**EXPERIMENT-3(a)**

**AIM**: **To draw use case diagram**

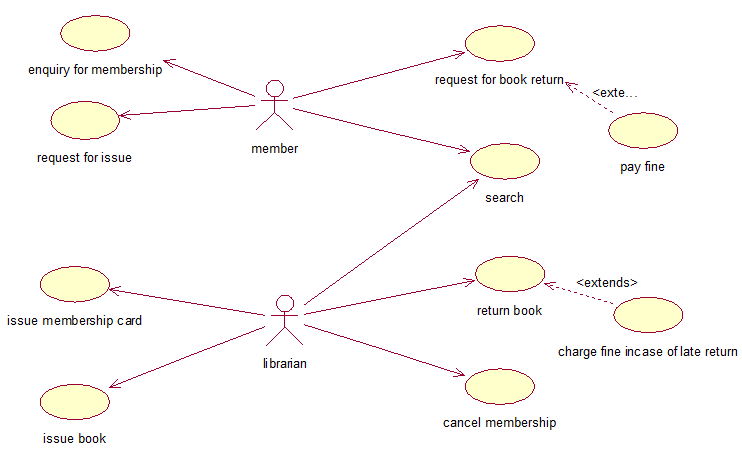
1. **Library Management System**
2. **Railway Reservation System**
3. **Hospital Management System**

**REQUIREMENTS**: Rational Rose enterprise

**THEORY**: A [use case](https://searchsoftwarequality.techtarget.com/definition/use-case) is a methodology used in system analysis to identify, clarify, and organize system requirements. In this context, the term "system" refers to something being developed or operated, such as a mail-order product sales and service [Web site](https://searchmicroservices.techtarget.com/definition/Web-site). Use case diagrams are employed in [UML](https://searchsoftwarequality.techtarget.com/definition/Unified-Modeling-Language) (Unified Modeling Language), a standard notation for the modeling of real-world objects and systems.

**DIAGRAM**:

1. **Library Management System**



**DESCRIPTION:**

* 1. Actors

1. Member
2. Librarian

1. Use case
2. Enquiry for membership

      Introduction: the customer enquires about the membership

Participating user: Member

Flow of events:

1. Basic flow: 1) the person enquires about the membership

                         2)  The department generates an enquiry number

1. Alternate flow- none

Pre-conditions- none

Post-conditions: the department must provide the customer with an enquiry number.

1. Request for issue

      Introduction: the customer requests for issue

Participating user: Member

Flow of events:

Basic flow: 1) the customer requests for issue

Alternate flow- none

Pre-conditions- he should be a Member

      Post-conditions: the department must provide the customer with book if available.

1. Request for book return

Introduction: the customer requests for book return.

Participating user: Member

Flow of events:

Basic flow: the customer requests for return

Alternate flow: none

Pre-conditions: the customer must be a Member and he must have issued the book

Post-condition: he will not have the issued book with him.

1. Search

Introduction: the customer will search for the book

Participating user: Member

Flow of events

Basic flow: he would search for the book on online portal.

Alternate flow: he may enquire for the book.

Pre-conditions: he must be a Member.

Post-condition: he may or may not issue the book

1. Issue book

Introduction: the Member will issue the book

Participating user: Member, Librarian

Flow of events

Basic flow: he would enter is Membership id and then he will be able to issue the book.

Alternate flow- none

Pre-conditions: he must be a Member

Post-condition: he will have the book issued in his account.

1. Return book

Introduction: the person would request for book return.

Participating user: Librarian

Flow of events

Basic flow: he will give the book back.

Alternate flow: his account will be updated.

Pre-conditions: the customer must be a Member and he must have issued the book.

Post-condition: the book issued will be deleted from his account.

1. Issue Membership card

Introduction: the customer would ask for a Membership card

Participating user: Librarian

Flow of events

Basic flow: he will show the identity proof and fill the form for Membership.

Alternate flow: none

Pre-conditions: he must not be a Member.

Post-condition: he will have filled the form

1. Cancel Membership

Introduction: the Member will request for cancelation of Membership

Participating user: Member, Librarian

Flow of events

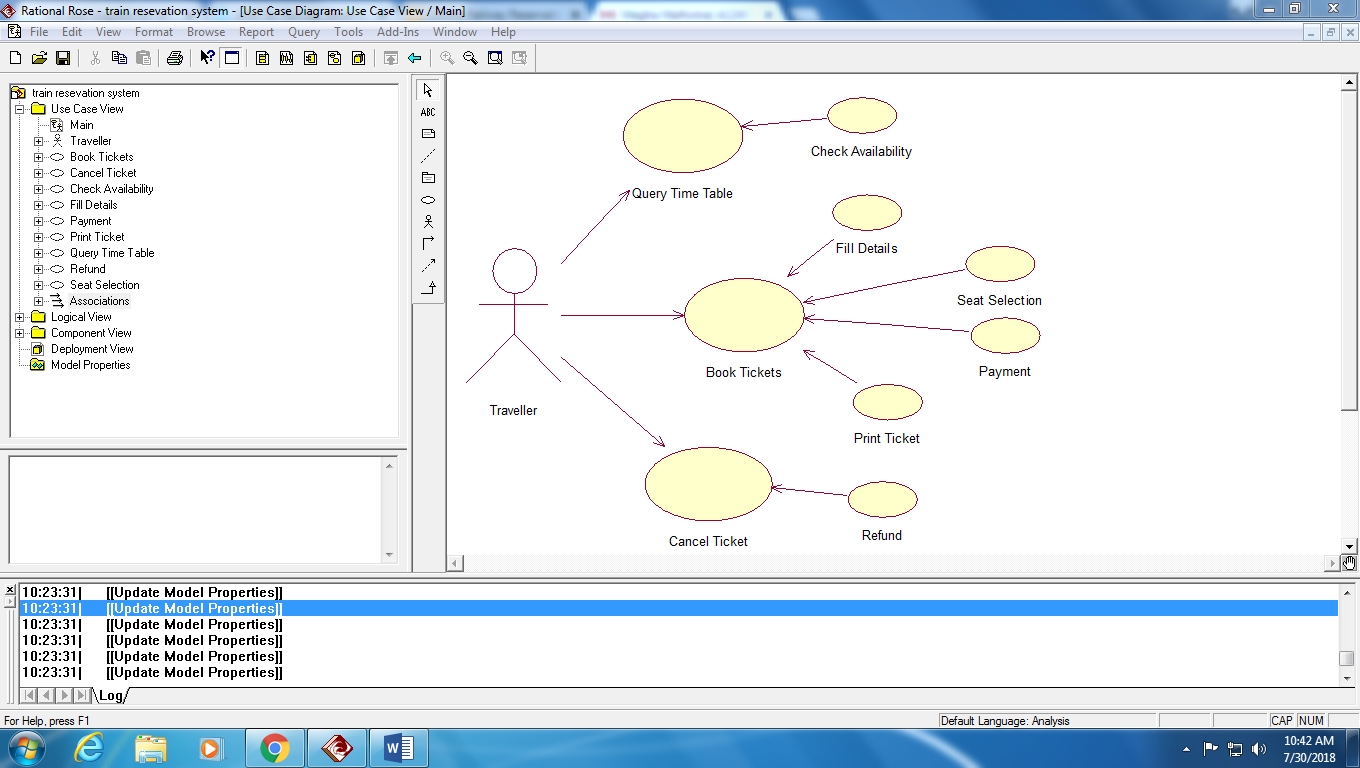
Basic flow: the person will apply for Membership cancellation

Alternate flow: the issued books will be returned.

Pre-conditions: he must be a Member and he must not have any book issued.

