DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

VISION & MISSION OF THE INSTITUTE

Vision of the Institute

To be a globally recognized institution that fosters innovation, excellence, and leadership in education, research, and technology development, empowering students to create sustainable solutions for the advancement of society.

Mission of the Institute

- To foster a transformative learning environment that empowers students to excel in engineering, innovation, and leadership.
- To produce skilled, ethical, and socially responsible engineers who contribute to sustainable technological advancements and address global challenges.
- To shape future leaders through cutting-edge research, industry collaboration, and community engagement.

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

DEPARTMENT VISION & MISSION

Vision:

To empower the students to be technologically adept, innovative, self-motivated and responsible global citizen possessing human values and contribute significantly towards high quality technical education by harmonizing innovation with sustainability.

Mission:

- To offer high-quality education in the computing fields by providing an environment where the knowledge is gained and applied to participate in research, for both students and faculty.
- To develop the problem solving skills in the students to be ready to deal with cutting edge technologies of the industry.
- To make the students and faculty excel in their professional fields by inculcating the communication skills, leadership skills, team building skills with the organization of various co-curricular and extra-curricular programmers.
- To provide the students with theoretical and applied knowledge, and adopt an education approach that promotes lifelong learning and ethical growth.

MARRI LAXMAN REDDY
INSTITUTE OF TECHNOLOGY AND MANAGEMENT
(AN AUTONOMOUS INSTITUTION)
(Approved by AICTE, New Delhi & Affiliated to JNTUH, Hyderabad)
Accredited by NAAC with 'A' Grade & Recognized Under Section 2(f) & 12(B) of the UGC act,1956

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CERTIFICATE

This is to certify that this manual is a **bonafide record of practical work** carried out in **Software Engineering Lab** for the B.Tech Computer Science and Engineering (CSE) III Semester Programme during the academic year **2026–2027**.

This manual has been prepared by the **Department of Computer Science and Engineering (CSE)** with our own efforts and to the best of our knowledge.

Signature of Lab Faculty Signature of HOD

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

PREFACE

This lab manual entitled **"Software Engineering Lab"** is intended for the use of II B.Tech I Semester Computer Science and Engineering (CSE) students of **Marri Laxman Reddy Institute of Technology and Management, Dundigal, Hyderabad**.

The main objective of the Software Engineering Lab is to provide hands-on experience with intelligent computing techniques such as neural networks, fuzzy logic, and genetic algorithms. Students learn to solve complex, real-world problems that do not have exact mathematical models.

It enhances skills in Artificial Intelligence and Machine Learning through practical implementation and experimentation. The lab prepares learners for advanced research and industry applications in adaptive and intelligent systems.

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

GENERAL INSTRUCTIONS

1. Students are instructed to come to the Software Engineering laboratory on time. Latecomers are not entertained in the lab.
2. Students should be punctual. Experiments will not be repeated for latecomers.
3. Students must come prepared with the experiments to be performed.
4. Students must display their identity cards before entering the lab.
5. Mobile phones are strictly not allowed inside the lab.
6. Any damage or loss of system components (keyboard, mouse, etc.) is the responsibility of the student, and a penalty will be imposed.
7. Students must update their records and observation books session-wise. The observation book should be signed by the faculty before leaving the lab.
8. Lab records must be submitted in the next lab session for correction.
9. Students should not move around unnecessarily during the lab session.
10. In case of emergency, students must obtain written permission from the concerned faculty.
11. Faculty members may suspend any student from the lab session on disciplinary grounds.
12. Students must not copy outputs from others and should write their own results.

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

SAFETY MEASURES

To ensure the safe and efficient use of the Computer Science and Engineering (AI & ML) laboratory, all students must strictly follow the safety guidelines given below:

1. General Conduct

- Maintain silence and discipline during lab sessions.
- Do not bring food, drinks, or chewing gum into the lab.
- Use lab resources responsibly and follow all instructions given by the instructor or lab assistant.

2. Electrical Safety

- Do not touch electrical switches, sockets, or plugs with wet hands.
- Avoid overloading power sockets with unauthorized devices.
- Immediately report any loose connections, sparks, or unusual noises.

3. Computer and Equipment Handling

- Handle computers, keyboards, mouse, and peripherals with care.
- Do not open or tamper with hardware components.
- Use only the assigned system; do not switch without permission.

4. Software and Data Safety

- Use only authorized software installed by the lab administrator.
- Do not install or modify any software without approval.
- Save your work frequently and maintain backups.

5. Cyber Security and Network Usage

- Keep login credentials confidential.
- Do not access restricted websites or servers.
- Avoid activities such as hacking, gaming, or using pirated content.

6. Emergency Preparedness

- Be aware of emergency exits, fire extinguishers, and first aid kits.
- In case of emergencies, remain calm and inform the instructor immediately.
- Follow evacuation procedures properly.

7. Post-Lab Procedures

- Log out and shut down the system properly.
- Keep the workstation clean and organized.
- Return borrowed materials to their place.

8. Hygiene and Cleanliness

- Wash or sanitize hands before and after using systems.
- Do not place unnecessary items on the workstation.
- Report spills or cleanliness issues to lab staff.

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

PROGRAM OUTCOMES (PO'S)

PO NO	NBA Statement / Vital Features		
	Graduate Attributes	**Program Outcomes**	**No. of key competencies**
PO1	Engineering knowledge	Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.	3
PO2	Problem analysis	Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.	10
PO3	Design/development of solutions	Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.	10
PO4	Conduct investigations of complex problems:	Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.	11
PO5	Modern tool usage	Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.	1

PO6	The engineer and society	Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice	5
PO7	Environment and sustainability	Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.	3
PO8	Ethics	Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.	3
PO9	Individual and team work	Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.	12
PO10	Communication	Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.	5
PO11	Project management and finance	Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.	12
PO12	Life-long learning	Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change	8

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

PROGRAM SPECIFIC OUTCOMES(PSO'S)

PO NO	NBA Statement / Vital Features	
	Program Specific Outcomes	**No. of key competencies**
PSO1	Applications of Computing: Ability to use knowledge in various domains to provide solution to new ideas and innovations.	2
PSO2	Programming Skills: Identify required data structures, design suitable algorithms, develop and maintain software for real world problems.	3
PSO3	Make use of computational and experimental knowledge for creating innovative career paths, to be an entrepreneur and desire for higher studies.	2

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

PROGRAM EDUCATIONAL OBJECTIVES (PEOs)

PO NO	NBA Statement / Vital Features
	Program Specific Outcomes
PEO1	**Professional Competence**
	Graduates will possess strong theoretical and practical knowledge in Artificial Intelligence and Machine Learning, enabling them to solve complex real-world problems, pursue higher education, or excel in professional careers.
PEO2	**Innovation and Research Orientation**
	Graduates will engage in innovative practices, cutting-edge research, and contribute to the advancement of AI and ML technologies through collaboration with industry and academia.
PEO3	**Leadership and Lifelong Learning**
	Graduates will exhibit leadership qualities, effective communication, and teamwork skills, and will continuously upgrade their knowledge to adapt to evolving technological landscapes.
PE04	**Entrepreneurial and Community Engagement**
	Graduates will leverage entrepreneurial skills and a sense of civic responsibility to create AI-driven solutions that benefit local and global communities.

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

25X0579: SOFTWARE ENGINEERING LAB

B.Tech II Year I Semester
L/T/P/C: 0/0/2/1

COURSE OBJECTIVES

- To have hands-on experience in developing a software project using various Software Engineering principles and methods in all phases of software development.

COURSE OUTCOMES

- Understand software project problem identification, requirement analysis, and structured documentation practices across development phases.
- Analyze software requirement specifications, design documents, testing artifacts, configuration management plans, and risk management records.
- Apply CASE tools for software design activities, modeling techniques, and structured system representation.
- Develop unit testing and integration testing cases aligned with functional and design requirements.
- Evaluate software quality through white box and black box testing techniques within structured software projects.

LIST OF EXPERIMENTS

Perform the following exercises for any two projects from the sample list or any other project:

1. Development of problem statements
2. Preparation of Software Requirement Specification (SRS), Design, and Testing documents
3. Preparation of Software Configuration Management and Risk Management documents
4. Study and usage of any Design Phase CASE tool
5. Performing system design using CASE tools
6. Develop test cases for unit testing and integration testing
7. Develop test cases for white box and black box testing techniques

SAMPLE PROJECTS

1. Passport Automation System
2. Book Bank
3. Online Exam Registration
4. Stock Maintenance System
5. Online Course Reservation System
6. E-Ticketing System
7. Software Personnel Management System
8. Credit Card Processing System
9. E-Book Management System
10. Recruitment System

TEXTBOOKS

1. *Software Engineering: A Practitioner's Approach* – Roger S. Pressman, 6th Edition, McGraw Hill
2. *Software Engineering* – Ian Sommerville, 7th Edition, Pearson Education
3. *The Unified Modeling Language User Guide* – Grady Booch, James Rumbaugh, Ivar Jacobson

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

VIRTUAL LAB DETAILS

- **Name of the Virtual Lab:** Software Engineering
- **Host Institute:** IIT Kharagpur
- **URL:** https://se-iitkgp.vlabs.ac.in
- **Academic Year:** 2026–2027
- **Semester:** III

List of Virtual Lab Experiments

1. Identifying requirements from problem statements
2. Estimation of project metrics
3. Modeling UML use case diagrams and capturing scenarios
4. E-R modeling from problem statements

MARRI LAXMAN REDDY
INSTITUTE OF TECHNOLOGY AND MANAGEMENT
(AN AUTONOMOUS INSTITUTION)
(Approved by AICTE, New Delhi & Affiliated to JNTUH, Hyderabad)
Accredited by NAAC with 'A' Grade & Recognized Under Section 2(f) & 12(B) of the UGC act,1956

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

LAB PLANNER

S.No	Experiment	CO	Virtual Lab	Date Planned	Date Conducted
1	Development of problem statements	CO1	NA		
2	SRS Preparation	CO1	NA		
3	Design & Testing Documents	CO2	NA		
4	SCM & Risk Management	CO2	NA		
5	CASE Tool Study	CO3	NA		
	CIE – I				
6	Design using CASE Tools	CO3	NA		
7	Unit & Integration Testing	CO4	NA		
8	White & Black Box Testing	CO4	NA		
	CIE – II				

MARRI LAXMAN REDDY
INSTITUTE OF TECHNOLOGY AND MANAGEMENT
(AN AUTONOMOUS INSTITUTION)
(Approved by AICTE, New Delhi & Affiliated to JNTUH, Hyderabad)
Accredited by NAAC with 'A' Grade & Recognized Under Section 2(f) & 12(B) of the UGC act,1956

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

SOFTWARE ENGINEERING LAB

LAB PLANNER

Dateplaned																															
Dateconducted																															
RollNumber	Exp No	CO	VL*	Exp No	CO	VL*	Exp No	CO	VL*	Exp No	CO	VL*	Exp No	CO	VL*		Exp No	CO	VL*	Exp No	CO	VL*	Exp No	CO	VL*	Exp No	CO	VL*	Exp No	CO	VL*	

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

SOFTWARE ENGINEERING LABORATORY

RUBRICS USED TO ASSESS LEARNING IN LABORATORIES

1. RUBRICS FOR DAY-TO-DAY EVALUATION

Parameter	MaxMark	Level-1 (VeryPoor)	Level-2 (Poor)	Level-3 (Average)	Level-4 (Good)	Level-5 (Excellent)
Observation Book	05	No observationsor irrelevant data.(0–1)	Incompleteor incorrectdata. (2)	Basicvalueswith some errors. (3)	Mostlycorrect withgood format.(4)	Fully correct, clear,andwell-formatted.(5)
RecordWriting	05	Not submitted.(0-1)	Submitted butmostly incomplete.(2)	Submitted withsome missing/wrongparts. (3)	Submittedwith minor issues. (4)	Fully complete, correct algorithm& flowchart.(5)
Result	05	Noresultormajor errors.(0–1)	Resultpartially obtained.(2)	Acceptable resultwith limitederror.(3)	Near- correct resultand reasonableerror.(4)	Accurateresult. (5)
Viva-Voce	05	Didnot answerany questions.(1)	Answered veryfewquestions.(2)	Answered some questionswithhelp.(3)	Answeredmost questions correctly.(4)	Answeredall questions accurately.(5)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

SOFTWARE ENGINEERING LABORATORY

2. RUBRICS FOR INTERNAL EVALUATION

Criterion	Max Marks	Level-1 *(VeryPoor)*	Level-2 *(Poor)*	Level-3 *(Average)*	Level-4 *(Good)*	Level-5 *(Excellent)*
Design/Tool/Apparat us Selection	2Marks	Incorrect tool/designandno reasoning. **(0)**	Tool/design Selection attemptedwith unclearlogic. **(0.5)**	Satisfactory Selectionwith partial justification. **(1)**	Correctselection andproper analysiswithfew errors.**(1.5)**	Smartselection with accurate, relevantanalysis. **(2)**
Execution (Code/Debug/Run) /Analysis/Method Used	4 Marks	Didnotattempt or completelyfailed to execute. **(0)**	Attemptedbut unableto proceed orwithmajor errors. **(1)**	Partial execution withsome logic/syntax errors. **(2)**	Mostlycorrect executionwith minimal help.**(3)**	Fully correct and Independentlye xecuted program. **(4)**
Results&Documentation	2 Marks	Incomplete orpoorly presented. **(0)**	Basic structure butlacks clarityor formatting. **(0.5)**	Completebut genericor with formattin g issues. **(1)**	Well-structure dand mostlyclear. **(1.5)**	Well-organized, professional ,and engaging documentatio n.**(2)**
Viva-Voce (Understandin gofConcepts)	2 Marks	Nounderstanding;could notanswerquestions. **(0)**	Answeredafew with difficulty .**(0.5)**	Answeredhalfthequestion swithbasic clarity. **(1)**	Good understandingwithconfident answers. **(1.5)**	Answered allquestions withclarityand depth. **(2)**

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

SOFTWARE ENGINEERING LABORATORY

3. RUBRICS FOR SEMESTER END EXAMINATIONS

Criterion	Max Marks	Level-1 (VeryPoor (*0–2marks)*	Level-2(Poor)*(3–4 marks)*	Level-3 (Average)*(5–6 marks)*	Level-4 (Good)*(7–9 marks)*	Level-5 (Excellent)*(10–12marks)*
Preparednessfor the Experiment	12 marks	Noclarityon objective o procedure. Unableto explain basics.	Limited ideaof the objective/procedure. Needed prompting.	Has basic understanding; minor gaps in concept or preparation.	Well-prepared, with clear understandingofstepsand background.	Fully preparedwith strong conceptual clarity and confident explanation.
Performanceinthe Laboratory	12 marks	Unable to perform experiment. Relied entirely on examiner'shelp.	Performed with multiple errors and constant support.	Performed with some errors; required occasional help.	Performed mostly independently with minimal support.	Performed independently,efficiently, and with precision.
Calculations & Graphs	12 marks	No or incorrect calculations. Graphs missing or irrelevant.	Multiple calculationerrors. Graphs/plots inaccurate or poorly labeled.	Calculations partially correct. Graphspresent butwithsome flaws.	Correct calculations and graphs with minor errors.	Accurate calculations and well-labeled graphswith proper interpretation.
Results& Error Analysis	12 marks	Noresultor invalidresult. No error analysis attempted.	Incorrect result with vague or no error discussion.	Acceptable result. Error analysis attempted but limited.	Correct resultwith sounderror discussion.	Accurate result with detailedand relevant error analysis.
Viva-Voce (Subject Knowledge)	12 marks	Unable to answer any questions. No conceptual understanding.	Answered few questions with poor logic.	Answeredhalf of the questionswith average understanding.	Answered most questions with clarity and confidence.	Answeredall questions with depth, clarity,and reasoning.

EXPERIMENT – 1

AIM

Development of problem statements.

INTRODUCTION

The Passport Automation System is designed to streamline and automate the process of passport application, verification, and issuance. It replaces the traditional manual system with a digital platform to improve efficiency and transparency.

EXISTING SYSTEM

In the existing system, passport applications are processed manually. Applicants must visit passport offices, fill out physical forms, submit documents, and wait for verification and approval.

PROCEDURE

PROBLEM STATEMENT

Passport Automation System (PAS) is designed to automate the process of passport application, verification, and dispatch. The system minimizes manual work and optimizes time and resource utilization.

The applicant fills an online registration form with personal details such as name, address, and identification information. These details are verified by the system using the existing database.

After initial verification:

- Data is forwarded to the regional administrator (Ministry of External Affairs)
- Manual verification is performed
- Appointment scheduling is provided
- Police verification is initiated
- Final approval leads to passport dispatch

The applicant can track the application status online at every stage.

(II) SOFTWARE REQUIREMENT SPECIFICATION (SRS)

1. INTRODUCTION

PAS acts as an interface between the applicant and passport issuing authority, improving efficiency and reducing complexity.

2. PURPOSE

Manual passport processing is time-consuming. This automated system:

- Reduces processing time
- Handles increasing applications
- Ensures security and validation

3. SCOPE

- Online application submission
- Document upload (scanned copies)
- Appointment scheduling
- Status tracking
- Communication between applicant and administrator
- Police verification integration

4. DEFINITIONS, ACRONYMS, ABBREVIATIONS

- **Administrator**: Central authority managing the system
- **Applicant**: Person applying for passport
- **PAS**: Passport Automation System
- **HTML**: Web page markup language
- **J2EE**: Java enterprise platform
- **HTTP**: Communication protocol
- **TCP/IP**: Network communication protocol

5. TECHNOLOGIES USED

- HTML
- JSP
- JavaScript
- Java

TOOLS USED

- Eclipse IDE
- Rational Rose (UML Tool)

6. OVERVIEW

SRS contains:

- Overall Description
- Specific Requirements

7. OVERALL DESCRIPTION

Product Perspective

PAS connects applicant and administrator through a secure interface.

