data stores.

### PROPOSED SYSTEM:

The Hospital Management System is designed for any hospital to replace their existing manual paper based system. The new system is to control the information of patients. Room availability, staff and operating room schedules and patient invoices. These services are to be provided in an efficient, cost effective manner, with the goal of reducing the time and resources currently required for such tasks .



# RESULT

Thus the UML diagrams to develop a Hospital Management system using STARUML has been done successfully.

**Experiment 6:**

**ONLINE BANKING SYSTEM**

**AIM**

The main purpose that banks have been serving since their inception is keeping our money safe for us. While keeping our money safe, they also let us earn a certain amount of interest on the money deposited with them. Traditional banks have been doing this, and internet banks continue the same function. The only difference is in the way the transactions are made.

We all know about internet banking but few of us actually understand about the history of internet banking and how it all came out. Knowing the history of internet banking can be incredibly useful, especially since it will allow us to have more respect for the little things that we take for granted. Internet banking has been around for quite a few years now, but has really only become prominent over the past year or so in particular. Internet banking offers an array of different advantages to the user, including account balances and history including year-to date information, the ability to transfer money from one account to another and to payees for bill payments, check history, reorders, and stop payments, check credit card balances and statements, complete online loan applications, secure interactive messaging with staff and much more.

Internet banking basically allows you to be able to do everything that you can do it all right from the convenience of your own home.

**Objectives**

· Create a banking system that is easily accessible by customers from the comfort of their homes, offices etc.

· Reduce the flow of human traffic and long queues at banks.

· Reduce the time wasted in going to banks to stay in queues.

· Promote efficient and effective banking for the banks by focusing on those services that still require physical presence at the banking hall.

**Intended Audience & Reading Suggestions.**

The different types of readers are:-

(a). Customers.

* Project Scope.
* Security Available.

(b). Employers.

(c). Developers.

* Project Scope.
* Use Case Module.

(d). Project Manager.

* System Features.
* Hardware Requirement.
* Software Requirement.
* Interface Requirement.

**Scope.**

The Scope of this project is limited to the activities of the operation units of the banking system which include opening of account, deposit of funds, withdrawal of funds & transfer.

* à Any bank can use this application to provide better service to their customers.
* à Customers can access his/her all accounts present in various branches of the same bank at one click.
* à Bank can publish various upcoming plans for customers through this application.
* à Manager can access all accounts present in the bank through this application.
* à Reduction in workload of all employees will be possible through this application as transaction rights are provided online to customers.
* àIt can be extended for global communication between all banks in the world.

**OVERALL DESCRIPTION**.

**Product Perspective.**

The client will have a client interface in which he can interact with the banking system. It is a web based interface which will be the web page of the banking application. Starting a page is displayed asking the type of customer he is, whether ordinary or a corporate customer. Then the page is redirected to the login page where the user can enter the login details. If the login particulars are valid then the user is taken to a home page where he has the entire transaction list that he can perform with the bank. All the above activities come under the client interface. The administrator will have an administrative interface which is a GUI so that he can view the entire system. He will also have a login page where he can enter the login particulars so that he can perform all his actions. This administrative interface provides different environment such that he can maintain database

& provide backups for the information in the database. He can register the users by providing them with username, password & by creating an account in the database. He can view the cheque book request & perform action to issue the cheque books to the clients.

**Product Features.**

The Internet banking system consists of following modules :

**1) Login Process**

This module allows valid customers to access the functionalities provided by the bank.

**2) Balance Enquiry**

àThis module maintains the balance details of a particular account.

**3) Update Profile**

àThis module allows the customer to update the profile of their account.

**4) Funds Transfer**

àThis module allows the customers to transfer funds from one account to another within the same bank.

**5) Change of Password**

àThis module allows customers to change their password.

**6) Mini Statements**

This module allows customers to view their transaction details.

**User Classes & Characteristic**

Customers : The normal users will have an account of fixed or savings and should have a minimum balance of Rs. 500. He can transfer funds to another account of the same bank & may view his monthly or annual statements.

Industrialists, Entrepreneur, Organizations academicians: These users will have all the three accounts & should have a minimum balance of 20,000 Rs. He can view the statements of his organization or industry.

**Operating Environment.**

Server Side

Hard Drive More than 160 GB.

RAM More than 1 GB.

Processor Pentium 4 or Higher.