Post-condition: none

1. **Railway Reservation System**



**DESCRIPTION:**

* 1. Actors

1. Traveler
   1. Use case
2. Query time table

Introduction: the person will check for train time table

Participating user: Traveler

Flow of events:

Basic flow: the person will access the online portal and will create a id.

Alternate flow: the id will be created.

Pre-conditions: he must have the id.

Post conditions: he may book a ticket.

1. Check availability

Introduction: the Traveler will check for availability of seats.

Participating user: Traveler

Flow of events:

Basic flow: he will login with the registered id.

Alternate flow: none

Pre-conditions: he must have a login id.

Post conditions: he may book a ticket.

1. Book tickets

Introduction: the Traveler will book the ticket.

Participating user: Traveler

Flow of events:

Basic flow: he will login with the registered id.

Alternate flow: seat allotment

Pre-conditions: he must have a login id.

Post conditions: he may cancel the ticket.

1. Fill details

Introduction: the Traveler may fill details to get an id.

Participating user: Traveler

Flow of events:

Basic flow: he will apply for id and password

Alternate flow: none

Pre-conditions: he should have a valid identity proof

Post conditions: none

1. Seat selection

Introduction: Traveler will request for seat allotment.

Participating user: Traveler

Flow of events:

Basic flow: traveler will tell the ticket number and then seat allotment will be done.

Alternate flow: none

Pre-conditions: he must have an id and he must have paid the fare.

Post conditions: seat will be allotted.

1. Payment

Introduction: the traveler will apply for payment

Participating user: Traveler

Flow of events:

Basic flow: the traveler will apply of ticket and select a seat preference.

Alternate flow: payment will be deducted according to the mode adopted.

Pre-conditions: he must have a login id

Post conditions: the payment will be received

1. Print ticket

Introduction: the Traveler will request for print of the ticket.

Participating user: Traveler

Flow of events:

Basic flow: the command for print will be given.

Alternate flow: none

Pre-conditions: the traveler must have completed the payment procedure.

Post-conditions: none

1. Cancel ticket

Introduction: the Traveler will cancel the ticket

Participating user: Traveler

Flow of events:

Basic flow: the ticket number will be asked and refund process will be carried

Alternate flow: none

Pre-conditions: the ticket must be booked and the payment must be made.

Post conditions: refund procedures may be applied

1. Refund

Introduction: refund of the ticket fare on cancelation

Participating user: Traveler

Flow of events:

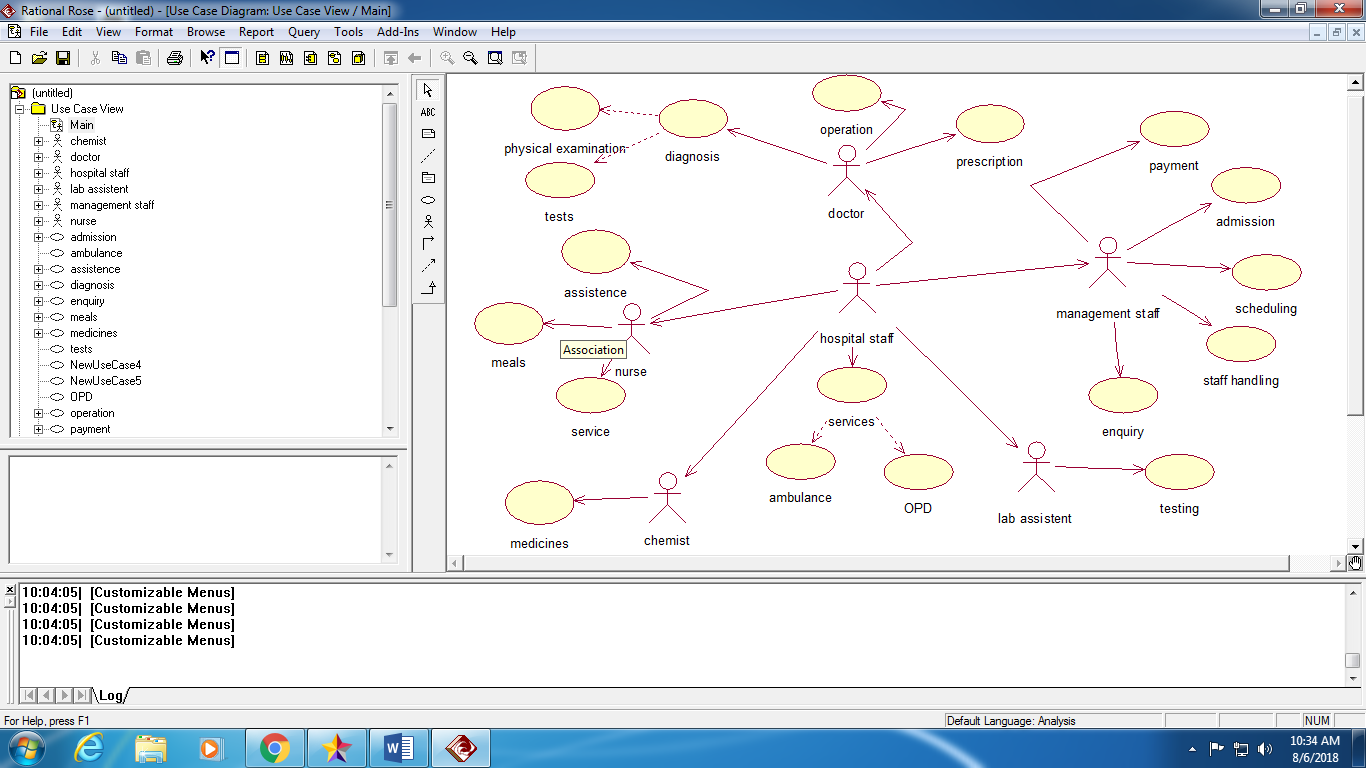
Basic flow: cancel process will be carried out and the refund will be generated.

Alternate flow: none.

Pre-conditions: the ticket must have been booked and the payment must have been made.

Post conditions: money will be transferred back to the account

1. **Hospital Management System**



**DESCRIPTION:**

* 1. Actors
     1. Hospital staff
     2. Doctor
     3. Nurse
     4. Chemist
     5. Management staff
     6. Lab assistant
  2. Use case
     1. Services

Introduction: services will be provided to the patient

Participating user: Hospital staff

Flow of events:

Basic flow: services will be provided according to the request

Alternate flow: none

Pre-conditions: none

Post conditions: get registered in the patient list

* + 1. Ambulance

Introduction: ambulance service will be provided to the patient

Participating user: Hospital staff

Flow of events:

Basic flow: services will be provided according to the request

Alternate flow: none

Pre-conditions: hospital staff must receive any information

Post conditions: get registered in the patient list as soon as the ambulance arrives

* + 1. OPD

Introduction: the patient will be able to access the OPD facilities

Participating user: Hospital staff

Flow of events:

Basic flow: services will be provided by the specific doctors

Alternate flow: none

Pre-conditions: hospital staff must receive any information and registration must be done in advance. Collect fees in advance.

Post conditions: none

* + 1. Prescription

Introduction: the doctor will prescribe the medicines

Participating user: Doctor

Flow of events:

Basic flow: the patient will register for appointment and then visit the doctor accordingly

Alternate flow: the patient may apply for priority visits

Pre-conditions: the patient must be suffering from some disorder

Post-conditions: the patient may visit the chemist

* + 1. Operation

Introduction: the patient will be operated by the surgeons.

Participating user: Doctor

Flow of events:

Basic flow: the surgery will be recommended by the doctor and all the procedures will be carried on accordingly

Alternate flow: there may be any immediate surgery requirements

Pre-conditions: the operation theater must be set up in advance and there should be no delay.