Software Interface

- Front End: JSP, HTML
- Server: Apache Tomcat
- Back End: Oracle 11g

Hardware Interface

- Client systems connected to server
- Database stored centrally

8. SYSTEM FUNCTIONS

- Applicant registration
- Appointment scheduling
- Status tracking
- Notifications (SMS/Email)
- Report generation
- Final approval and database storage

9. USER CHARACTERISTICS

- **Applicant**: Submits application
- **Administrator**: Verifies and approves
- **Police**: Conducts background verification

10. CONSTRAINTS

- Requires computer access
- Security risks exist
- Accurate data entry required

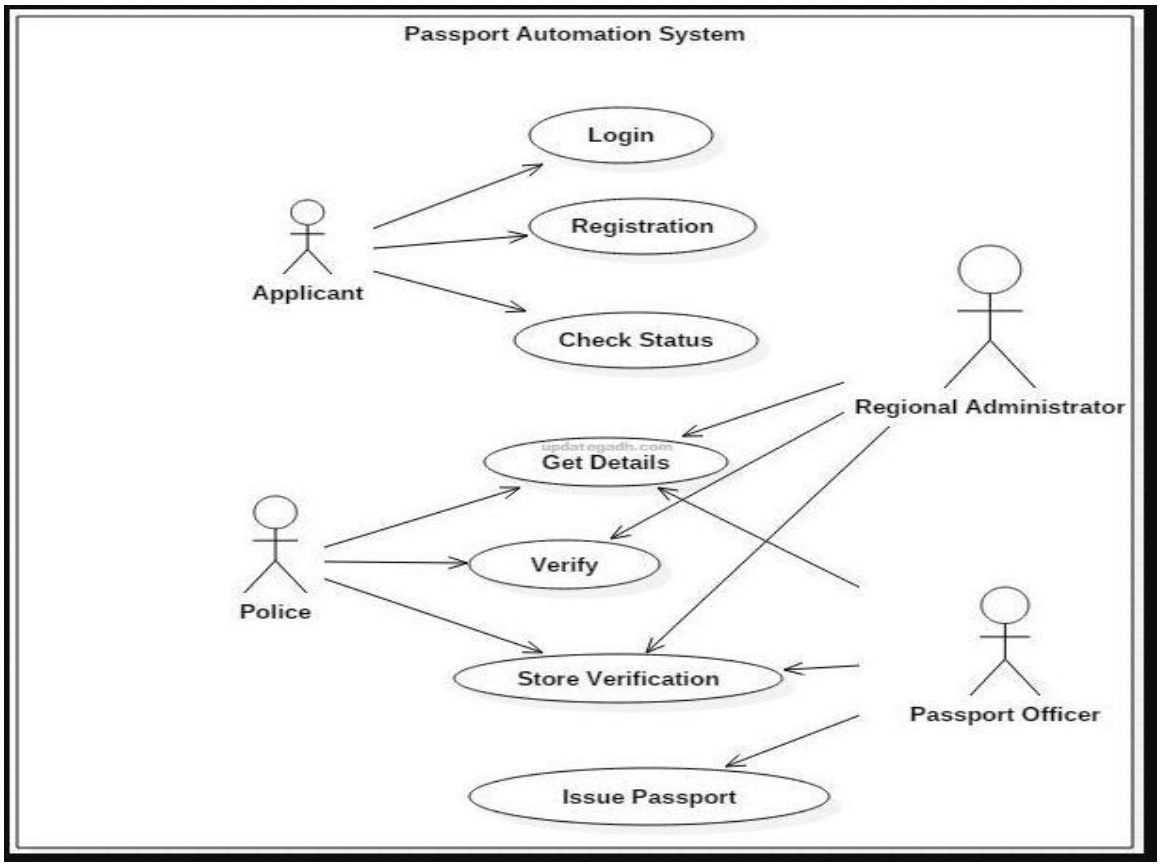
11. ASSUMPTIONS

- Basic computer knowledge required
- Documents must be scanned

(III) UML DIAGRAMS

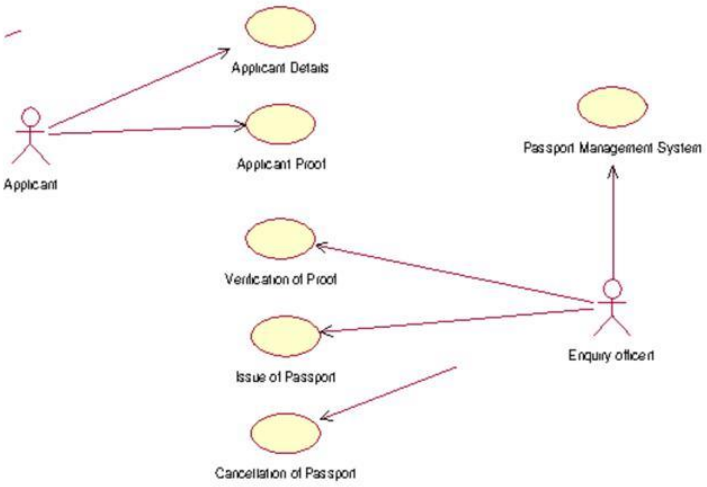
---

1. USE CASE DIAGRAM

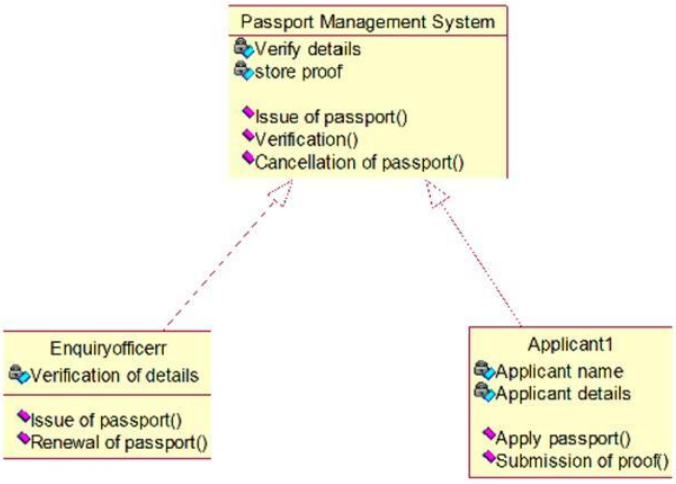


1. PASSPORT AUTOMATION SYSTEM

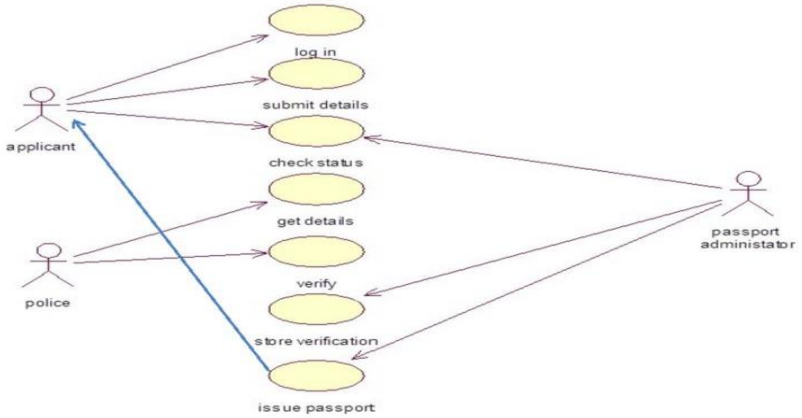
USE CASE DIAGRAM:



CLASS DIAGRAM:



USE CASE DIAGRAM:



Actors

- Applicant
- Passport Officer
- Police

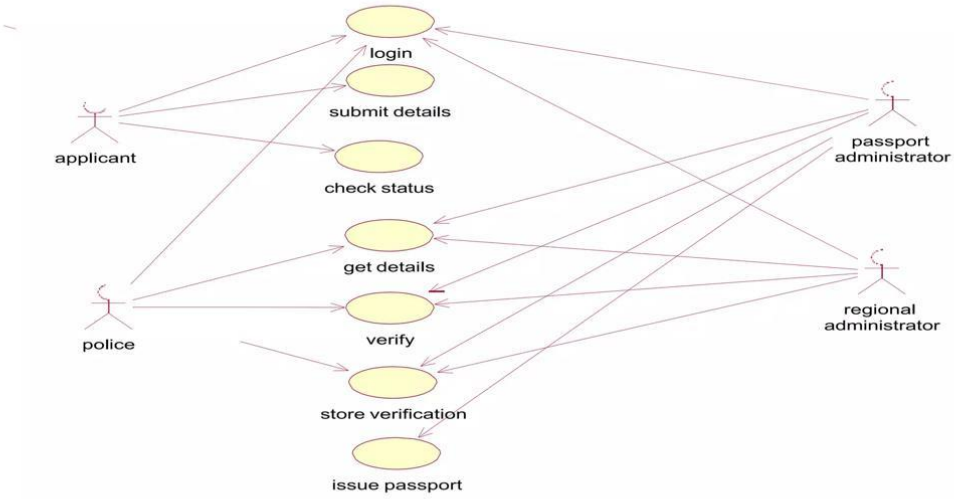
Use Cases

- Login
- Registration
- Verification
- Check Status
- Enquiry
- Dispatch Passport

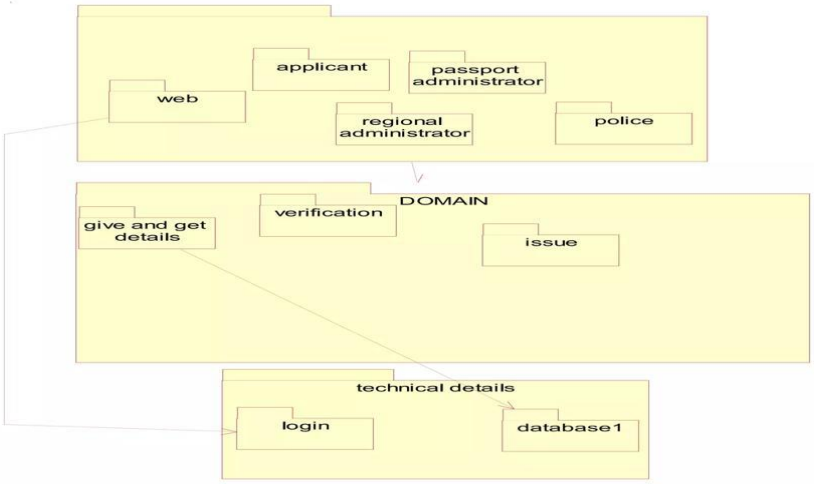
---

2. ACTIVITY DIAGRAM

Example use case diagram for passport automation system



Example package diagram for passport automation system



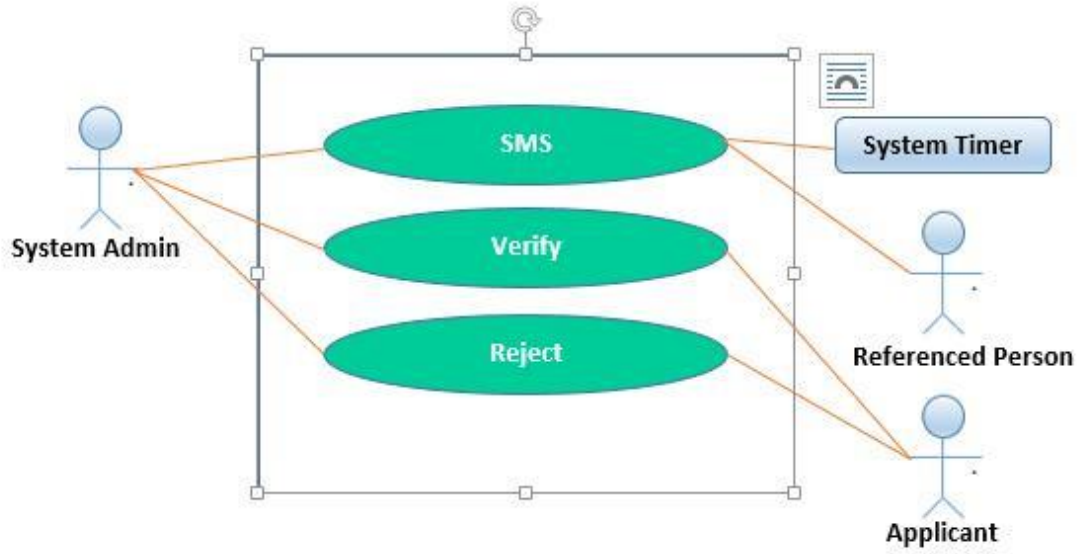
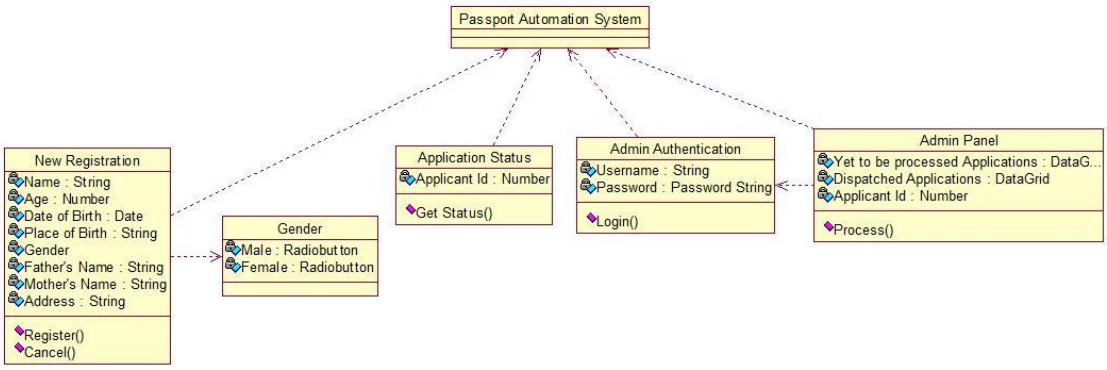


Fig 4: Use Case Diagram for Verification (Level 2.2)

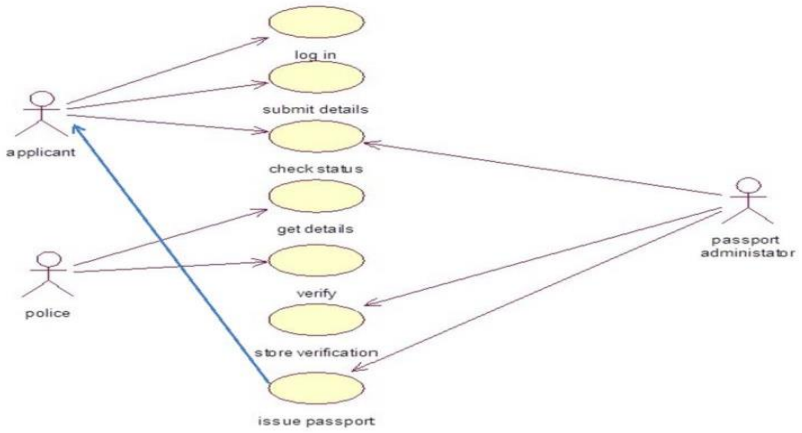
Description

Represents workflow from application submission to passport dispatch.

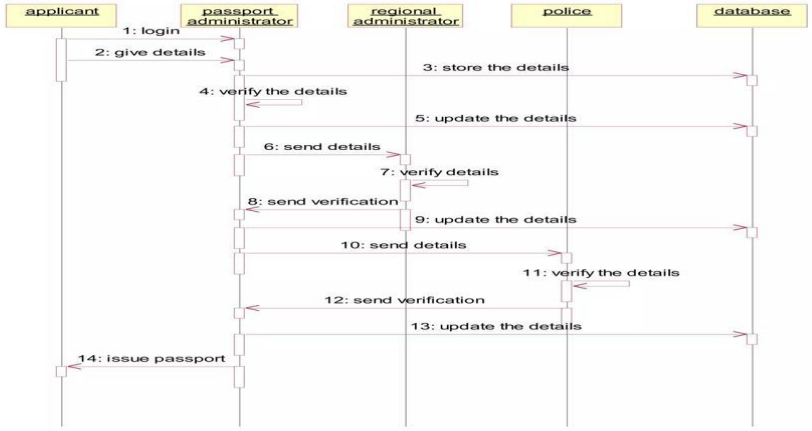
3. CLASS DIAGRAM



USE CASE DIAGRAM:



Example sequence diagram for passport automation system

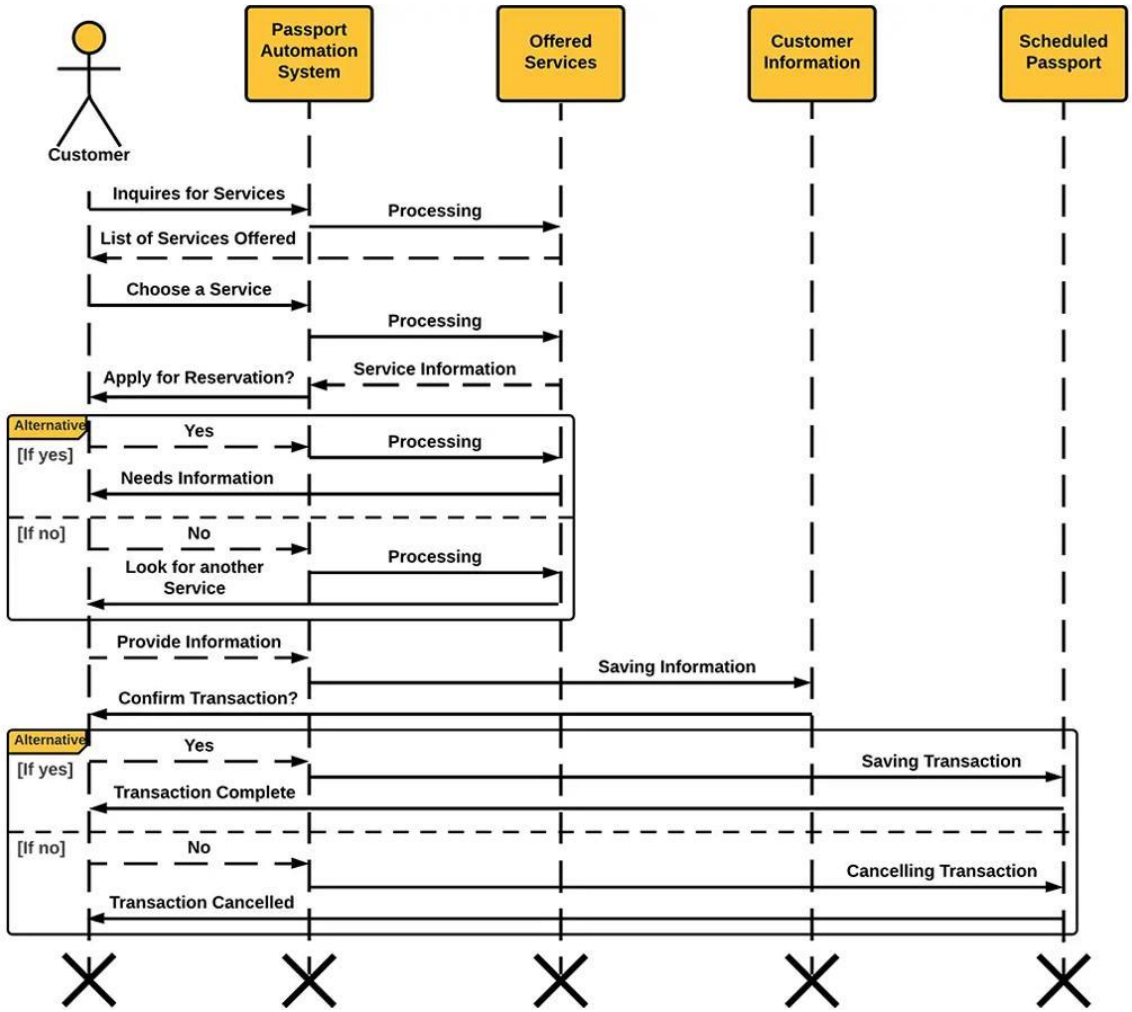


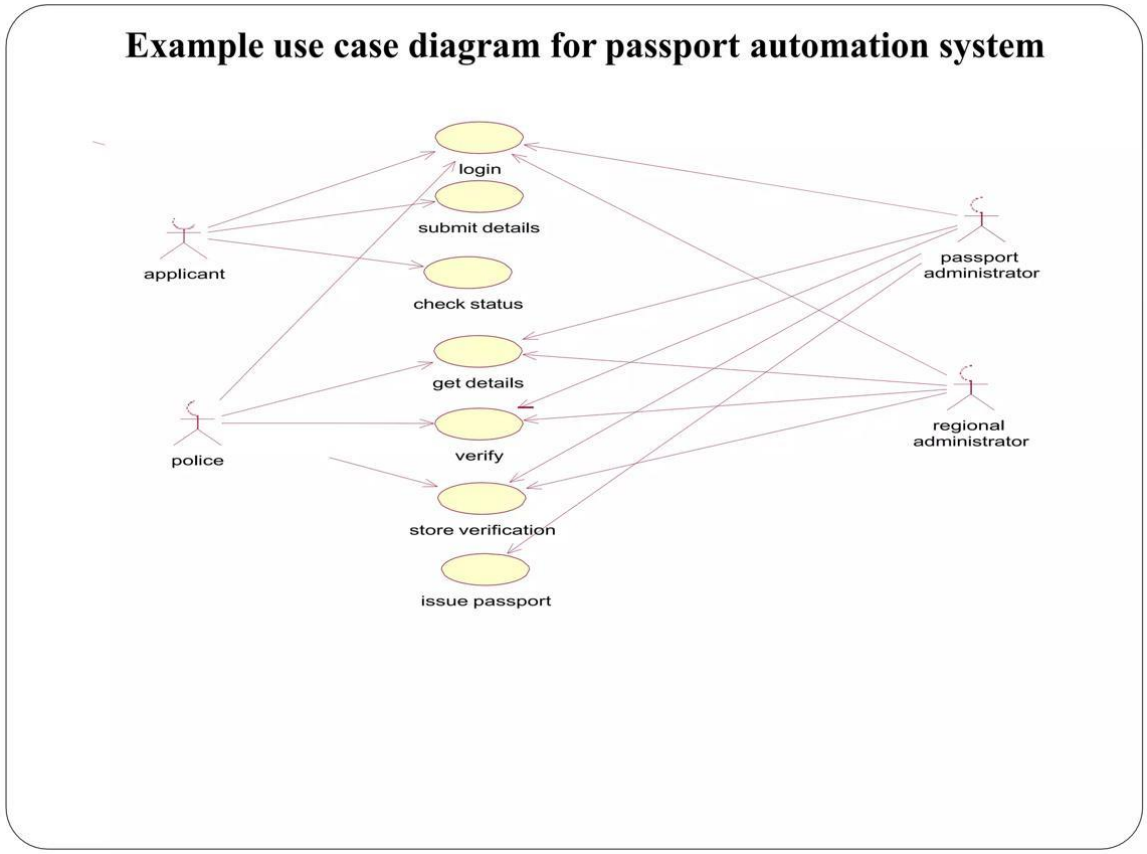
Classes

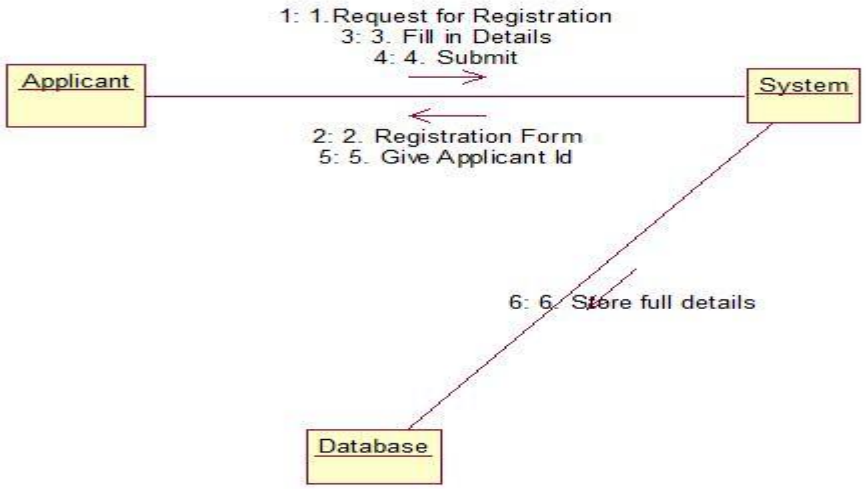
1. Login
2. Appointment
3. Registration
4. Authority
5. Verification

---

4. SEQUENCE DIAGRAM







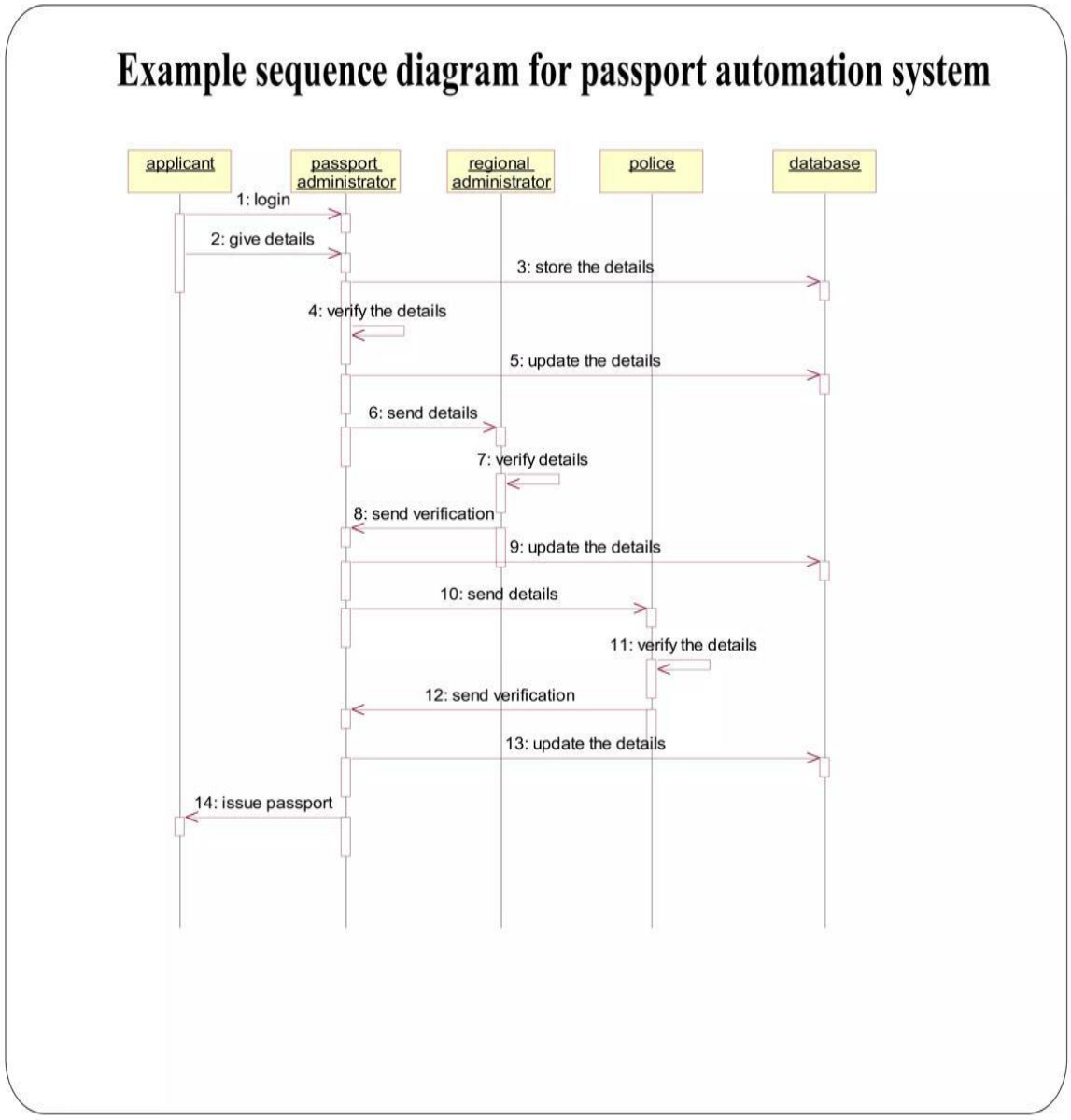
Description

Shows step-by-step interaction between:

- Applicant

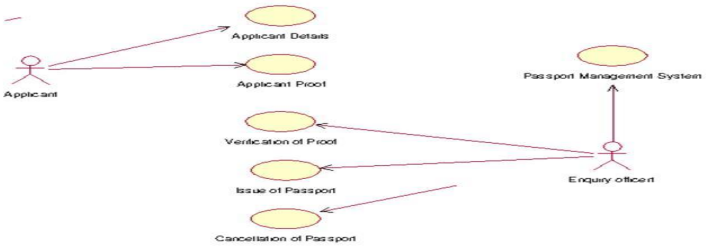
- System
- Database

5. COLLABORATION DIAGRAM



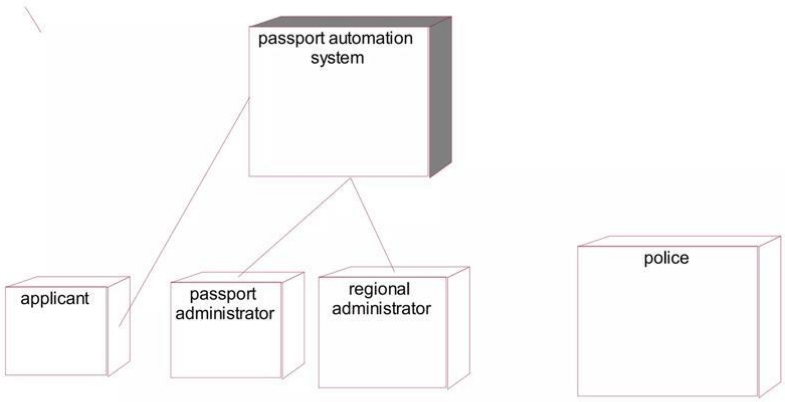
1. PASSPORT AUTOMATION SYSTEM

USE CASE DIAGRAM:



CLASS DIAGRAM:

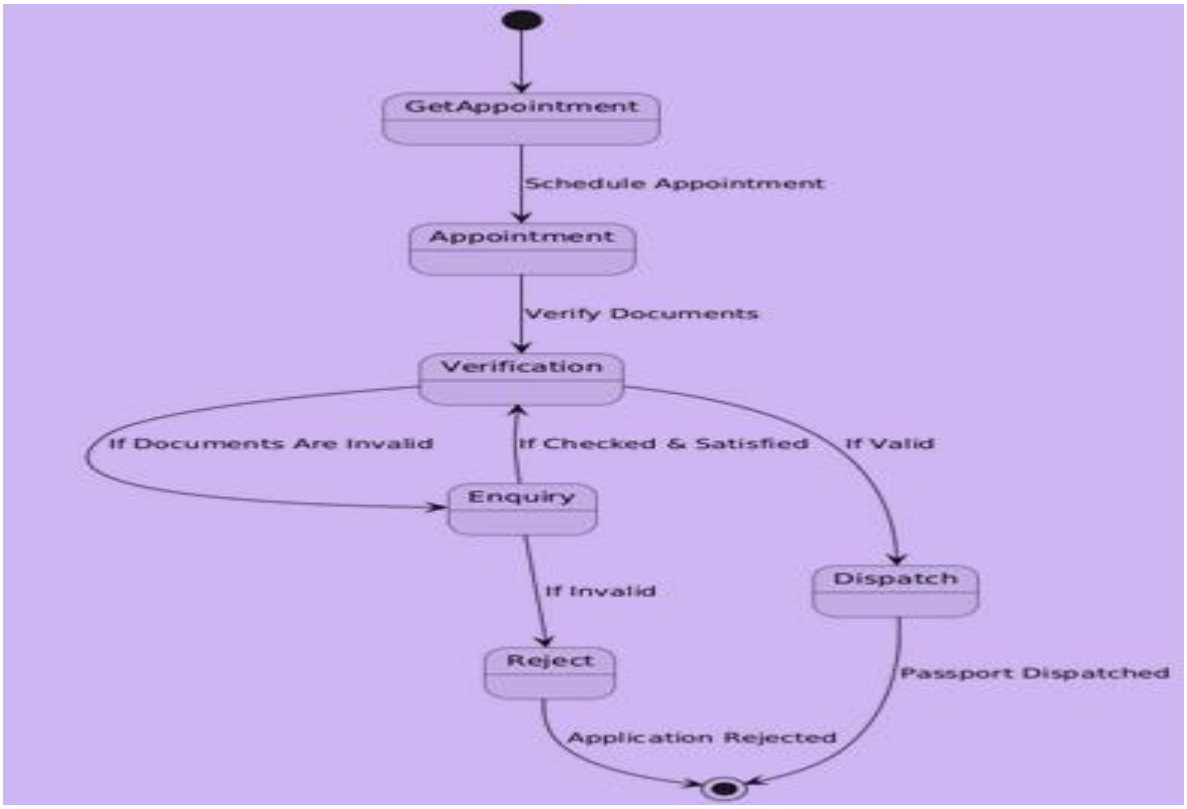
Example deployment diagram for passport automation system



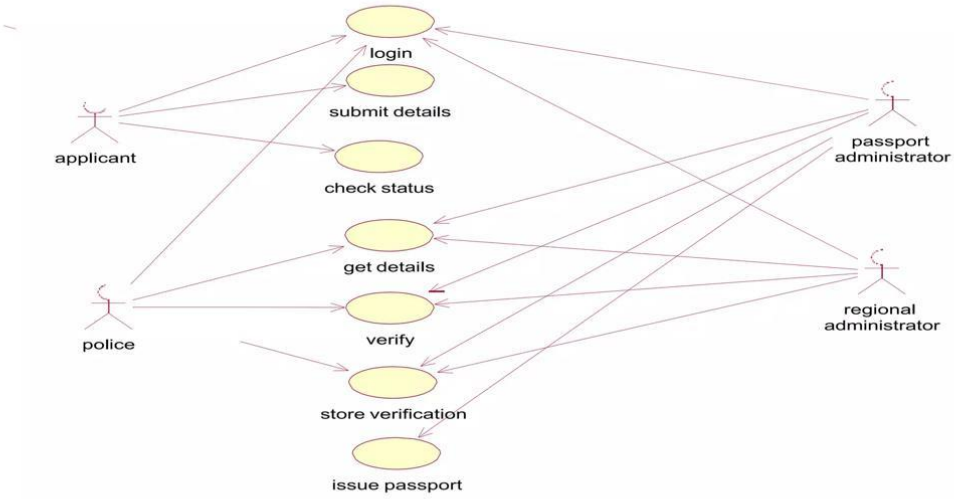
Description

- Shows interaction between objects
- Applicant submits details
- Authority verifies details
- System approves or rejects