Client Side

Hard Drive More than 160 GB.

RAM More than 1 GB.

Processor Pentium 4 or Higher.

**Design & Implementation Constraint.**

1. This system works only on a single server
2. This is designed in ASP.Net.
3. Language used is C#
4. Limited to HTTP/HTTPS protocols.

**User Documentation.**

A registered user can have following facilities:

1. Accounts and accounts status.
2. The balance enquiry.
3. The fund transfer standards.
4. Cheque Book Request.
5. Password Changing.

**Assumption & Dependencies.**

Assumption:

The details of customers such as username, password, account type and their corresponding authority details should be manually entered by the administrator before using this system.

•Every user should be comfortable working with computers and net browsing.

He should be aware of the banking system. He must have basic knowledge of English too.

**EXTERNAL INTERFACE REQUIREMENT.**

**User Interface**

There are four different ways for a user to interact with the system:

Viewers:

Many unknown persons or un-authenticated persons visit the Bank official site via the internet. They collect the information and search what schemes are available on the bank web page. Those viewers or visitors became customers of the bank.

New User:

Who all visited that Bank webpage or heard about the bank those persons getting ready to start an account in the bank. They register the bank application form, submit and start an account in the nearest bank. Existing User:

The Existing user is the most typical user of the Online Banking system. Each Users have their own account and registered or authorized login access. The Existing user can login in online to their account and perform the operation of deposit, withdrawal, transfer, balance queries and transactions. All the operations of the banking online are helpful for users because it saves time and is an efficient process.

Administrator:

Admin is the master user of the system because they are the main role of the system. Admin grant and maintain the database of the existing user and grant the permissions to users. It over rules all other users

**Hardware Interface**

Client Side:

User on Internet : Web Browser, Operating System (any)

Application Server : WAS

Database Server : DB2

Network : Internet

Development Tools : ASP.Net, HTML,OS(Windows).

Server Side:

Processor : Pentium IV or higher.

RAM : 1 GB or more.

Disk Space : More than 160 GB.

**Software Interface.**

User on Internet : Web Browser, Operating System (any)

Application Server : WAS

Database Server : DB2

Network : Internet

Development Tools : ASP.Net, HTML, OS(Windows),

**Other Non-functional Requirements.**

**Performance Requirement.**

System can withstand even though many customers request the desired service. Access is given to only valid users of bank who requires the services such as balance enquiry, update profile, funds transfer, mini statements, and request for stop payments and for cheque book

It is available all week for all 24 hours.

**Safety Requirement.**

By incorporating a secure database and proven DB2 UDB into the system, reliable performance and integrity of data is ensured. There must be a power backup for the server system. Since the product is of 24x7 availability there should be power backup for the server which provides the information . Every day the data should be backed up even when the operation of a user is not successful i.e., while performing the operation power failure occurs then data should be backed up.

**Security Requirement**

Sensitive data is protected from unwanted access by user’s appropriate technology and implementing strict user-access criteria. Facility of unique user number and Password in such a way that unauthorized user cannot log in. Operational rights for each user/terminal can be defined. Thus,a user can have access to specific terminals and specific options only.

Online Banking uses the SSL (Secure Socket Layer) protocol for transferring data. SSL is encryption that creates a secure environment for the information being transferred between customer browser and Bank. Online Banking uses a 128-bit digital certificate from VeriSign for encryption of the Secure Sockets Layer (SSL) session. SSL is the industry standard for encrypted communication and ensures that customer's interaction with the Bank over the Internet is secure. Secure Socket Layer (SSL) protects data in three key ways:

Authentication:- ensures that you are communicating with the correct server. This prevents another computer from impersonating a Bank.

Encryption :- scrambles transferred data.

Data integrity :- verifies that the information sent by the customer to the Bank wasn't altered during the transfer. The system detects if data was added or deleted after the customer sent the message. If any tampering has occurred, the connection is dropped

**Software Quality Attributes.**

**Usability.**

The users of the system are members and the administrators who maintain the system. The members are assumed to have basic knowledge of the computers and Internet browsing. The administrators of the system have more knowledge of the internals of the system and are able to rectify the small problems that may arise due to disk crashes, power failures and other catastrophes to maintain the system.

The proper user interface, user’s manual, online help and the guide to use and maintain the system must be sufficient to educate the users on how to use the system without any problems.