Post-conditions:

* + 1. Diagnosis

Introduction: the doctor will diagnose the diseases if any

Participating user: Doctor

Flow of events:

Basic flow: the doctor will diagnose the problem

Alternate flow: none

Pre-conditions: patient must register at the reception

Post-conditions: patient might be prescribed with some tests and medicines

* + 1. Physical examination

Introduction: the doctor will perform physical examination for diagnosis of some skin disorders etc

Participating user: Doctor

Flow of events:

Basic flow: the doctor will perform physical examination

Alternate flow: none

Pre-conditions: the patient must be registered at the reception

Post-conditions: the patient might be prescribed some tests or medicines

* + 1. Tests

Introduction: tests will be prescribed by the doctor.

Participating user: Doctor

Flow of events:

Basic flow: the patient will approach the doctor and he may suggest some diagnostic tests.

Alternate flow: none

Pre-conditions: the patient must register at reception

Post-conditions: tests may be performed by the lab assistant.

* + 1. Assistance

Introduction: the patient will be assisted by the staff.

Participating user: Nurse

Flow of events:

Basic flow: the patient will be assisted while he needs assistance

Alternate flow: none

Pre-conditions: he must contact the staff when he requires.

Post-conditions: none

* + 1. Meals

Introduction: meals will be served to admit patient as per the diet chart.

Participating user: Nurse

Flow of events:

Basic flow: the nurse will take the food according to the prescribed diet chart to every patient.

Alternate flow: there may be some changes the diet on a regular basis.

Pre-conditions: patient must be allotted with nutritionist and the nurse.

Post-conditions: patient may not be satisfied with food provided or the staff.

* + 1. Service

Introduction: the patient will be provided with the services

Participating user: Nurse

Flow of events:

Basic flow: the nurse would provide room to room service as per the requirement of the patient.

Alternate flow: nurses may give the glucose or some injections or some medicinal doses as recommended by the doctor.

Pre-conditions: the patient must be admitted to the hospital for any kind of service.

Post-conditions: the patient may deny treatment.

* + 1. Medicines

Introduction: medicines are given by a chemist

Participating user: Chemist

Flow of events:

Basic flow: the patient will show the prescription and take the medicines from the chemist.

Alternate flow: none

Pre-conditions: the patient must have visited the doctor once.

Post-conditions: none

* + 1. Testing

Introduction: the lab assistant would perform various tests recommended by the doctor and produce the final reports after tests.

Participating user: Lab assistant

Flow of events:

Basic flow: the patient will produce the slip where doctor has recommended test.

Alternate flow: none

Pre-conditions: the patient must have visited the doctor once.

Post-conditions: the patient will visit the doctor to show reports.

* + 1. Payment

Introduction: the payment will be made at the reception and it will be taken care by the management staff of further bills and receipts.

Participating user: Management staff

Flow of events:

Basic flow: the bill will be prepared and the patient will pay the fees.

Alternate flow: none

Pre-conditions: the bill must be taken care of from the beginning itself.

Post-conditions: payment must be completed on time.

* + 1. Admission

Introduction: the management staff will register the new patients.

Participating user: Management staff

Flow of events:

Basic flow: the patient will come and his information will be saved.

Alternate flow: none

Pre-conditions: none

Post-conditions: the patient may avail other facilities

* + 1. Scheduling

Introduction: the scheduling of patient and doctors will be taken care of.

Participating user: Management staff

Flow of events:

Basic flow: the time management will be done and the patient will be allotted time accordingly.

Alternate flow: none

Pre-conditions: the patient will have registered before allotment of time with the doctor.

Post-conditions: the patient might avail other facilities.

* + 1. Staff handling

Introduction: the hospital staff, their salary, etc will be handled under this.

Participating user: Management staff

Flow of events:

Basic flow: the staff information must be stored in the database.

Alternate flow: none

Pre-conditions: the person must be an employee.

Post-conditions: none

* + 1. Enquiry

Introduction: the person may enquire about anything in hospital

Participating user: Management staff

Flow of events:

Basic flow: the person may ask anything from the reception staff.

Alternate flow: none

Pre-conditions: none

Post-conditions: none

**EXPERIMENT-3(b)**

**AIM**: **To draw class diagram**

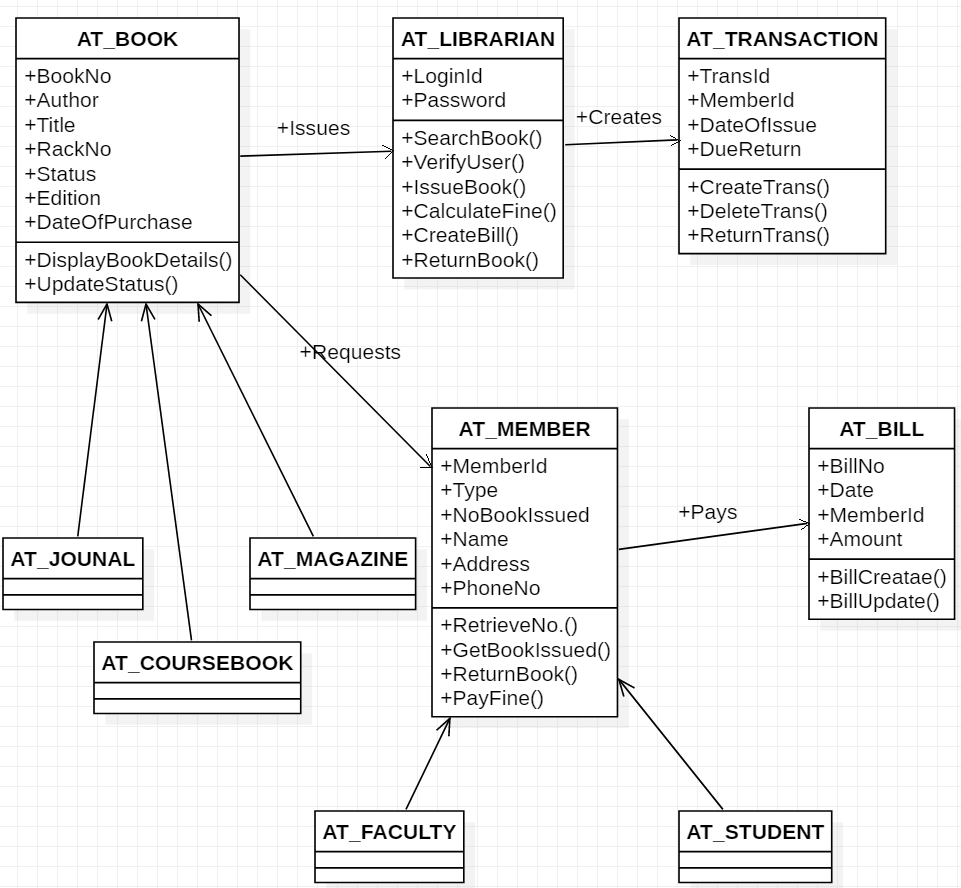
1. **Library Management System**
2. **Railway Reservation System**
3. **Hospital Management System**

**REQUIREMENTS**: Star UML

**THEORY**: Class diagrams are one of the most useful types of diagrams in UML as they clearly map out the structure of a particular system by modeling its classes, attributes, operations, and relationships between objects. With our UML diagramming software, creating these diagrams is not as overwhelming as it might appear. This guide will show you how to understand, plan, and create your own class diagrams.

**DIAGRAM**:

1. **Library Management System**



**DESCRIPTION:**

* 1. Book

Attributes

* + 1. Book id- it specifies the book id.
    2. Author- it specifies the author of the book.
    3. Title- it specifies the title of the book
    4. Rack no. - it specifies the rack no. of the book
    5. Status- it specifies the status of the book
    6. Edition- it specifies the edition of the book
    7. Date of purchase- - it specifies the date of purchase of the book

Operations

1. Display book details- it displays book details.
2. Update status- it updates the status of the book.
   1. Librarian

Attributes

* + 1. Login id- it tells the login id of the person.
    2. Password- it specifies the password of the person.