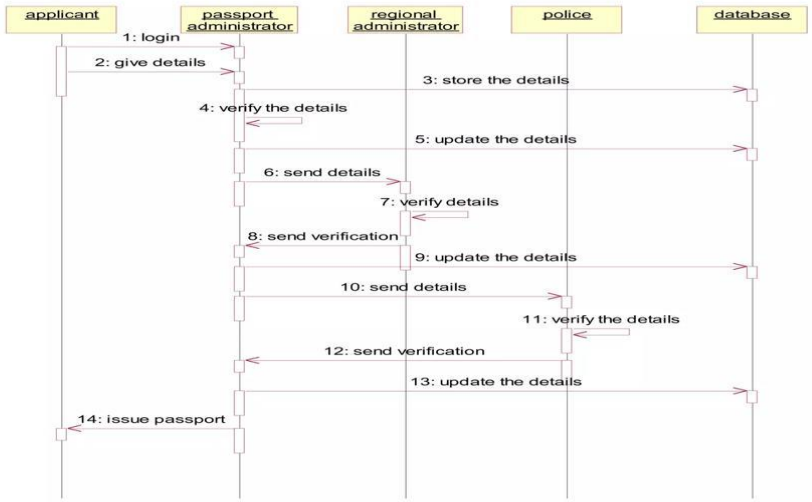
6. STATECHART DIAGRAM



Example use case diagram for passport automation system



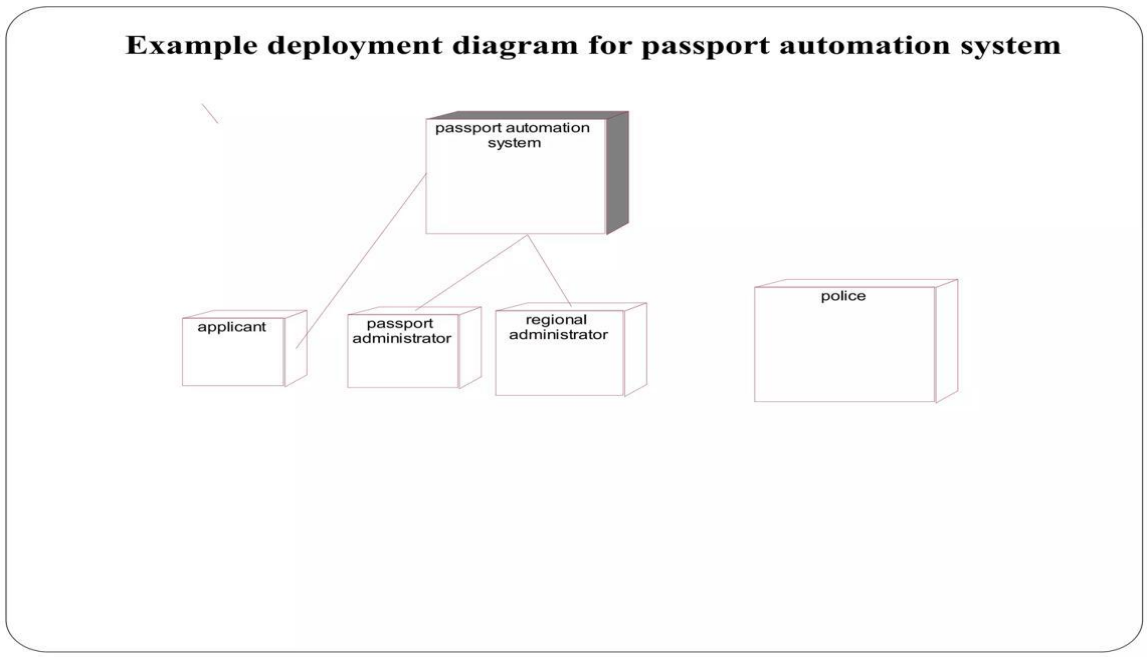
Example sequence diagram for passport automation system

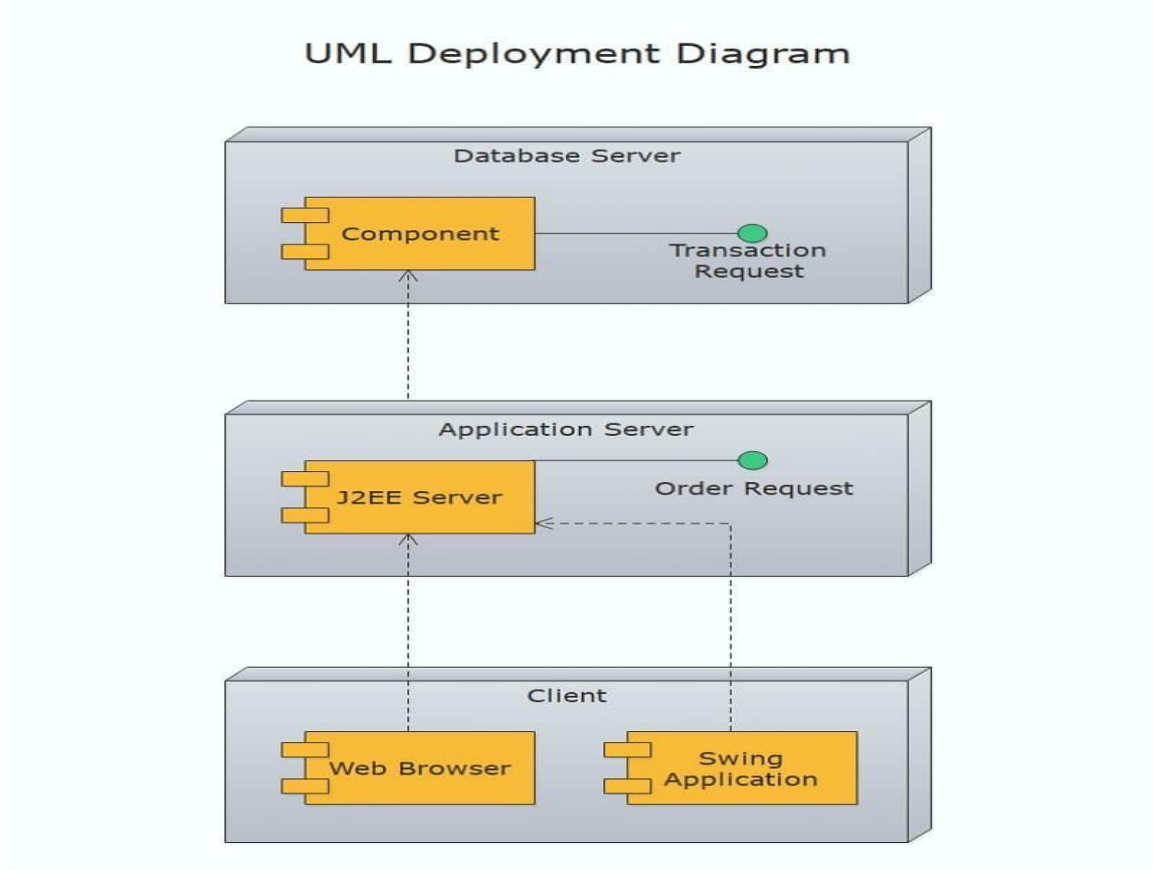


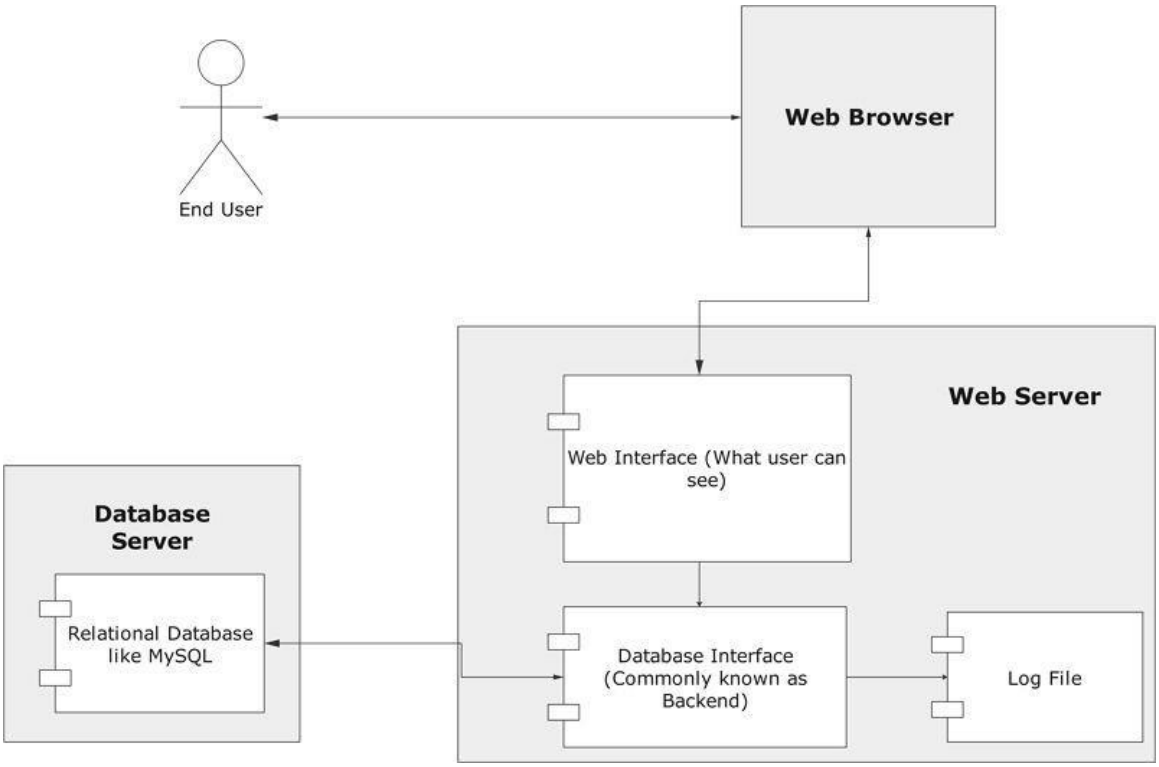
States

- Initial
- Data Entry
- Verification
- Approved
- Rejected
- Dispatch

7. DEPLOYMENT DIAGRAM




UML Deployment Diagram
Database Server
Component
Transaction Request
Application Server
J2EE Server
Order Request
Client
Web Browser
Swing Application

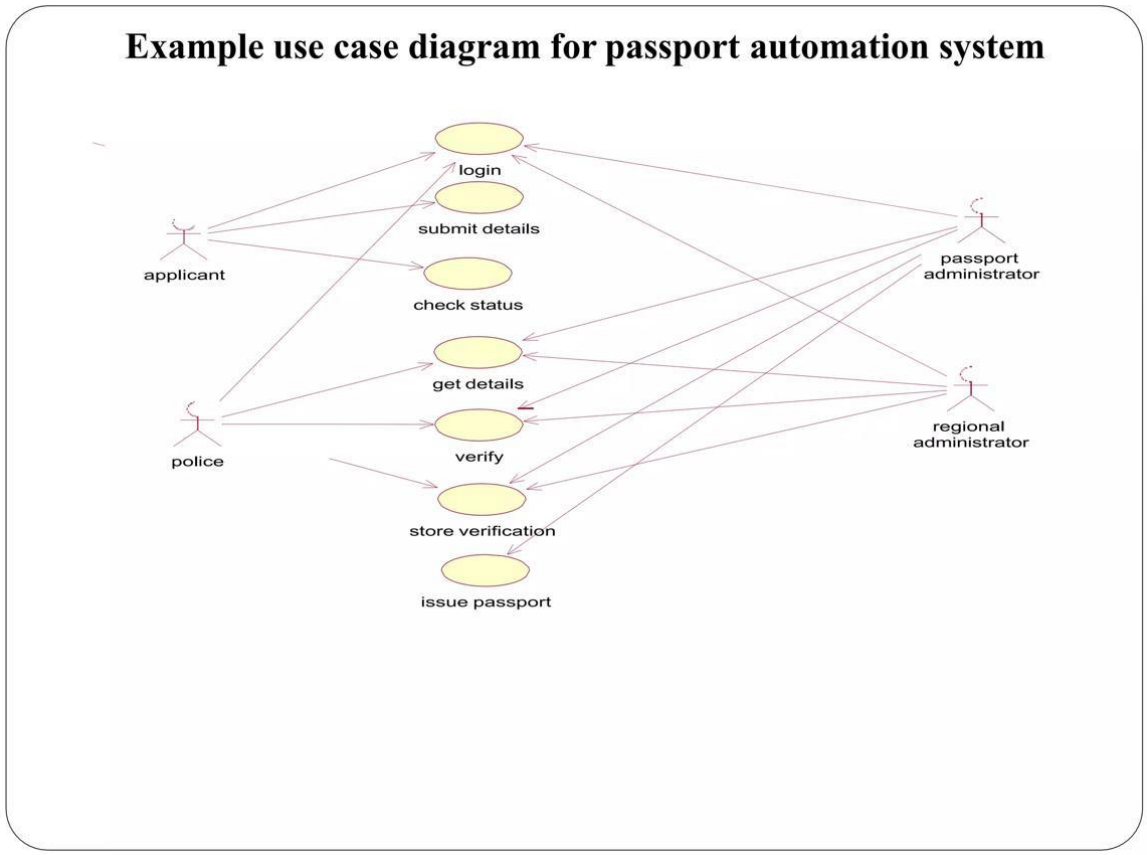

End User
Web Browser
Web Server
Web Interface (What user can see)
Database Server
Relational Database like MySQL
Database Interface (Commonly known as Backend)
Log File

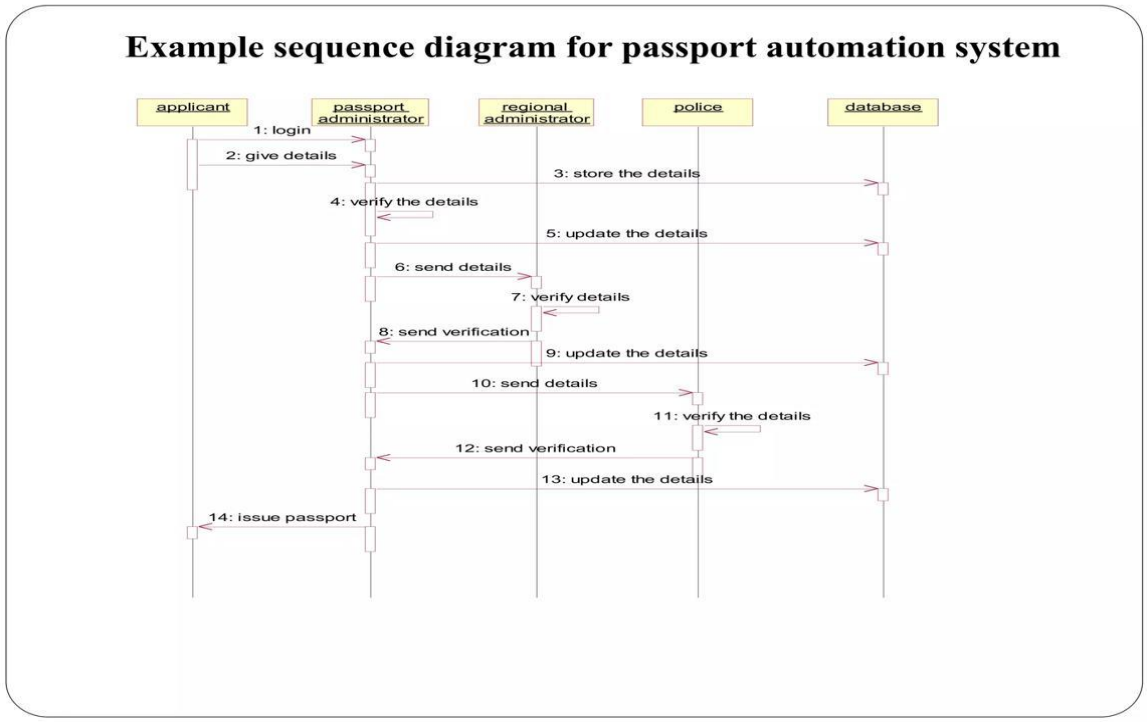
Description

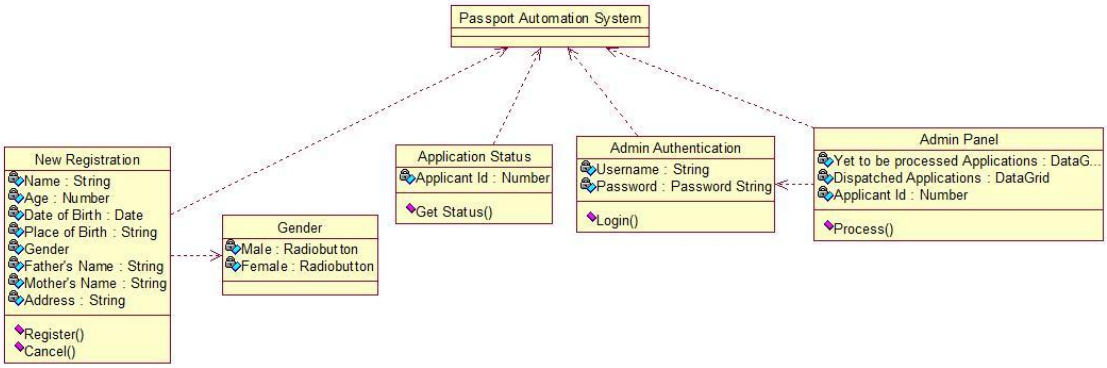
Shows physical architecture:

- Client systems
- Server
- Database

8. COMPONENT DIAGRAM







Description

Represents system modules:

- Login Module
- Registration Module
- Verification Module
- Database Module

EXPERIMENT –2: BOOK BANK SYSTEM

AIM

Preparation of Software Requirement Specification Document, Design Documents and Testing Phase related documents.

PROCEDURE

(I) PROBLEM STATEMENT

A Book Bank system is used to manage the lending of books and magazines to registered members (students). It also handles:

- Purchase of new books
- Maintenance of multiple copies of popular books
- Removal of outdated or damaged books
- Reservation of unavailable books
- Notification to users when reserved books become available

The system allows easy creation, updating, and deletion of records related to:

- Books
- Members
- Loans
- Reservations

(II) SOFTWARE REQUIREMENT SPECIFICATION (SRS)

1. INTRODUCTION

The Book Bank System acts as an interface between Students and the Librarian, improving efficiency and reducing manual work.

2. PURPOSE

Manual book issuing is time-consuming. This system:

- Speeds up issuing and returning
- Handles large numbers of students
- Maintains accurate records

3. SCOPE

- Online student registration
- Book search and reservation
- Issue and return management
- Librarian control panel
- Report generation

4. DEFINITIONS & ACRONYMS

- Librarian: System administrator
- Student: User who borrows books
- HTML: Web page language
- J2EE: Java platform
- HTTP: Communication protocol
- TCP/IP: Network protocol

5. TECHNOLOGIES USED

- Visual Basic
- Oracle 11g

TOOLS USED

- Visual Studio
- Rational Rose (UML Tool)

6. OVERVIEW

The SRS consists of:

- Overall Description
- Specific Requirements

7. OVERALL DESCRIPTION

Product Perspective

The system provides a simple and secure interface between students and the librarian.