**Reliability.**

The system is safety critical. If it moves out of normal operation mode, the requirement to drop to the next lower floor and open its doors is given priority. This emergency behaviour shall not occur without reason.

The system has to be very reliable due to the importance of data and the damages incorrect or incomplete data can do.

**Availability.**

When in normal operating conditions, requests by a user for a servicer shall be handled within 1second. Immediate feedback of the system's activities shall be communicated to the user by link page clicked. At peak system load, individual users at either the server in the security office, at the links or inside the banking system shall not experience any delay in the service response to their commands longer than 1 second.

The system is available 100% for the user and is used 24 hrs. A day and 365 days a year. The system shall be operational 24 hours a day and 7 days a week.

**Security.**

There shall be no security mechanisms in place to keep unwanted users out of the system. However, all users of the system shall not be able to perform actions or request actions from the Banking system, which will cause harm to any person or damage to the system or its environment.

**Maintainability.**

There shall be design documents describing the internal works of the software. There shall be access on the control panel and servers for the purpose of upgrading the software or flashing any firmware.

**Portability.**

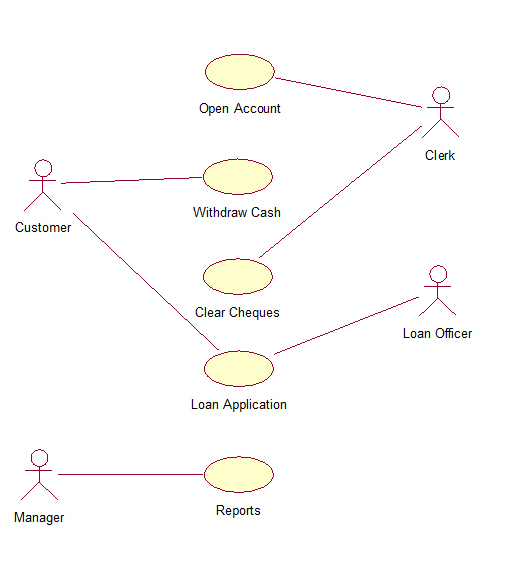
There are no portability requirements. Requirement Organization: All requirements shall be organized according to object. First general requirements for all service types shall be described. Following are sections for each service type and their special requirements. Last are requirements related to other objects like the users view pages and any other.

**System Design.**

Use Case diagram

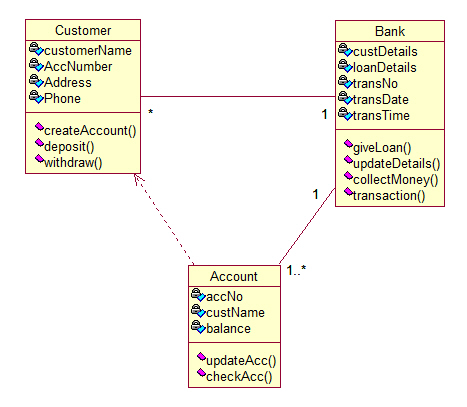
A use case diagram in the Unified Modeling Language(UML) is a type of behavioral diagram defined by and created from a use-case analysis.its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals(represented as use cases),and any dependencies between those use cases.

Use case diagrams are formally included in two modeling languages defined by the OMG:unified modeling language(UML) and the systems modeling language(sysML)



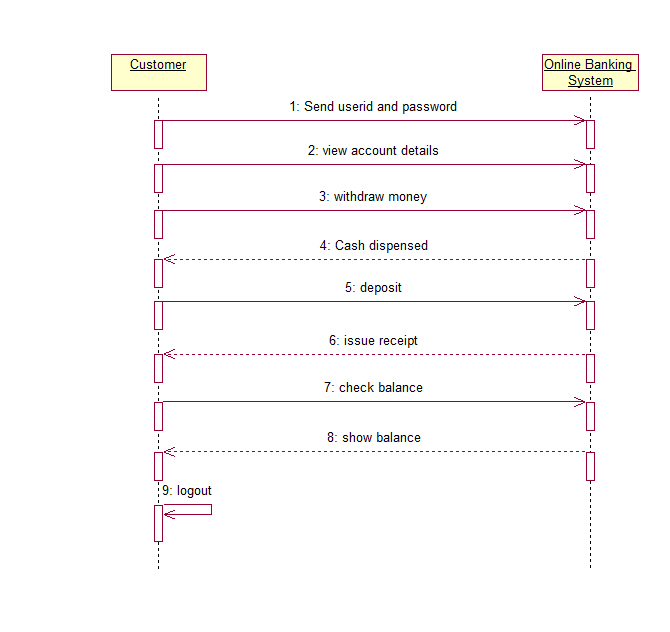
Class Diagram

A Class is a category or group of things that has similar attributes and common behavior. A Rectangle is the icon that represents the class it is divided into three areas. The upper most area contains the name, the middle; area contains the attributes and the lowest areas show the operations. Class diagrams provide the representation that developers work from. Class diagrams help on the analysis side, too.