Operations

1. search book- it is used to search book.
2. verify members- it is used to verify member.
3. issue book- it is used for issuing books.
4. calculate fine- it is used to calculate fine
5. create bill- it is used for creating bills.
6. return book- it is used for returning the book.
   1. Transaction

Attributes

* + 1. Trans id- it tells the transaction id.
    2. Member id- it tells the member id.
    3. Date of issue- it tells the date of issue.
    4. Due return- it is concerned with the return of the book.

Operations

1. Create trans- it creates transaction.
2. Delete trans- it deletes transaction.
3. Retrieve trans- it retrieves transaction.
   1. Member

Attributes

* + 1. Member id- it tells the member id.
    2. Type- it tells the type of member.
    3. No book issued- it specifies the number of books issued.
    4. Name – it tells the name of the person.
    5. Address- it tells the address of the person.
    6. Phone no.- it tells the phone number of the member.

Operations

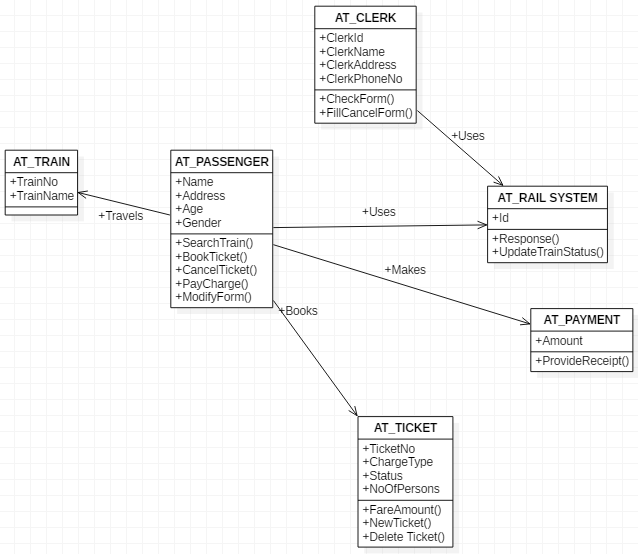
1. retrieve member – it is used to retrieve the member.
2. get book issued- it is used to get the book issued.
3. return book – it is used to return the book.
4. pay fine- it is used to pay the fine.
   1. Bill

Attributes

* + 1. Biil no.- it tells the bill number.
    2. Date – it tells the date of bill.
    3. Member id- it tells the member id.
    4. Amount- it tells the amount to be paid.

Operations

1. Bill create – it helps in creating bills
2. Bill update- it updates the bill.
3. **Railway Reservation System**



**DESCRIPTION:**

* + - 1. Passenger

Attributes

* 1. Name – it tells the name of the customer.
  2. Age- it tells the age of the customer.
  3. Gender- it tells the gender of the customer.
  4. Address- it tells the address of the customer.

Operations

* 1. Search train- it is used to search for trains.
  2. BookTicket- it is used for booking purpose.
  3. CancelTicket- it is used for cancelling tickets.
  4. PayCharge- it is used for paying ticket charges.
     + 1. Train

Attributes

* + - * 1. TrainNo.- it tells the train number.
        2. TrainName- it tells the name of the train.
      1. Ticket

Attributes

* + - * 1. TicketNo.- it specifies the ticket number.
        2. ChargeType- it tells about the charge type.
        3. Status- it tells about the ticket status.
        4. NoOfPerson- it tells about the number of persons.

Operations

* + - * 1. Print ticket- it is used for printing purpose.
        2. NewTicket- it is used for making new ticket
      1. Rail System

Attributes

* + - * 1. Id- it shows user id

Operations

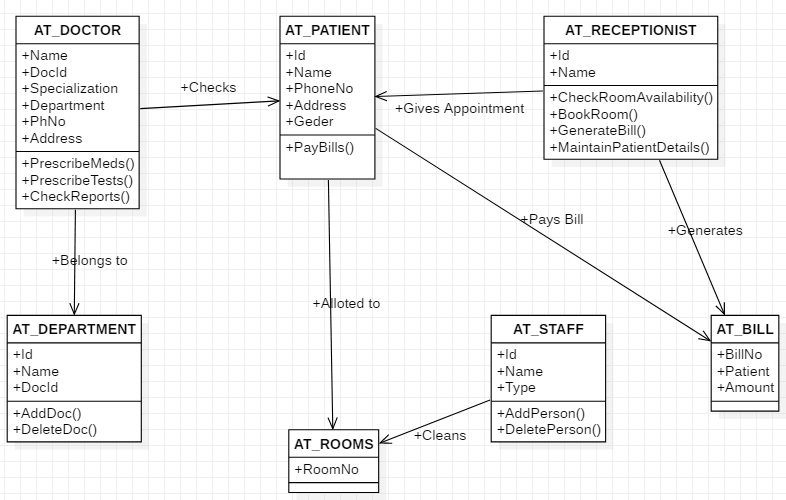
* + - * 1. Response- it is gives response
        2. UpdateTrainStatus- it updates the train status
      1. Payment

Attributes

* + - * 1. Amount – it tells the total amount.

Operations

1. ProvideReciept – it provides the payment reciept.
2. **Hospital Management System**



**DESCRIPTION:**

* 1. Doctor

Attributes

* + - 1. DocId- it specifies the doctor id.
      2. Name- it specifies the doctor name.
      3. Department- it specifies the department.
      4. Specialization- it specifies the doctor specialization.
      5. PhNo. - It specifies the doctor’s phone no.
      6. Address- it specifies the doctor’s address.

Operations

1. Prescribe – it is used for prescription purpose.
2. Check report- it is used to check reports.
   1. Patient

Attributes

* + - 1. Id- it tells the patient id.
      2. Name- it tells the patient name.
      3. Tel no. - it tells the patient number.
      4. Address- it tells the patient address.
      5. Age - it tells the patient age.
      6. Sex- it tells the patient sex.
      7. Room no. - it tells the patient room number.

Operations

1. Pay bills- it is used to pay bills.
   1. Receptionist

Attributes

* + - 1. Id- it tells the id.
      2. name- it tells the name.

Operations

1. check room- it is used to check rooms.
2. book room- it is used to book rooms.
3. generate bill- it is used to generate bills.
4. draw salary- it is used to draw salary.
5. keep patient details- it keeps patient details.
   1. Department

Attributes

* + - 1. Id- it tells the id.
      2. Name- it tells the name.
      3. Doctor id- it tells the id of doctor.

Operations

1. Add department- it is used to add department.
2. Delete department- it is used to delete department.
3. Add doctor- it is used to add doctor.
4. Delete doctor- it is used to delete doctor.
   1. Room

Attributes

* + - 1. Room no.- it tells the room number.
      2. Location- it tells the location.
  1. Staff

Attributes

* + - 1. Id- it tells the id.
      2. Name- it tells about name.
      3. Type- it tells the staff type.

Operations

1. draw salary- it is used to draw the salary.
2. add person- it is used to add a person.
3. delete person- it is used to delete a person.
4. edit person- it is used to edit the details of the person.
   1. Bill

Attributes

* + - 1. Bill no.- it specifes the bill number.
      2. Patient name – it specifies the patient name.
      3. Amount- it tells the amount of bill.