Software Interface

- Front End: Visual Studio
- Back End: Oracle 11g

Hardware Interface

- Client systems connected to server
- Centralized database

8. SYSTEM FUNCTIONS

- Student registration
- Book issue and return
- Reservation system
- Book entry by librarian
- Report generation

9. USER CHARACTERISTICS

- Student: Requests and borrows books
- Librarian: Manages system and approvals

10. CONSTRAINTS

- Requires computer access
- Security risks exist
- Accurate data entry required

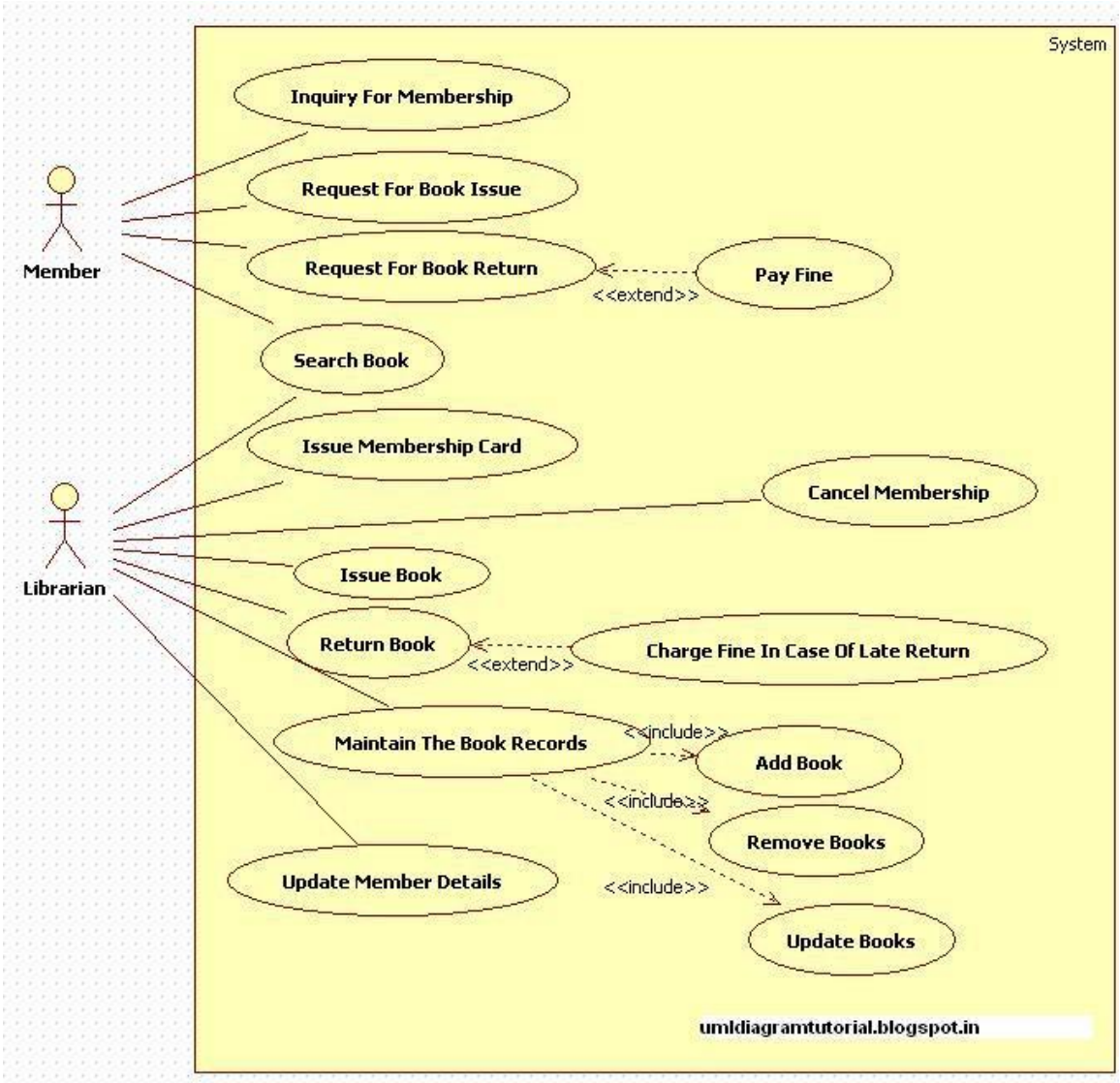
11. ASSUMPTIONS

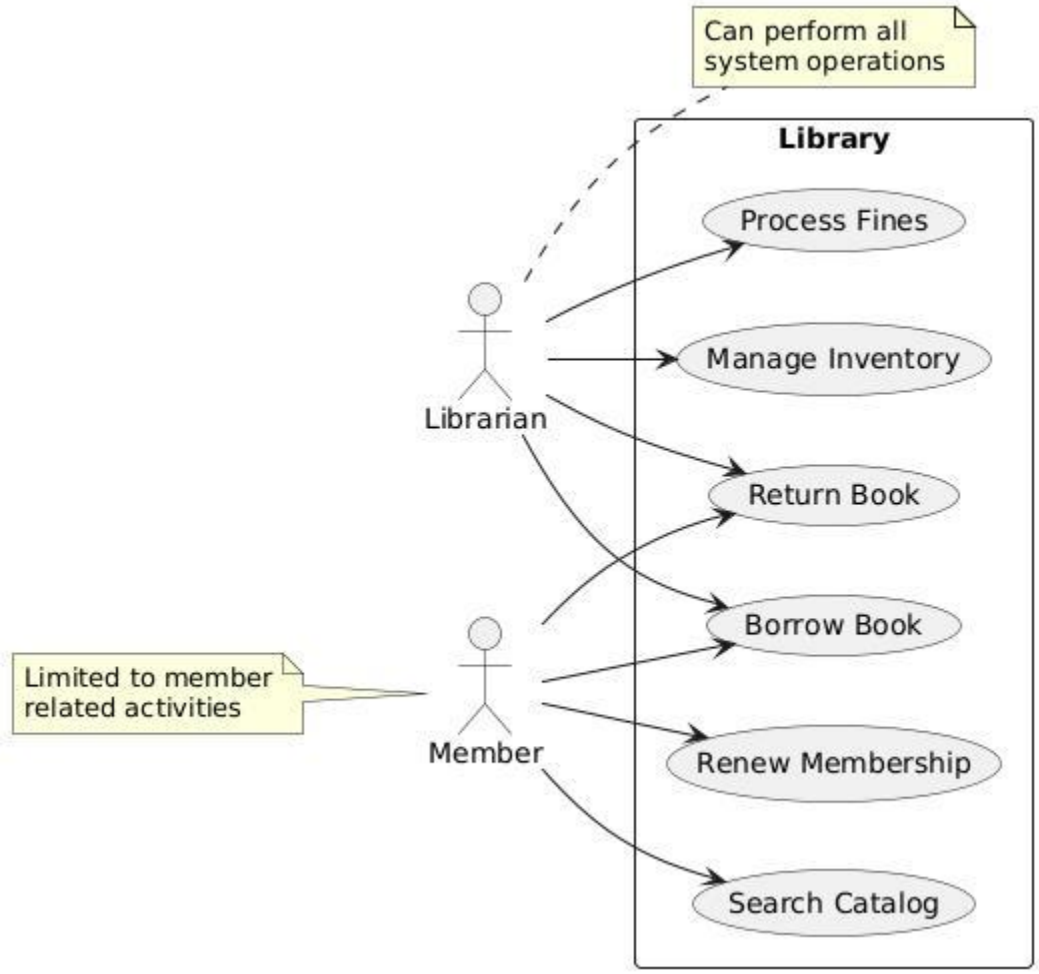
- Basic computer knowledge required

- Documents may need scanning

(III) UML DIAGRAMS

1. USE CASE DIAGRAM





Actors

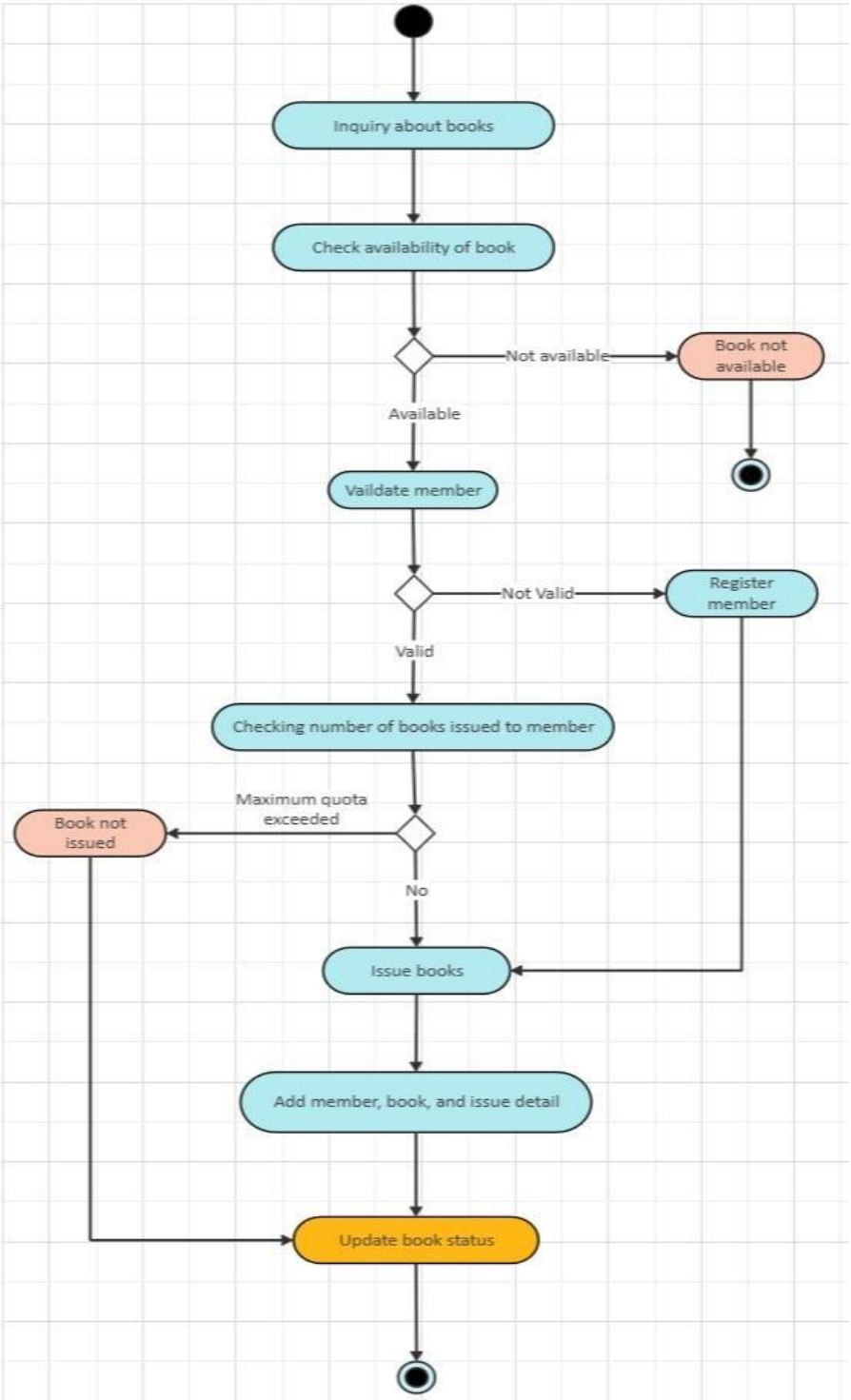
- Student
- Librarian
- Vendor

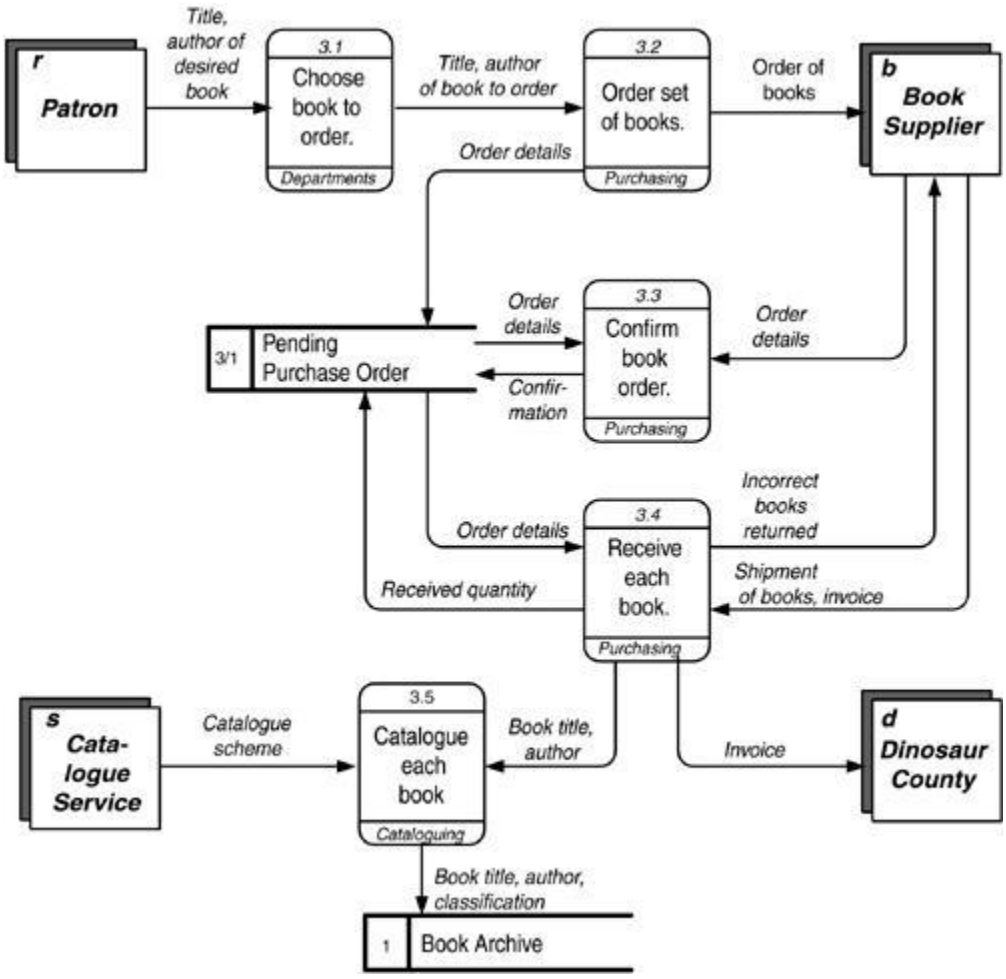
Use Cases

- Book Issue
- Book Return
- Book Order
- Book Entry
- Search Book Details

2. ACTIVITY DIAGRAM

(a) Order Book Process

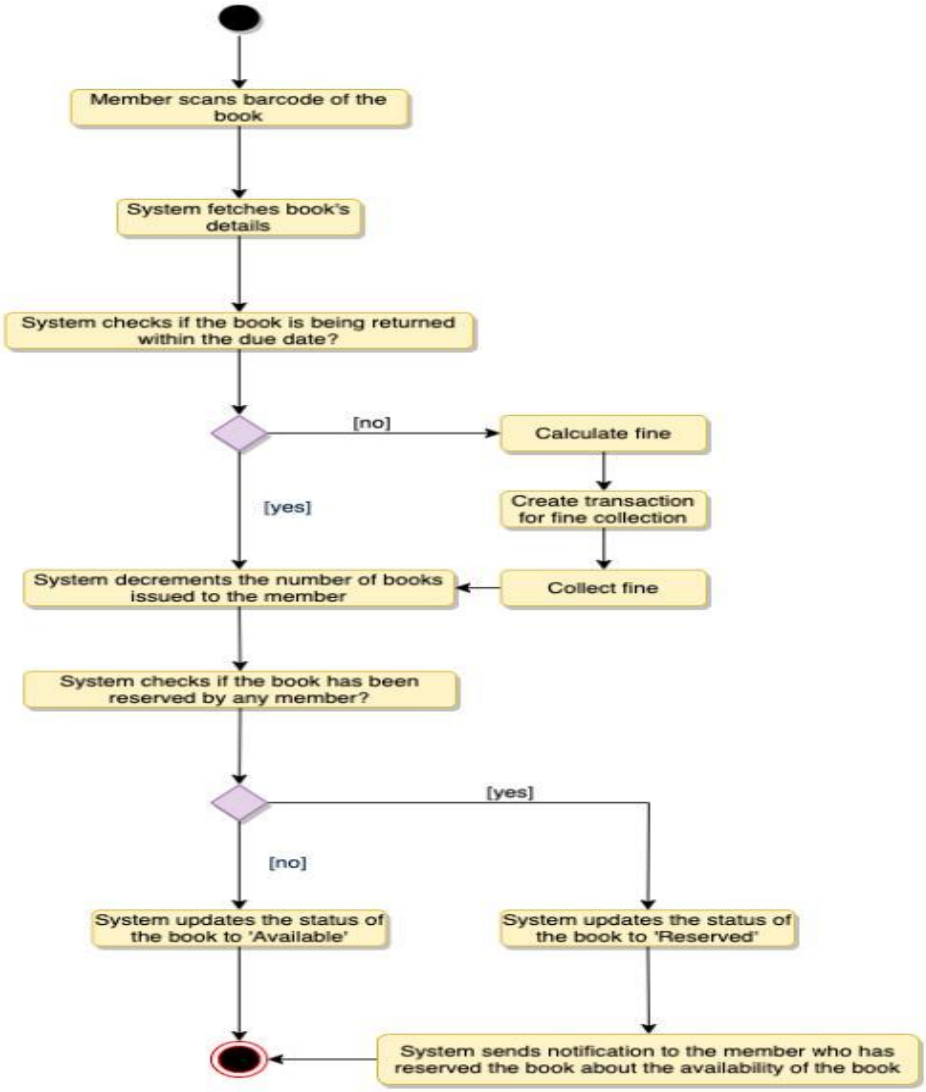




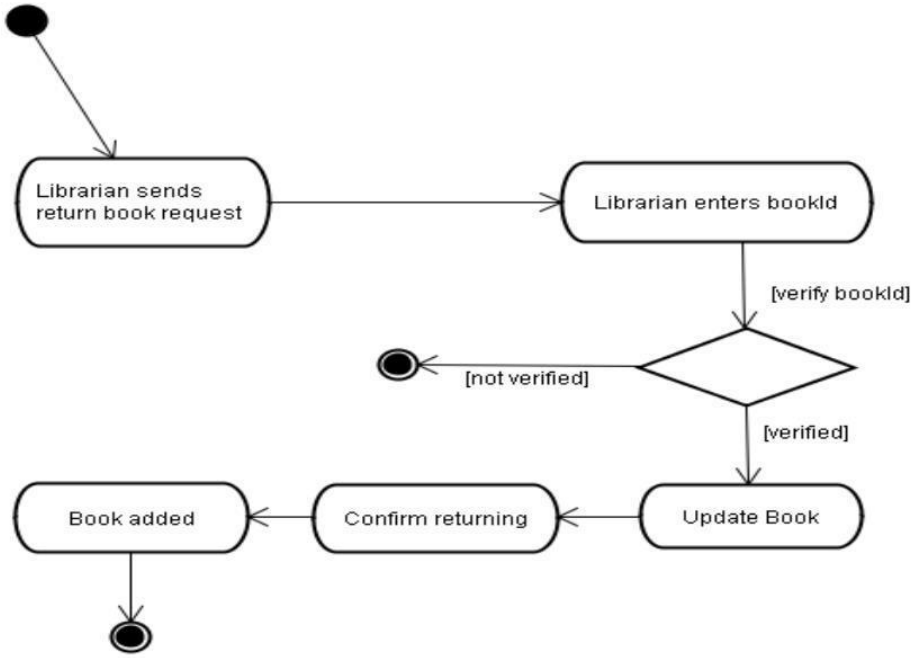
Flow

- Collect quotations
- Check conditions
- Place order
- Receive books
- Make payment