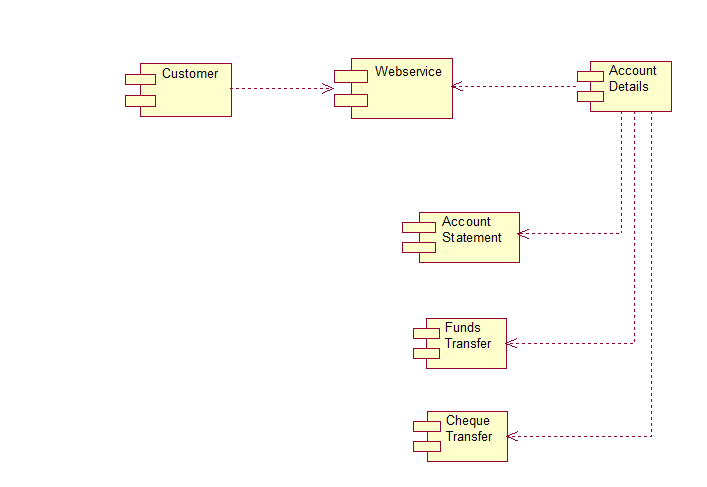


Sequence Diagram

A Sequence Diagram is an interaction diagram that emphasizes the time ordering of messages; a collaboration diagram is an interaction diagram that emphasizes the structural organization of the objects that send and receive messages. Sequence diagrams and collaboration diagrams are isomorphic, meaning that you can take one and transform it into the other.



Component Diagram

/

# RESULT

Thus the project to develop an Online Banking system using Rational Rose Software is done successfully.

**Experiment 9:**

**Develop test cases for Unit Testing and Integration Testing**

**Procedure:**

**Unit Testing :** is a type of software testing where individual units or components of a software are tested. The purpose is to validate that each unit of the software code performs as expected. Unit Testing is done during the development (coding phase) of an application by the developers. Unit Tests isolate a section of code and verify its correctness. A unit may be an individual function, method, procedure, module, or object.

**Objective of Unit Testing**

The objective of Unit Testing is:

1. To isolate a section of code.
2. To verify the correctness of the code.
3. To test every function and procedure.
4. To fix bugs early in the development cycle and to save costs.
5. To help the developers to understand the code base and enable them to make changes quickly.
6. To help with code reuse

**Unit Test Tools:**

Here are some commonly used Unit Testing tools:

1. Jtest
2. Junit
3. NUnit
4. EMMA
5. PHPUnit

**A Java Program to get and print a message “Hello World”**

**App.java**

package com.mycompany.app;

/\* Hello world!\*/

public class App {

private final String message = "Hello World!";

public App() {}

public static void main(String[] args) {

System.out.println(new App().getMessage());

}

private final String getMessage() {

return message;

}

**Code for Unit Test**

**AppTest.java**

import java.io.ByteArrayOutputStream;

import java.io.PrintStream;

import org.junit.Before;

import org.junit.Test;

import org.junit.After

import static org.junit.Assert.\*;

/\* Unit test for simple App. \*/

public class AppTest

{

private final ByteArrayOutputStream outContent = new ByteArrayOutputStream();

@Before

public void setUpStreams() {

System.setOut(new PrintStream(outContent)); }

@Test

public void testAppConstructor() {

try {

new App();

} catch (Exception e) {

fail("Construction failed.");

}

}

@Test

public void testAppMain() {

App.main(null);

try {

assertEquals("Hello World!" + System.getProperty("line.separator"), outContent.toString());

}

catch (AssertionError e) {

fail("\"message\" is not \"Hello World!\"");

}

}

@After

public void cleanUpStreams() {

System.setOut(null);

}

}

**Note:**  Compile and Execute the code using Java (Eclipse IDE can be used)