**EXPERIMENT-3(c)**

**AIM: To create sequence diagram for**

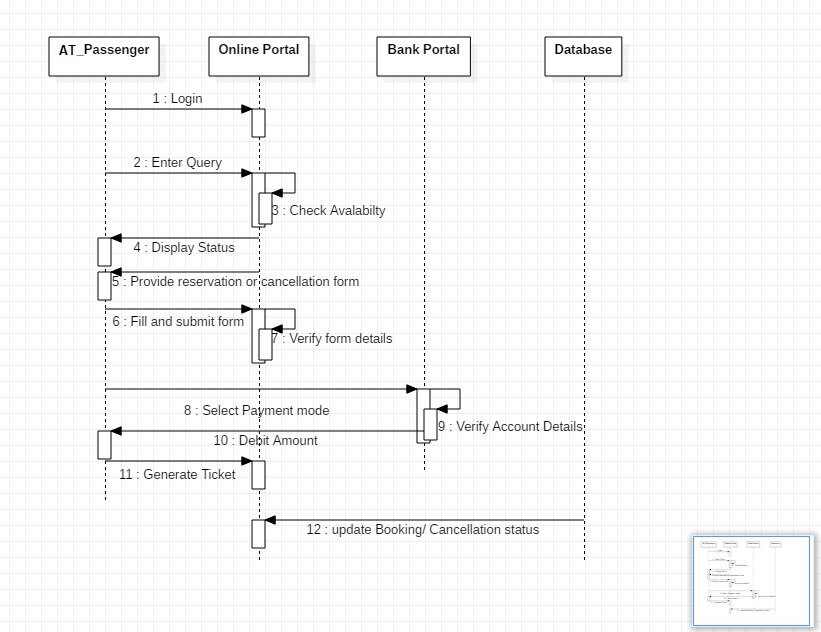
1. **Railway Management System**
2. **Library Management System**
3. **Hospital Management System**

**REQUIREMENTS:** Star UML

**THEORY:** Sequence diagrams describe interactions among classes in terms of an exchange of messages over time. A sequence diagram is a good way to visualize and validate various runtime scenarios. These can help to predict how a system will behave and to discover responsibilities a class may need to have in the process of modeling a new system. It describes how—and in what order—a group of objects works together. These diagrams are used by software developers and business professionals to understand requirements for a new system or to document an existing process. Sequence diagrams are sometimes known as event diagrams or event scenarios.

**DIAGRAM:**

1. **Railway Management System**



**DESCRIPTION:**

1. Passenger and Online Portal

Functions

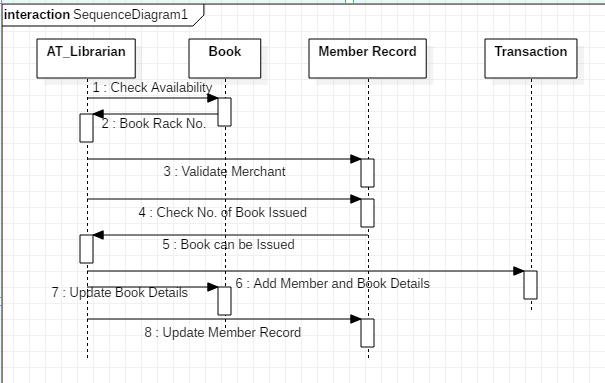
1. Passenger logins into the online portal.
2. Passenger enter query into the online portal.
3. Online portal checks availability of seats and train status.
4. Online portal displays status of the trains to the passenger.
5. Online portal provides either reservation or cancellation form to the passenger.
6. Passenger fills and submits the form to the online portal.
7. Online portal verifies the details filled by the passenger.
8. Online portal generates the desired ticket for the passenger.
9. Passenger and Bank Portal

Functions

1. Passenger selects the payment mode for making payment for the ticket.
2. Bank portal verifies the details entered by the passenger for payment.
3. The amount is debited from the passenger’s account by the bank portal.
4. Online Portal and Database

Functions

1. The online portal updates the status and availability of the trains, after a passenger books a ticket, in the database.
2. **Library Management System**



**DESCRIPTION**

1. Librarian and Book

Functions

1. Librarian checks the availability of the book into library.
2. If book is available in the library it will known to librarian.
3. Librarian will calculate the fine amount of the book.
4. Librarian will update the status of book like availability or not.
5. Librarian and Member Record

Functions

1. Librarian will verify the member record whether it is good or bad.
2. Members record will provide the availability of book to Librarian or how many book is issued by the member.
3. Librarian will update the member record like issuing, fine , bill ,etc.
4. Librarian and Bill

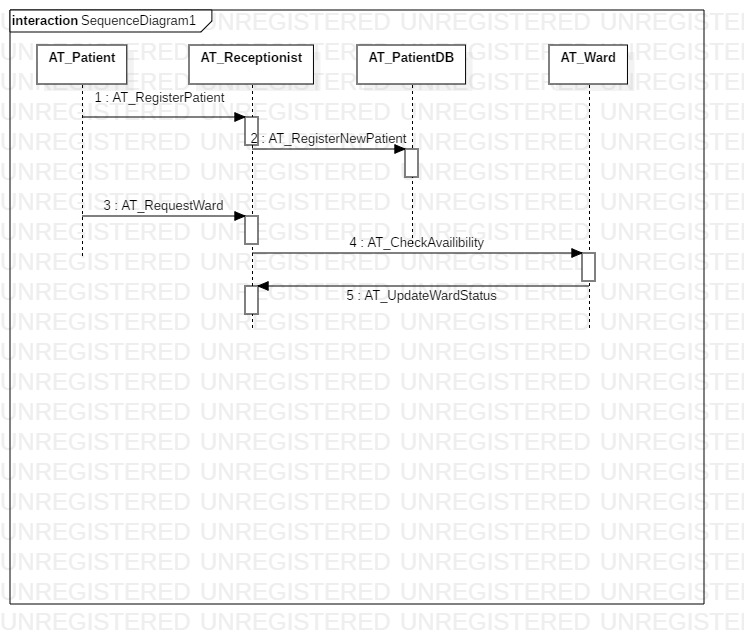
Functions

1. Librarian will generate the bill of book issued by the member.
2. Member will pay fine of book to the Librarian.

1. Book and Member Record

Functions

1. It simply tell that how many book is issued by member.
2. **Hospital Management System**



**DESCRIPTION:**

1. Patient and Receptionist

Functions

1. Patient asks for the appointment of doctor from the reception.
2. Receptionist confirms the appointment from the doctor to the patient.
3. Receptionist asks for the payment from the patient.
4. Patient pays the consultation fee at the reception to the receptionist.
5. Receptionist and Doctor

Functions

1. Receptionist takes appointment on behalf of the patient from the doctor.
2. Receptionist checks if the doctor is available.
3. Doctor checks for the free slots and timings when he is available.
4. Receptionist will fix the patient’s appointment.
5. Patient and Doctor

Functions

1. Patient consults the doctor for checkup and medication.
2. Doctor treats the patient and provides the medical aid and services.
3. Doctor and Nurse

Functions

1. Nurse assists the doctors in providing proper health care medication to patient.
2. Doctor needs a nurse in Operation Theater to assist him while a surgery is performed.

**EXPERIMENT-3(d)**

**AIM: To create activity diagram for**

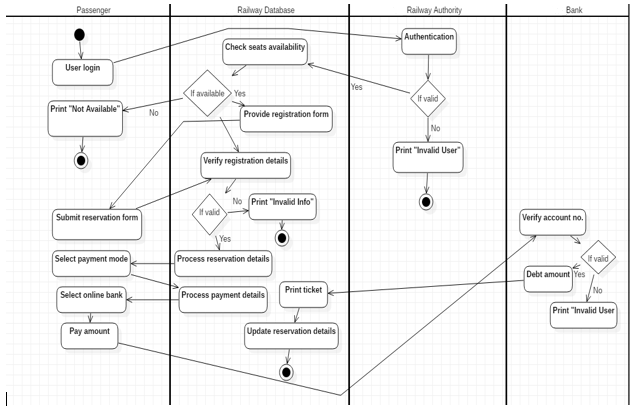
1. **Railway Management System**
2. **Library Management System**
3. **Hospital Management System**

**REQUIREMENTS:** Star UML

**THEORY:** Activity diagram is used to show message flow from one activity to another. Activity is a particular operation of the system. Activity diagrams are not only used for visualizing the dynamic nature of a system, but they are also used to construct the executable system by using forward and reverse engineering techniques. The only missing thing in the activity diagram is the message part. It does not show any message flow from one activity to another. Activity diagram is sometimes considered as the flowchart. Although the diagrams look like a flowchart, they are not. It shows different flows such as parallel, branched, concurrent, and single.