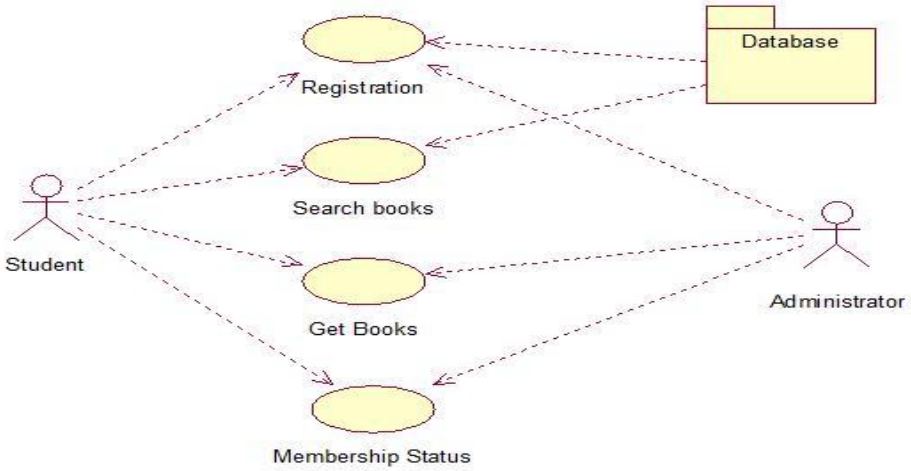
(b) Return Book Process

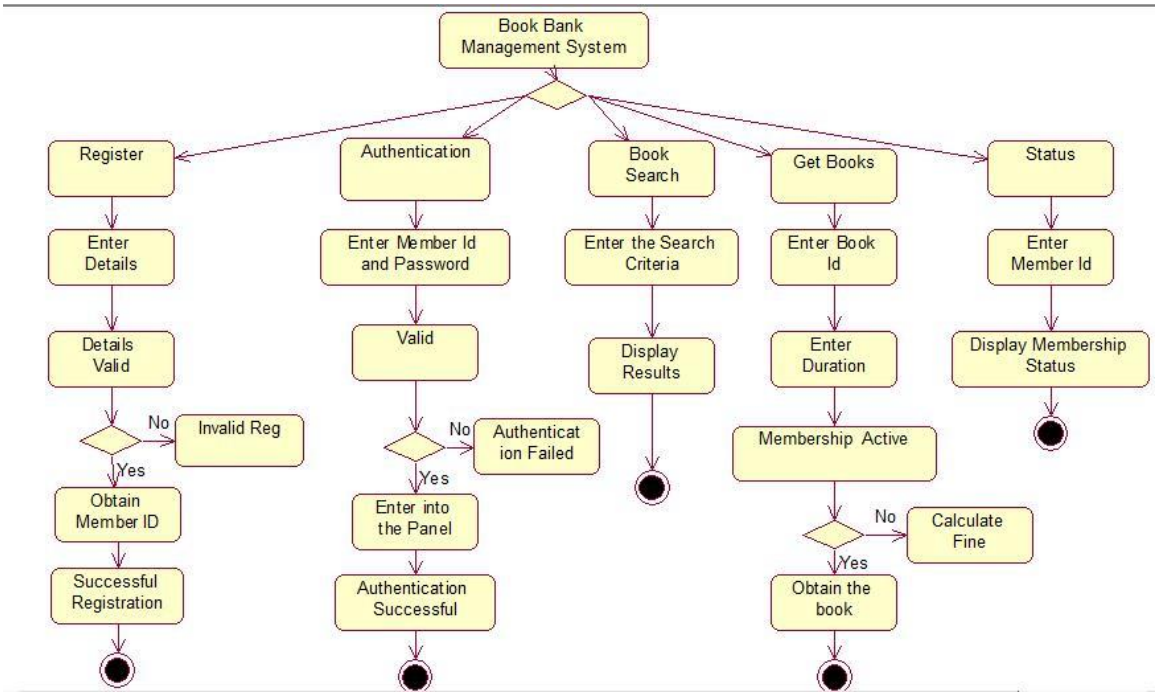


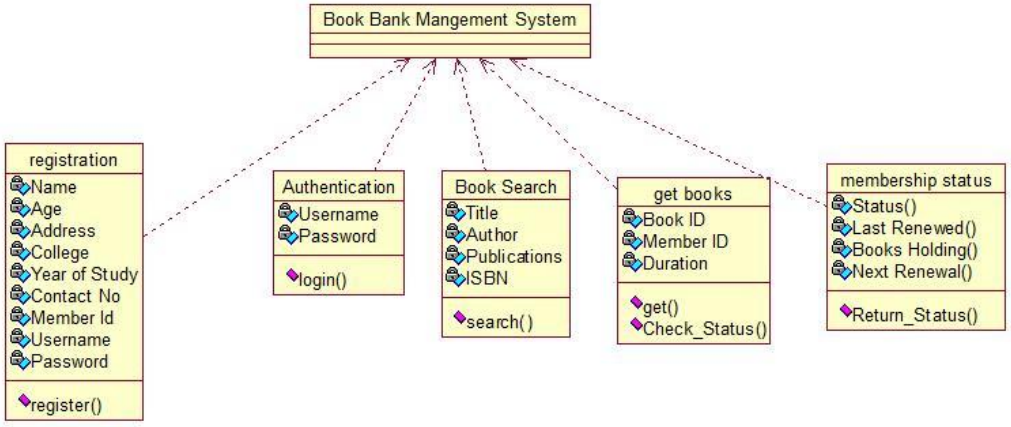
Activity Diagrams – Return Book



3. CLASS DIAGRAM



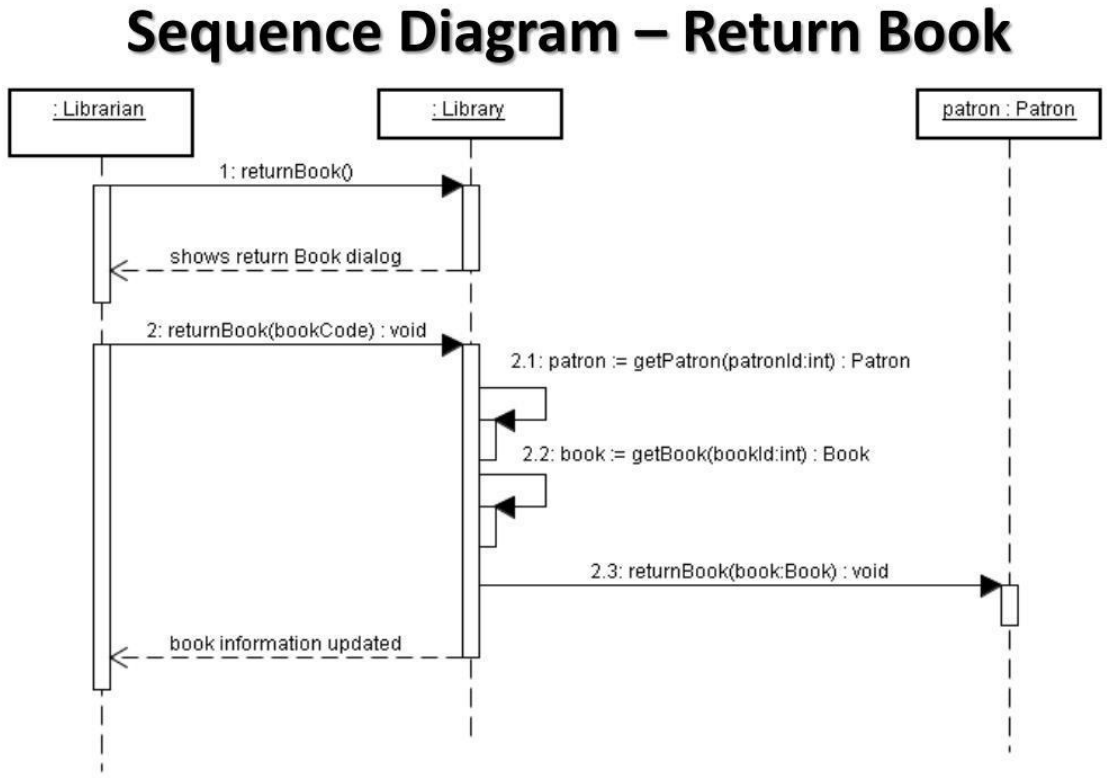


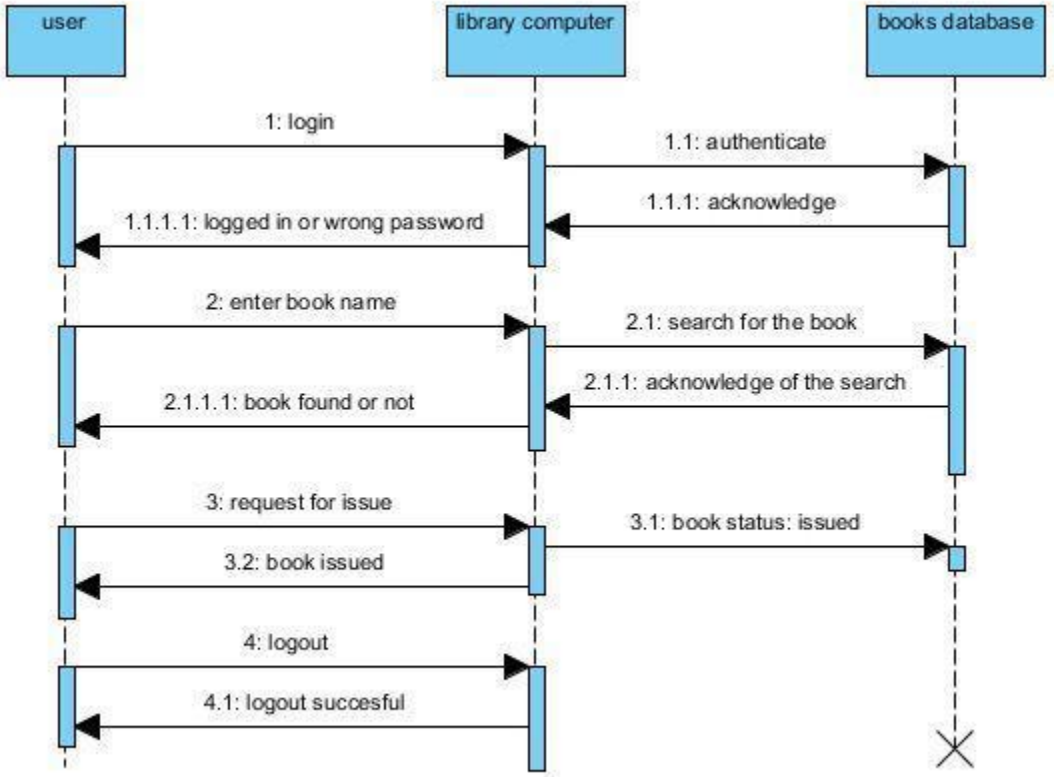


Classes

1. Student
2. Book
3. Issue
4. Return
5. Vendor
6. Details

4. SEQUENCE DIAGRAM



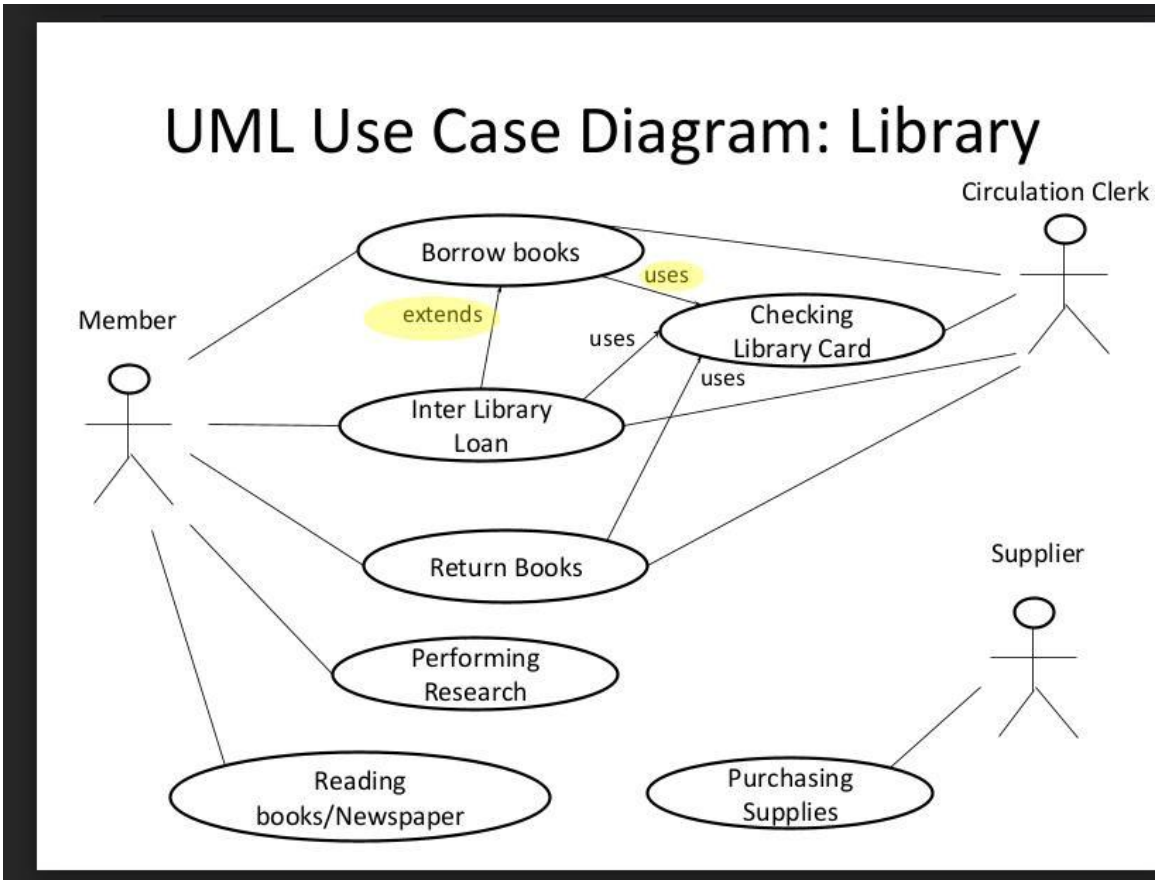


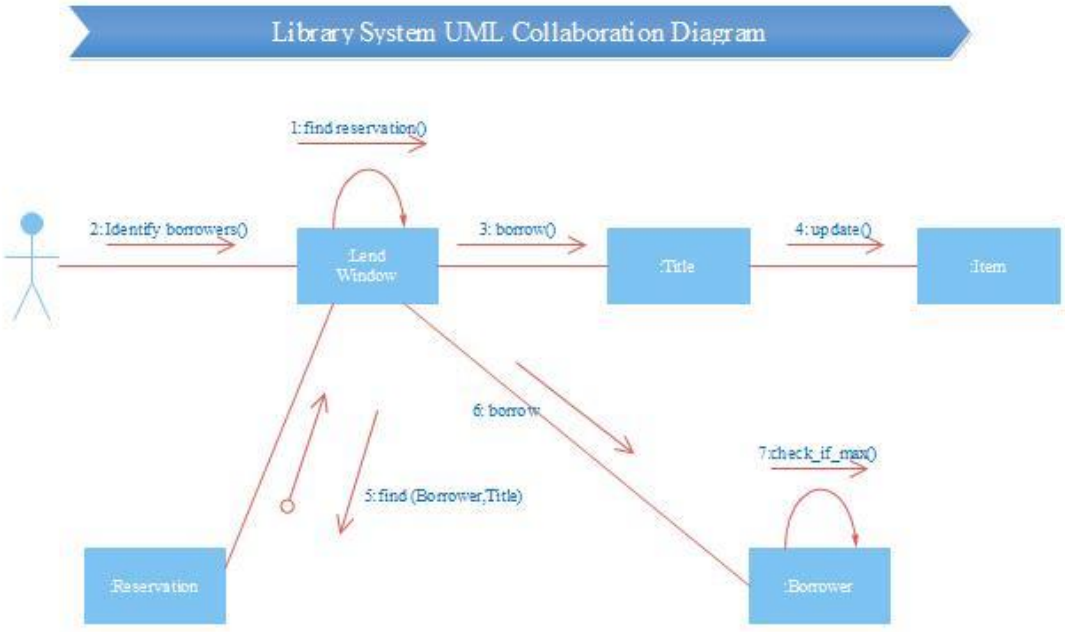
Description

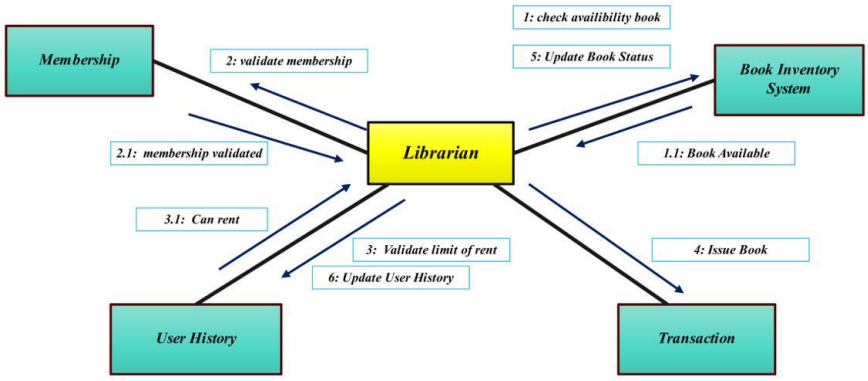
Shows interaction between:

- Student
- Librarian
- Database

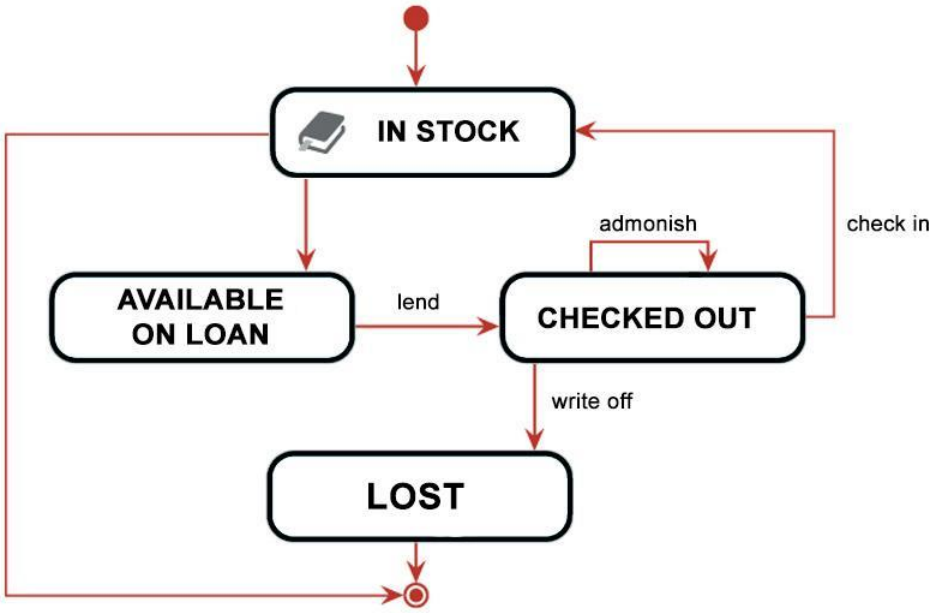
5. COLLABORATION DIAGRAM



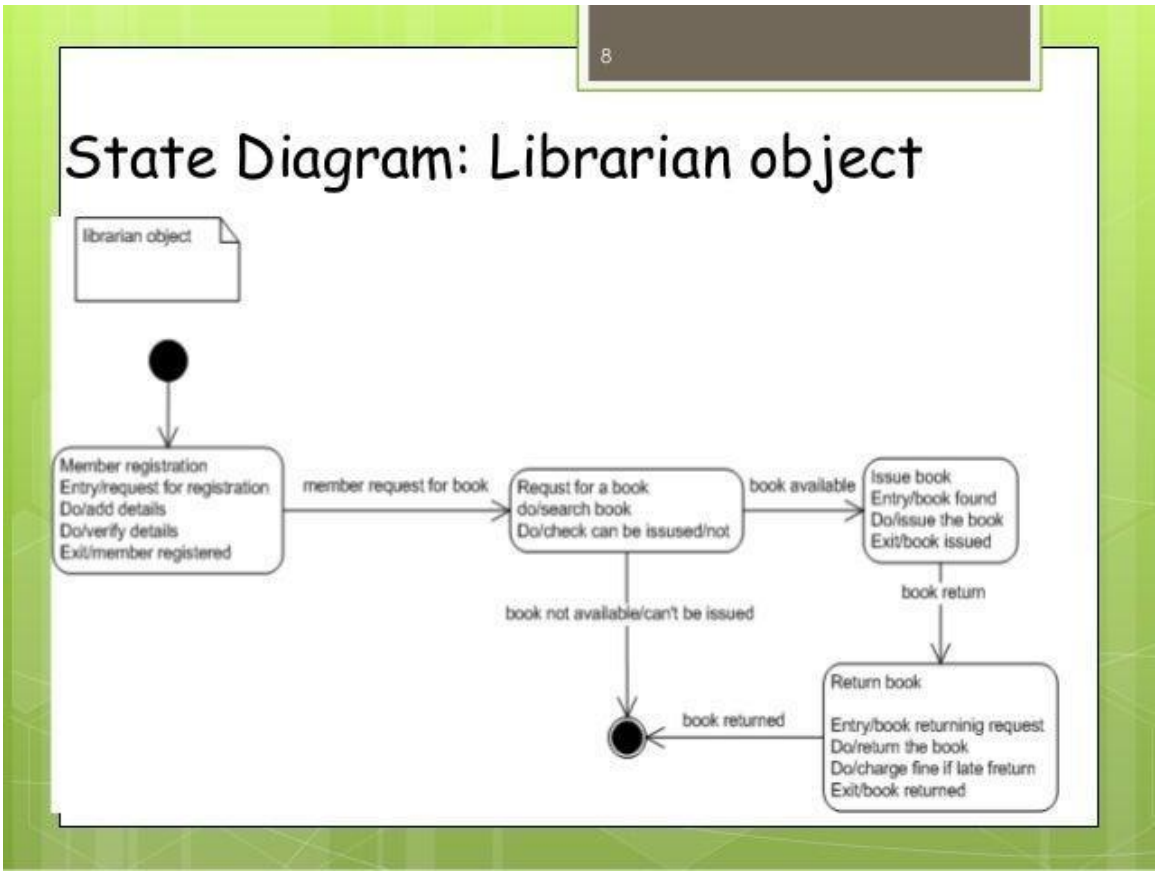




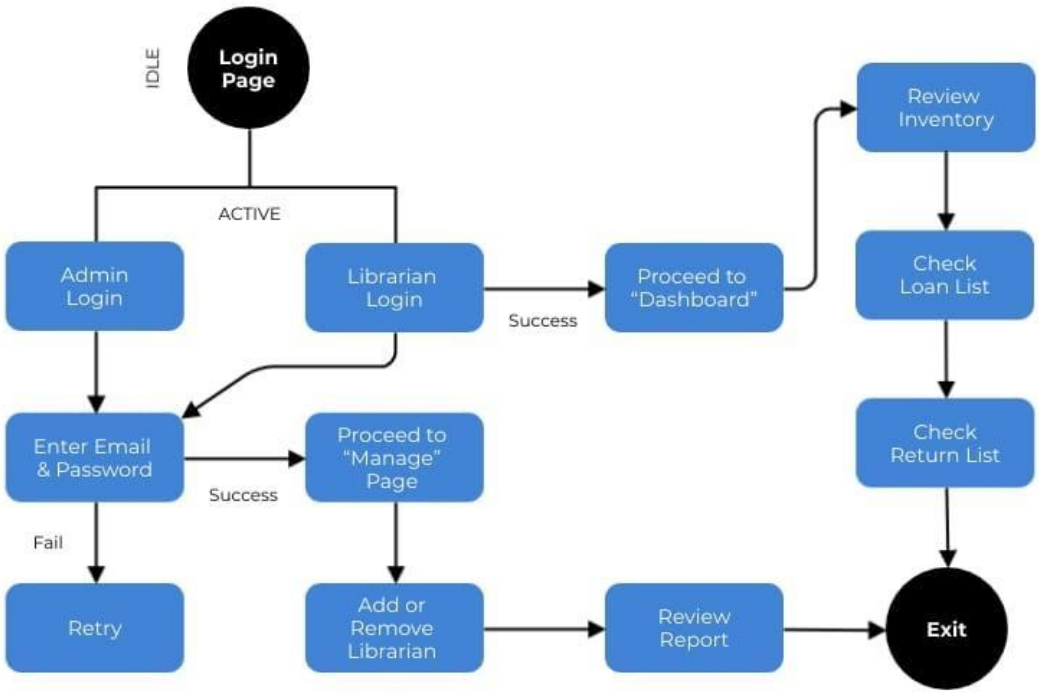
6. STATECHART DIAGRAM



STATE DIAGRAM


8
State Diagram: Librarian object
librarian object
Member registration
Entry/request for registration
Do/add details
Do/verify details
Exit/member registered
member request for book
Requst for a book
do/search book
Do/check can be issused/not
book available
Issue book
Entry/book found
Do/issue the book
Exit/book issued
book return
book not available/can't be issued
Return book
Entry/book returninig request
Do/return the book
Do/charge fine if late freturn
Exit/book returned
book returned