**DIAGRAM:**

1. **Railway Management System**

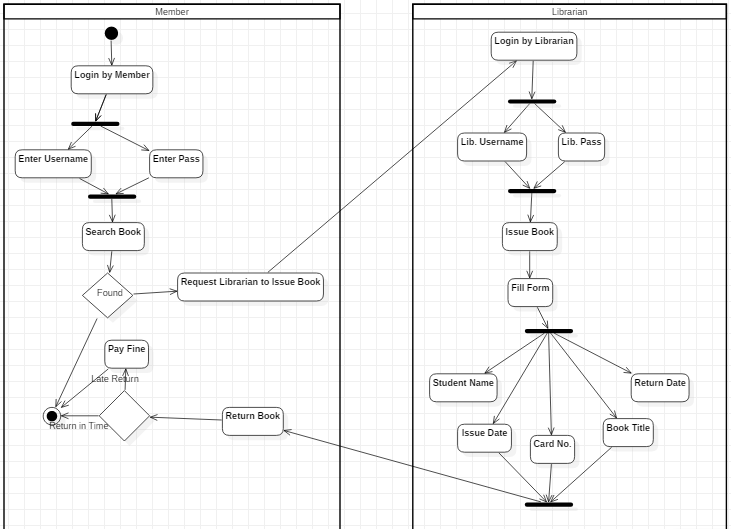


**DESCRIPTION:**

1. User login - The user enters using his details.
2. Authentication - The railway authority checks if the user is authentic or not.
3. Condition (if valid)
4. Check seats available [Yes] - The railway database checks for the availability of seats
5. Print "Invalid User" [No] - The user details entered are invalid.
6. Condition (if available)
7. Provide registration form [Yes] - The user is provided with a registration form.
8. Print "Not Available" [No] - No tickets are available.
9. Submit reservation form - The user submits the reservation form.
10. Verify registration details - The railway database checks the details in the reservation form.
11. Condition (if valid)
12. Process reservation details [Yes] - The railway database processes the registration details.
13. Print "Invalid Info" [No] - The details entered are not valid.
14. Select payment mode - The user selects how he wishes to pay the ticket amount.
15. Process payment details - The railway database processes the payment details.
16. Select online bank - The user selects the bank through which he wishes to pay.
17. Pay amount -The user pays the amount.
18. Verify account number - The bank verifies the account number entered by the user.
19. Condition (if valid)
    1. Debt amount [Yes] - The amount is paid to the railway authority.
    2. Print "Invalid User" [No] - The user details entered are invalid.
20. Print ticket - The railway database prints the ticket for the user.

15.  Update reservation details - The railway database updates the reservation details in the database.

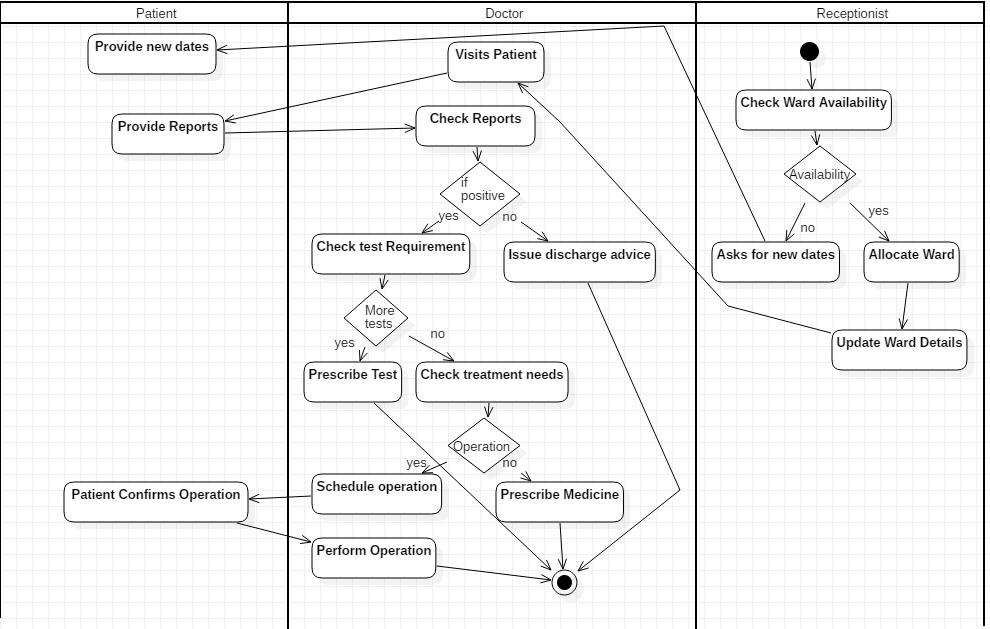
1. **Library Management System**



**DESCRIPTION:**

1.Login by student -The student wishes to enter the library software.

1. Enter username -The student enters his username at the login page.
2. Enter password - The student enters his password at the login page.
3. Search for book - The student searches for the book he needs.
4. Condition (found)
   1. Request librarian to issue book - Student asks the librarian to issue him the book.
5. Login by librarian -The librarian wishes to enter the library software.
6. Enter username - The librarian enters his name at the login page.
7. Enter password - The librarian enters his password at the login page.
8. Issue book - The librarian issues book to the student.
9. Fill form - Librarian fills the form for issue of book
10. Name of member
11. Return date
12. Issue date
13. Card no.
14. Book title
15. Return book - The student wants to return the book he issued.
16. Condition
17. Pay fine (late) - The student pays the fine for late return.
18. End (Returned in time).
19. **Hospital Management System**



**DESCRIPTION:**

1. Check ward availability: The patient gets admitted to a hospital by checking ward availability.
2. Condition (availability) [Yes]:
3. Allocate ward: The room is allocated to the patient.
4. Condition (availability) [No]:
5. Ask for new dates: Receptionist asks for new dates if the ward is not available.
6. Provides new date: Patient confirms the dates according to the need and emergency.
7. Update ward details: The details of the ward are updated and charges are applied depending upon the services by the receptionist.
8. Visits patient: The doctor visits the patient after the ward is been allocated successfully.
9. Provides reports: The doctor provides the reports to the patient after the checkup.
10. Check reports: The patient checks reports and visits the doctor for further processes.
11. Condition (if positive):
12. Check tests requirements [Yes]: The doctor checks the requirements for further tests.
13. Issue discharge advice [No]: Doctor advises patient discharge if the reports are not positive.
14. Condition (more tests):
15. Prescribe tests [Yes]: If required the doctor prescribes the further tests to be done.
16. Check treatment needs [No]: The further treatment needs are checked upon.
17. Condition (operation) [Yes]:
18. Schedule operation: If there is a need for the operation then it is scheduled.
19. Patient confirms operation: If the patient is ready for the operation then it takes place.
20. Perform operation: Doctor performs the operation according to the diseases.
21. Condition (operation) [No]:
22. Prescribe medicine: Doctor prescribes medicines if there is no need of operation.

**EXPERIMENT-3(e)**

**AIM: To create state transition diagram for**

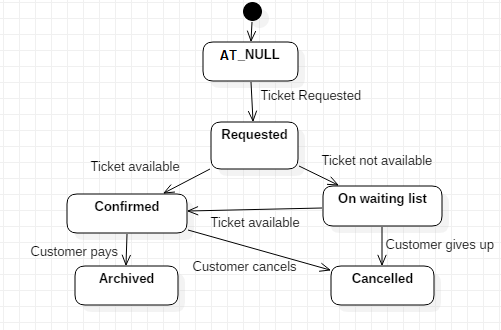
1. **Railway Management System**
2. **Library Management System**

**REQUIREMENTS:** Star UML

**THEORY:** State-transition diagrams describe all the states that an object can have, the events under which an object changes state (transitions), the conditions that must be fulfilled before the transition will occur (guards), and the activities undertaken during the life of an object (actions). State-transition diagrams are very useful for describing the behavior of individual objects over the full set of use cases that affect those objects. State-transition diagrams are not useful for describing the collaboration between objects that cause the transitions.

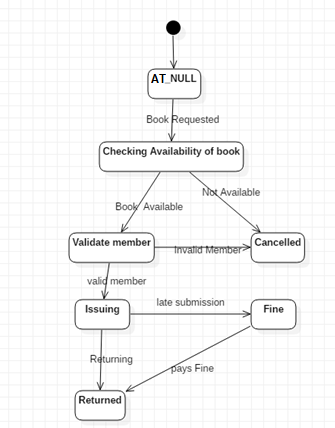
**DIAGRAM:**

1. **Railway Management System**



**DESCRIPTION:**

1. Null- This is the initial stage. We are in this state when nothing is being done.
2. Requested- The system jumps to this state from the Null state once a ticket is requested. This state means that a request from the user has been registered and is being processed.
3. Confirmed- This state means that the ticket that the user needed has been booked. The system comes to this state from the Requested state if the desired ticket is available.
4. On Waiting List-  The system goes to this state if the ticket that the user requested is not available.
5. Archived- This is one of the final states of the system. This is reached when the process of booking of ticket is complete with the user having paid the amount of the ticket. In this state, the information is updated in the database.
6. Cancelled- This state is reached in two ways. Firstly, when the system is in On Waiting List state and the user decides to give up. Secondly, when the system is in Confirmed state and the user decides to cancel his ticket. This is also a final state.
7. **Library Management System**



**DESCRIPTION:**

1. Null- This is the initial stage. We are in this state when nothing is being done.
2. Checking availability of book- The system jumps to this state from the Null state when a book is requested. This state means that a request from the user is being processed.
3. Validate member- The system checks if the member is valid or not after the book is available.
4. Cancelled- The system goes to the Cancelled state if the person is an invalid member.
5. Issuing- After the validity of the member is checked, the system jumps from Validate member state to the Issuing state.
6. Fine- The system jumps from the Issuing state to this state due to late submission of the book.
7. Returned- This state is reached in two ways. Firstly, when the book is returned by the member. Secondly, when the member pays fine for the late submission.

**EXPERIMENT-3(f)**

**AIM: To create communication diagram for**

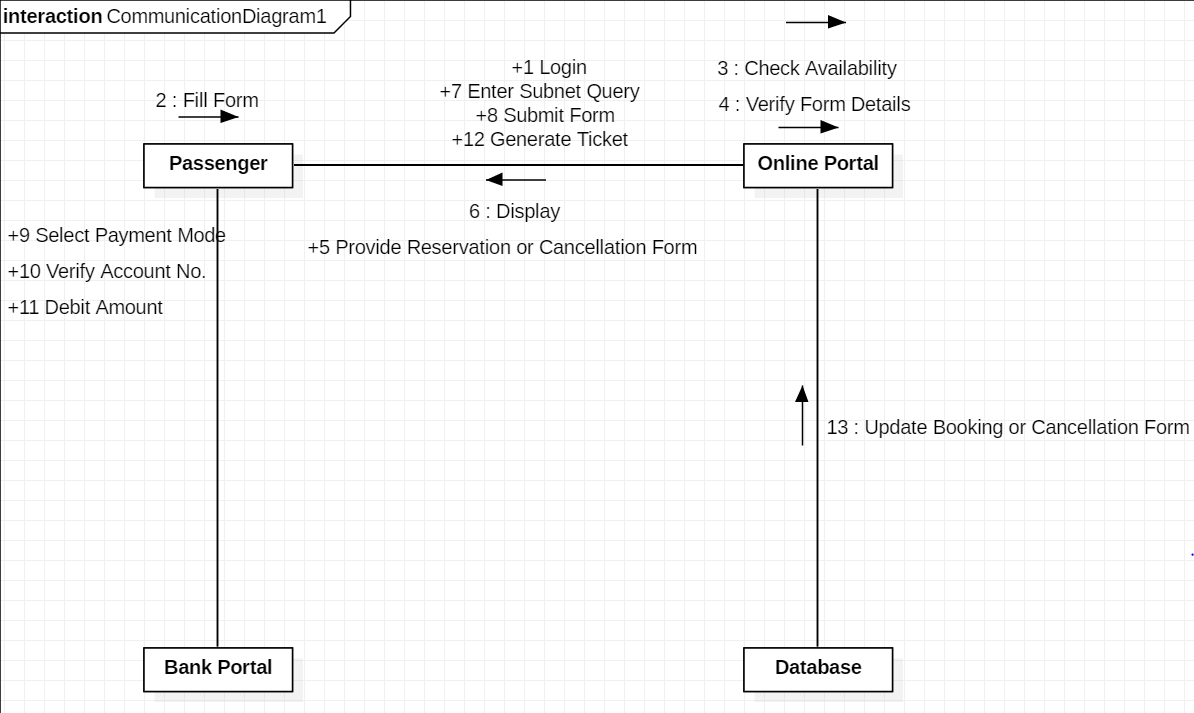
1. **Railway Management System**
2. **Library Management System**

**REQUIREMENTS:** Star UML

**THEORY:** Communication diagram is an illustration of relationships and interactions among software objects in UML. The concept is more than a decade old although it has been refined as modelling paradigms have evolved. A collaboration diagrams a flowchart that portrays the roles, functionality and behavior of individual objects as well as the overall operation of the system in real time. Objects are shown as rectangles with naming labels inside. These labels are preceded by colons and may be underlined.

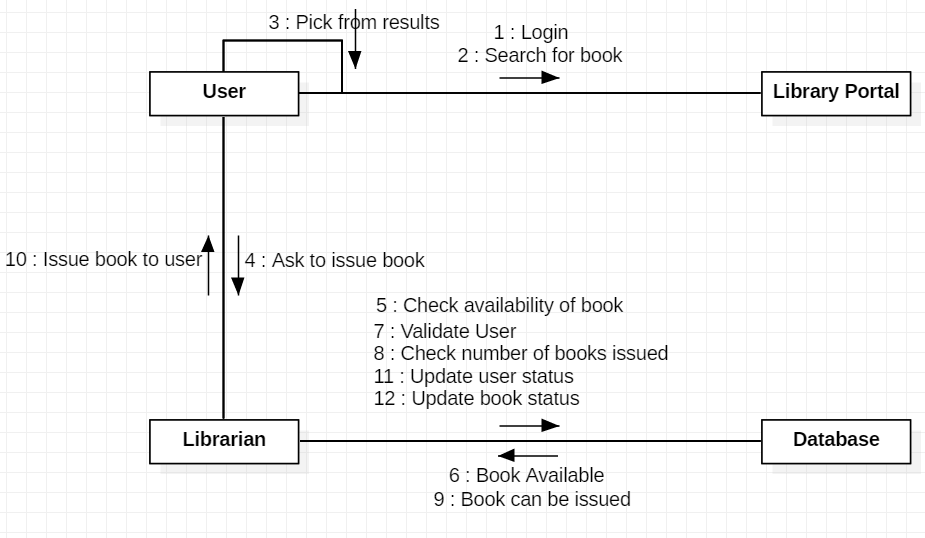
**DIAGRAM:**

1. **Railway Management System**



**DESCRIPTION:**

1. Passenger and Online Portal
2. The passenger logs into the online portal.
3. The passenger fills form.
4. The passenger submits the form.
5. The online portal checks availability of train.
6. The online portal verifies the form details.
7. The passenger enters the query.
8. The online portal provides reservation or cancellation form.
9. The online portal generates ticket.
10. The online portal displays it to passenger.
11. Passenger and Bank Portal
12. The passenger selects the payment mode.
13. The bank portal debits the amount to be paid.
14. The bank portal verifies the account number.
15. Online Portal and Database
16. The databases updates booking or cancellation form.
17. **Library Management System**



**DESCRIPTION:**

1. User and Library Portal
2. The user logins into the library portal.
3. The user searches the required book.
4. The library portal shows the result matching the searched book.
5. The user picks one of the books from the result.
6. User and Librarian
7. The user asks the librarian to issues the book.
8. The librarian issues the book to the user.
9. Librarian and Database
10. The librarian checks the availability of the book in the database.
11. The database responds the book is available.
12. The librarian checks if the book user asked is valid or not.
13. The librarian checks the number of books already issued by the user.
14. The database responds that the book can be issued by the user.
15. The librarian updates the records in the database.

**EXPERIMENT-3(g)**

**AIM: To create component diagram for Railway Management System.**

**REQUIREMENTS:** Star UML

**DESCRIPTION:**

Component diagram is a different in terms of nature and behavior. Component diagrams are used to model physical aspects of a system. Component diagram are used to visualize the organization and relationship among component of a system.

Component diagram is a special kind of diagram in UML. The purpose is also different from all other diagrams discussed so far. It does not describe the functionality of the system, but it describes the component used to describe that functionality.

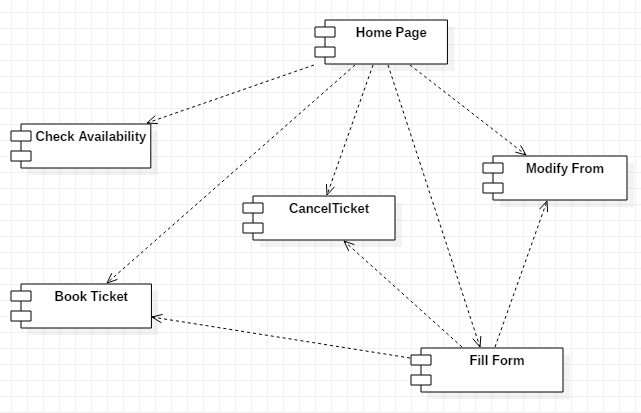
Component diagrams can also be described as the static implementation view of a system. Static implementation represents the organization of the components at a moment.

A single component diagram cannot represent the entire system, but a collection of diagrams is used to represent the whole.

So, the purpose of the component diagram can be summarized as:

* Visualize the components of the system.
* Construct executable by using forward and reverse engineering.
* Describe the organization and relationship of the components.

**DIAGRAM:**



**EXPERIMENT-3(h)**

**AIM: To create component diagram for Library Management System.**

**REQUIREMENTS:** Star UML

**DESCRIPTION:**

Component diagram is a different in terms of nature and behavior. Component diagrams are used to model physical aspects of a system. Component diagram are used to visualize the organization and relationship among component of a system.

Component diagram is a special kind of diagram in UML. The purpose is also different from all other diagrams discussed so far. It does not describe the functionality of the system, but it describes the component used to describe that functionality.

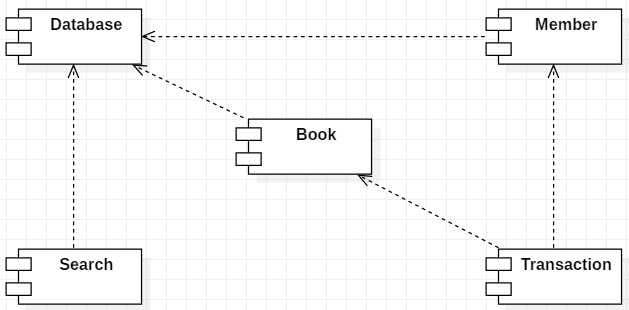
Component diagrams can also be described as the static implementation view of a system. Static implementation represents the organization of the components at a moment.

A single component diagram cannot represent the entire system, but a collection of diagrams is used to represent the whole.

So, the purpose of the component diagram can be summarized as:

* Visualize the components of the system.
* Construct executable by using forward and reverse engineering.
* Describe the organization and relationship of the components.

**DIAGRAM:**



**EXPERIMENT-3(i)**

**AIM: To create deployment diagram for Railway Management System.**

**REQUIREMENTS:** Star UML

**DESCRIPTION:**

Deployment diagrams are used to visualize the topology of the physical components of a system where the software components are deployed. So, the diagrams are used to describe the static deployment view of a system. Deployment diagrams consist of nodes and their relationships.

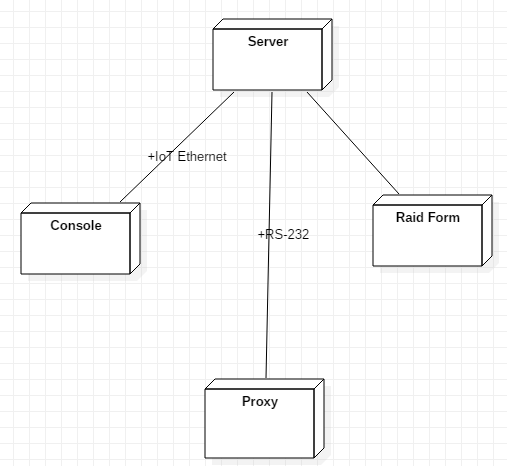
The name Deployment itself describes the purpose of the diagram. Deployment diagrams are used for describing the hardware components where software components are deployed. Component diagrams and deployment diagrams are closely related.

Component diagrams are used to describe the components and deployment diagrams show how they are deployed in hardware. UML is mainly designed to focus on software artifacts of a system. But these two diagrams are special diagrams used to focus on software components and hardware components.

So, most of the UML diagrams are used to handle logical components but deployment diagrams are made to focus on hardware topology of a system. Deployment diagrams are used by the system engineers. The purpose of deployment diagrams can be

* Visualize hardware topology of a system.
* Describe the hardware components used to deploy software components.
* Describe runtime processing nodes.

**DIAGRAM:**



**EXPERIMENT-3(j)**

**AIM: To create deployment diagram for Library Management System.**

**REQUIREMENTS:** Star UML

**DESCRIPTION:**

Deployment diagrams are used to visualize the topology of the physical components of a system where the software components are deployed. So, the diagrams are used to describe the static deployment view of a system. Deployment diagrams consist of nodes and their relationships.

The name Deployment itself describes the purpose of the diagram. Deployment diagrams are used for describing the hardware components where software components are deployed. Component diagrams and deployment diagrams are closely related.

Component diagrams are used to describe the components and deployment diagrams show how they are deployed in hardware. UML is mainly designed to focus on software artifacts of a system. But these two diagrams are special diagrams used to focus on software components and hardware components.

So, most of the UML diagrams are used to handle logical components but deployment diagrams are made to focus on hardware topology of a system. Deployment diagrams are used by the system engineers. The purpose of deployment diagrams can be

* Visualize hardware topology of a system.
* Describe the hardware components used to deploy software components.
* Describe runtime processing nodes.

**DIAGRAM:**