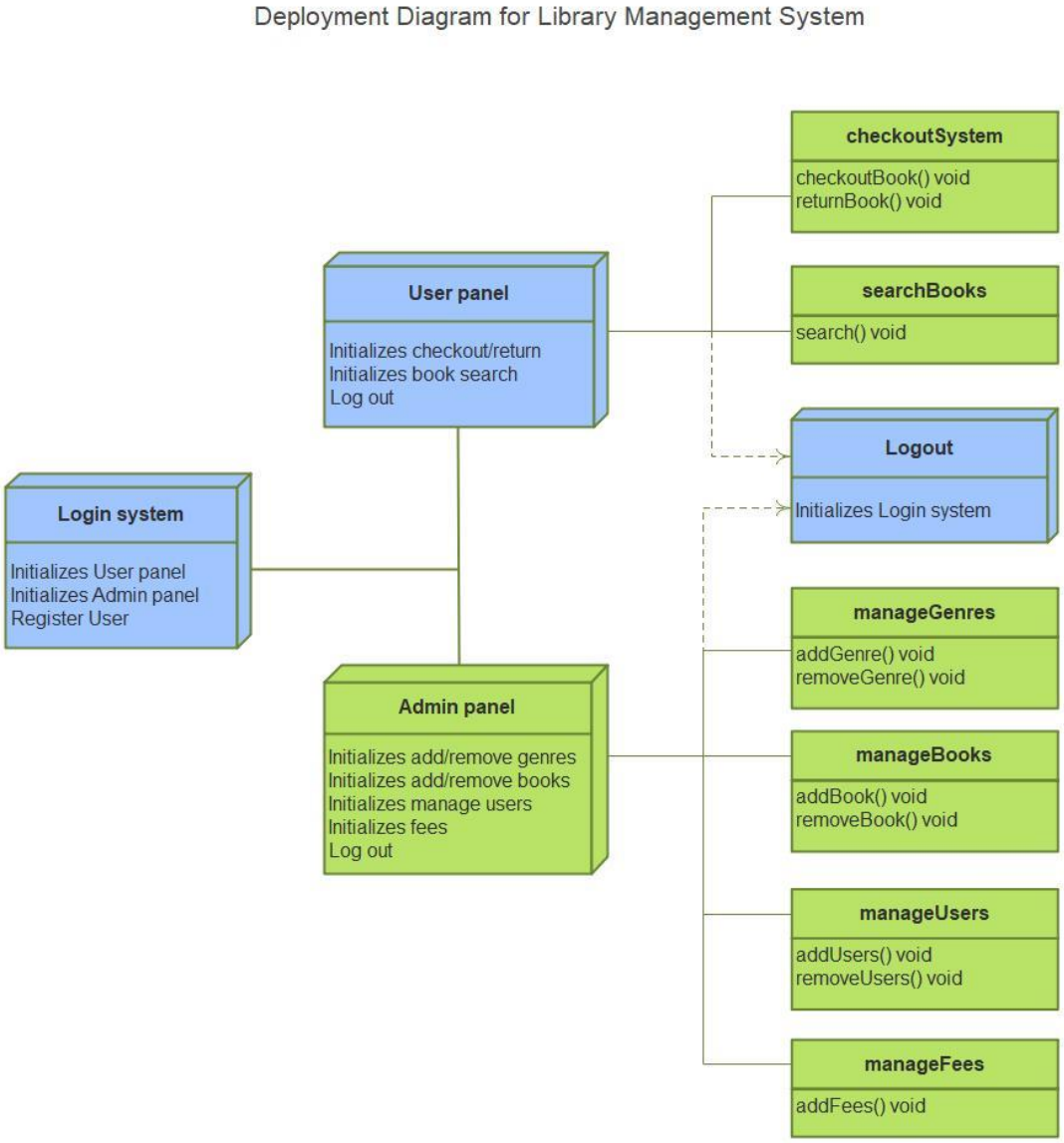
Library Management System State Diagram

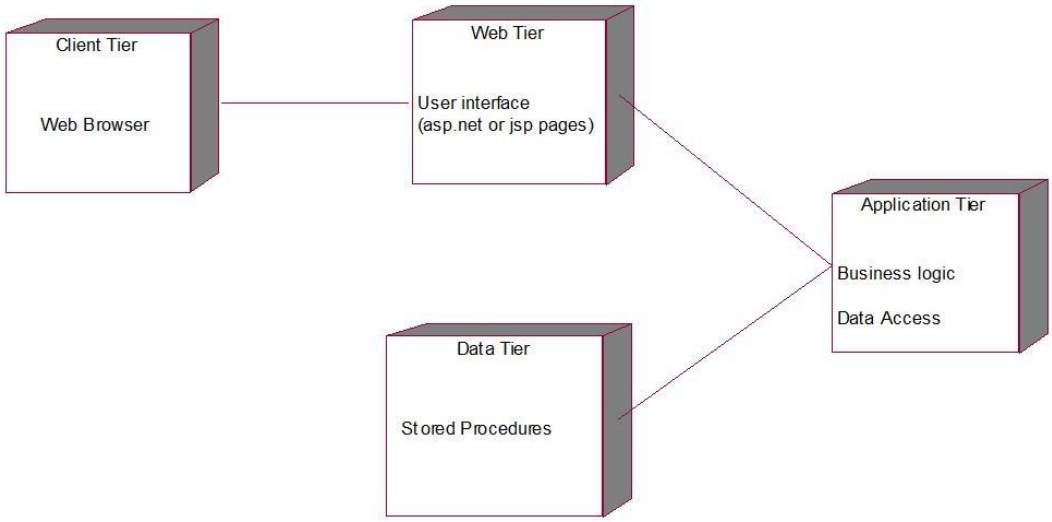


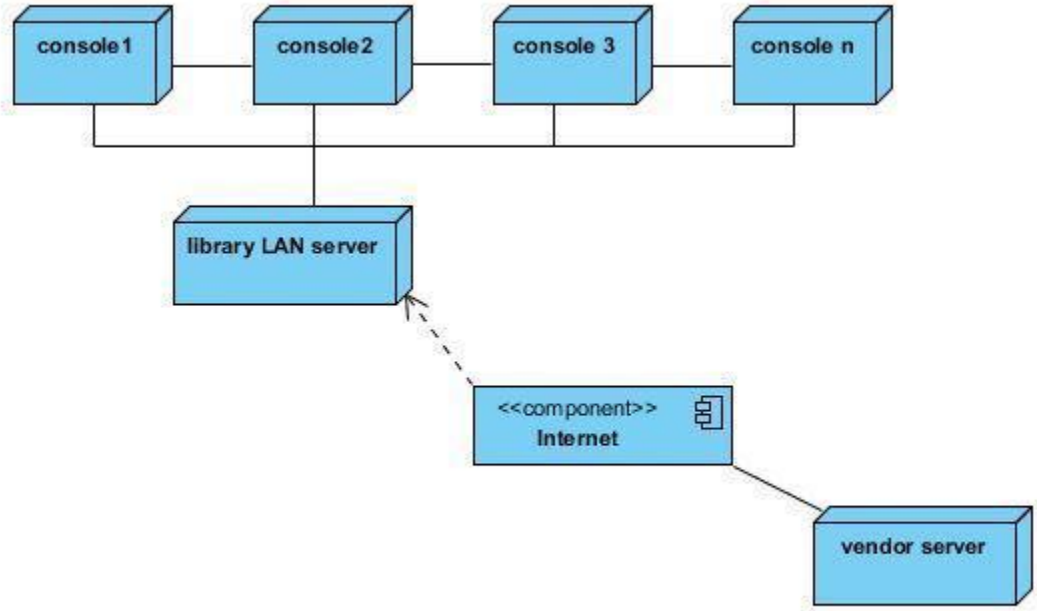
States

- Available
- Issued
- Reserved
- Returned

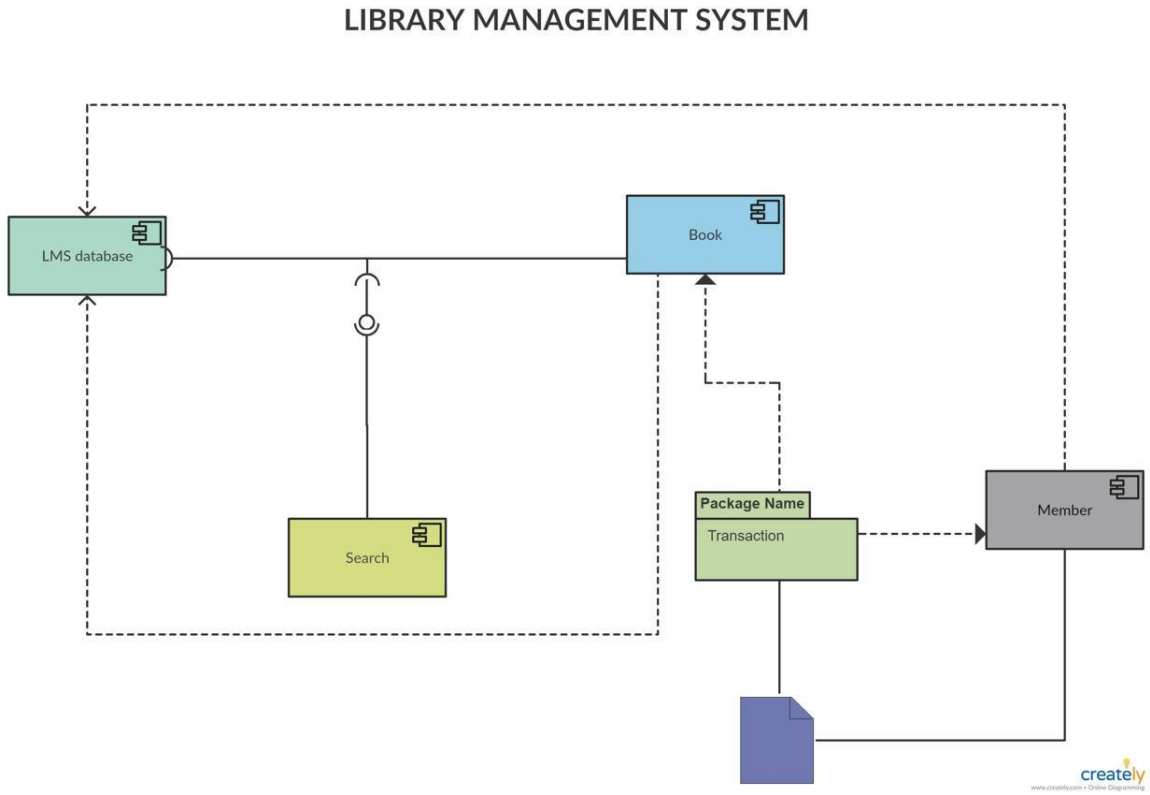
7. DEPLOYMENT DIAGRAM

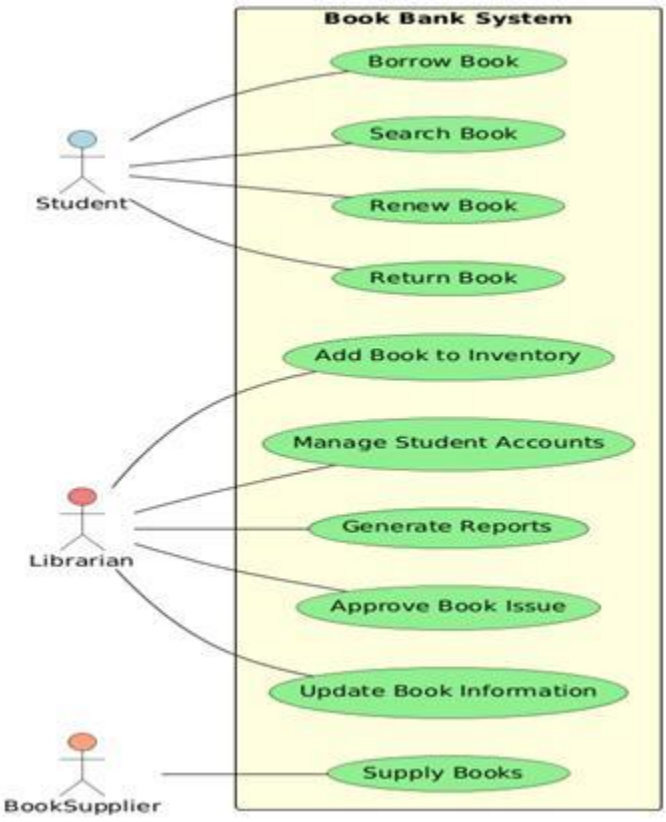


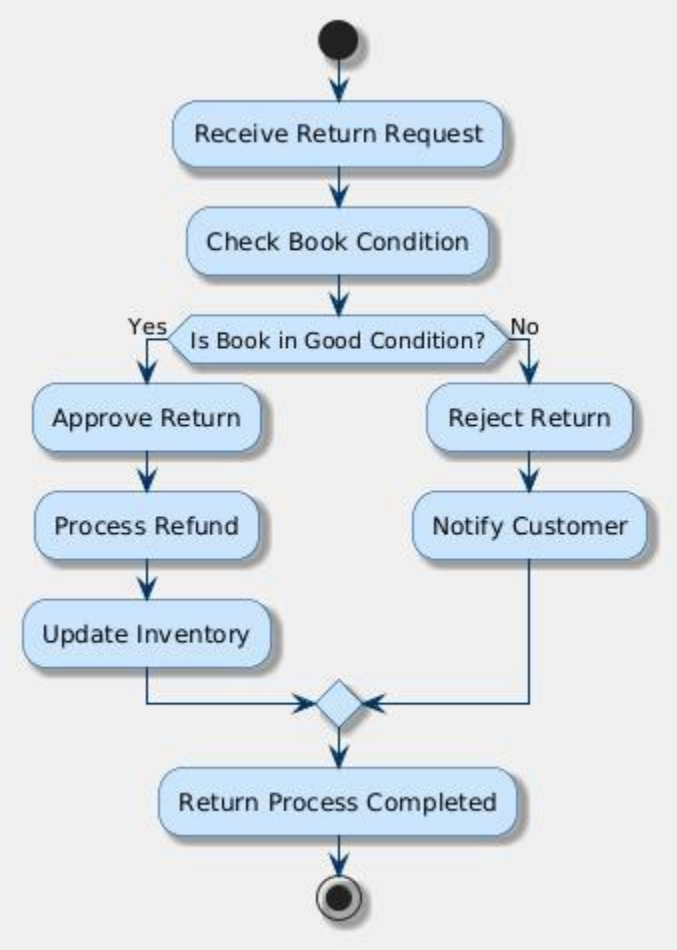

Client Tier
Web Browser
Web Tier
User interface
(asp.net or jsp pages)
Application Tier
Business logic
Data Access
Data Tier
Stored Procedures


console1
console2
console 3
console n
library LAN server
<<component>>
Internet
vendor server

8. COMPONENT DIAGRAM






Receive Return Request
Check Book Condition
Yes
Is Book in Good Condition?
No
Approve Return
Reject Return
Process Refund
Notify Customer
Update Inventory
Return Process Completed

EXPERIMENT –3: ONLINE EXAM REGISTRATION SYSTEM

AIM

Preparation of Software Configuration Management and Risk Management related documents.

PROCEDURE

(I) PROBLEM STATEMENT

The Exam Registration System (ERS) is designed to automate the process of exam registration for students.

- Students fill an online registration form (name, register number, etc.)
- The system verifies the details with the database
- Verified data is forwarded to the Exam Controller
- Exam schedule and fee details are provided
- Students complete fee payment
- After approval, the hall ticket is generated and issued

This system reduces manual effort and improves processing speed.

(II) SOFTWARE REQUIREMENT SPECIFICATION (SRS)

1. INTRODUCTION

ERS acts as an interface between the **Student** and the **Exam Controller**, improving efficiency in issuing hall tickets.

---

2. PURPOSE

Manual hall ticket processing is slow and inefficient. This system:

- Reduces processing time
- Handles large number of students
- Ensures data accuracy and security

---

3. SCOPE

- Online registration
- Exam details viewing

- Fee processing
- Status tracking
- Communication between student and controller

---

4. DEFINITIONS & ACRONYMS

- **Exam Controller**: System administrator
- **Student**: User applying for exam
- **ERS**: Exam Registration System
- **HTML, J2EE, HTTP, TCP/IP**: Web technologies

---

5. TECHNOLOGIES USED

- HTML
- JSP
- JavaScript
- Java

TOOLS USED

- Eclipse IDE
- Rational Rose (UML Tool)

---

6. OVERVIEW

SRS includes:

- Overall Description
- Specific Requirements

---

7. OVERALL DESCRIPTION

Product Perspective

ERS provides a secure interface between student and exam controller.

Software Interface

- Front End: JSP, HTML

- Server: Apache Tomcat
- Back End: Oracle 11g

Hardware Interface

- Client–Server architecture
- Central database

---

8. SYSTEM FUNCTIONS

- Student registration
- Exam schedule display
- Fee processing
- Status updates (SMS/Email)
- Report generation

---

9. USER CHARACTERISTICS

- **Student**: Registers and pays fees
- **Exam Controller**: Verifies and approves applications

---

10. CONSTRAINTS

- Requires computer access
- Security risks exist
- Accurate data entry required

---

11. ASSUMPTIONS

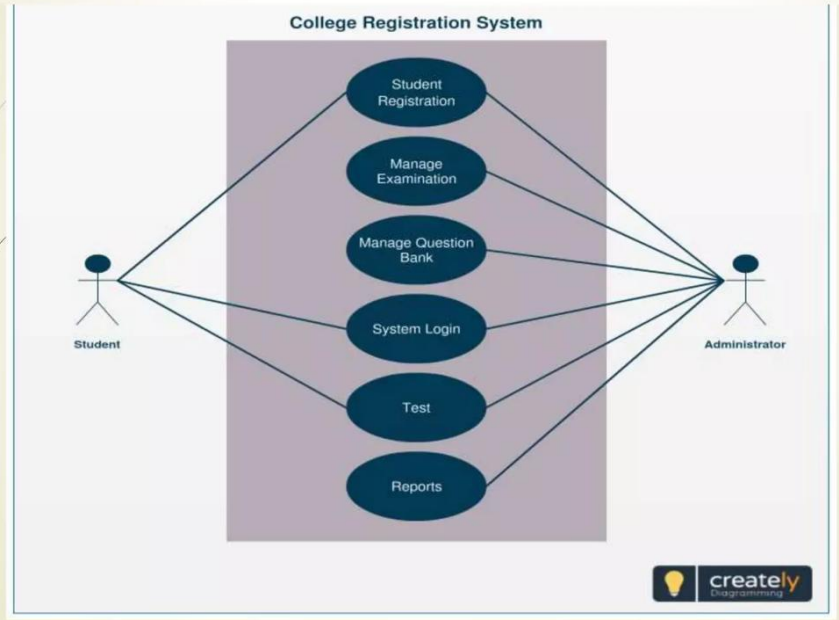
- Basic computer knowledge required
- Documents may need scanning

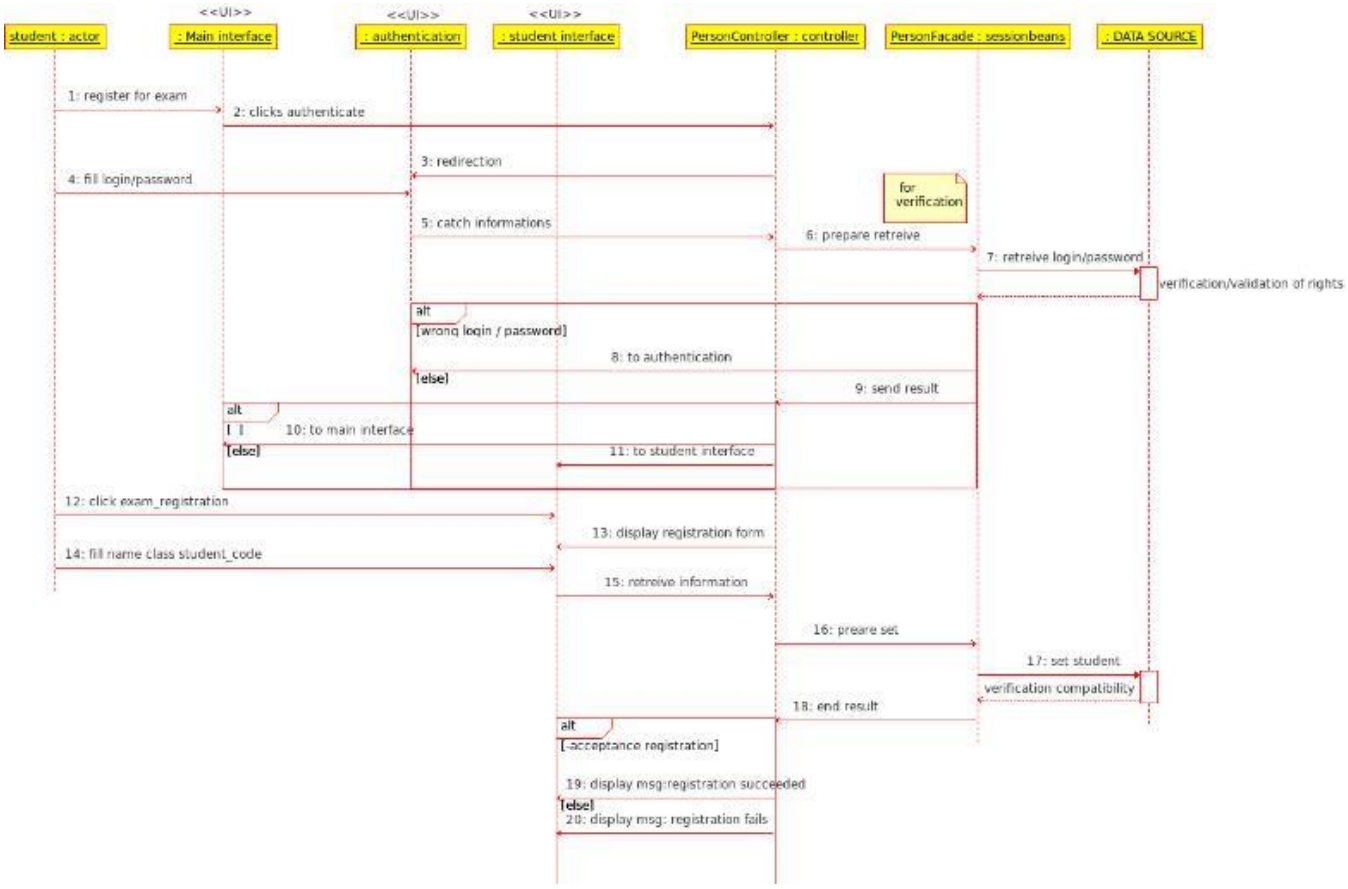
(III) UML DIAGRAMS

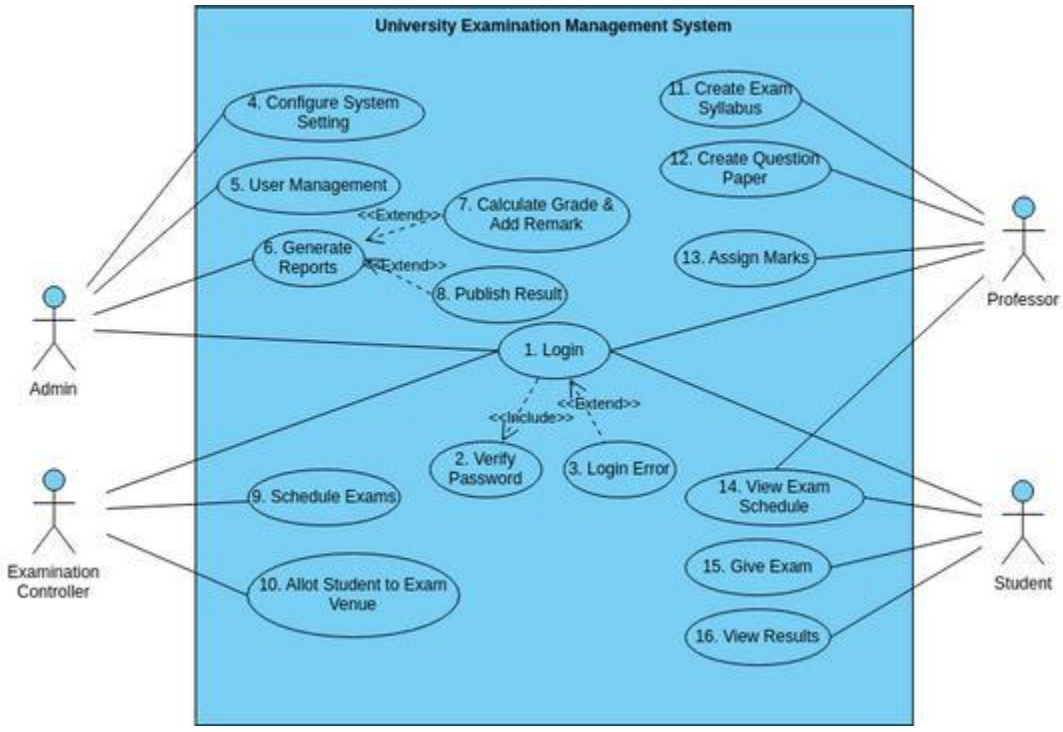
---

1. USE CASE DIAGRAM

Use Case diagrams







6

Actors

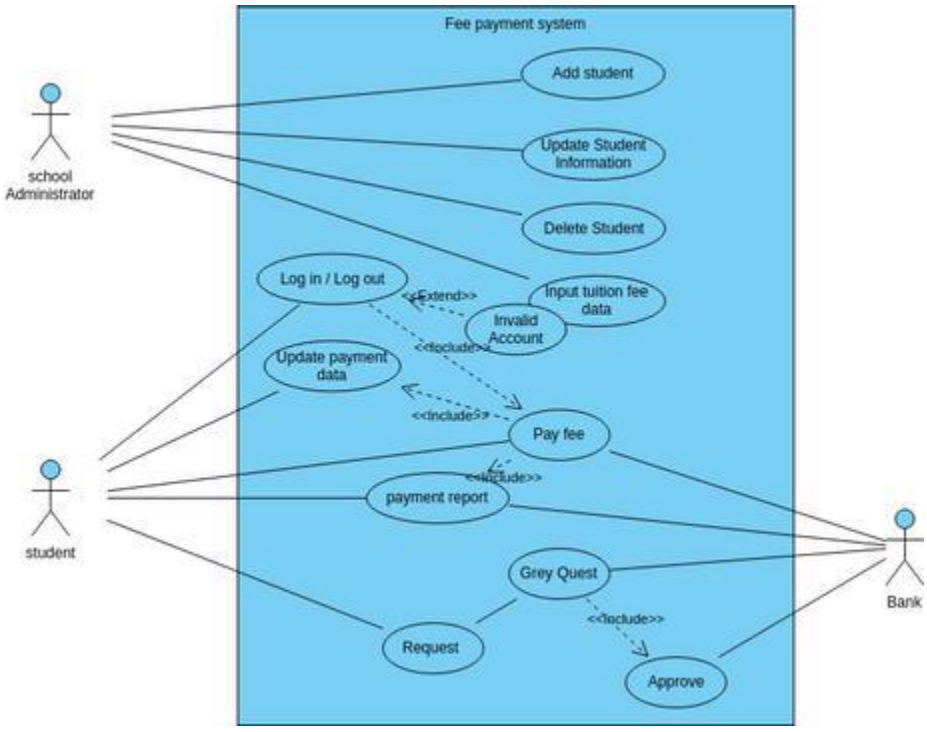
- Student
- System Database

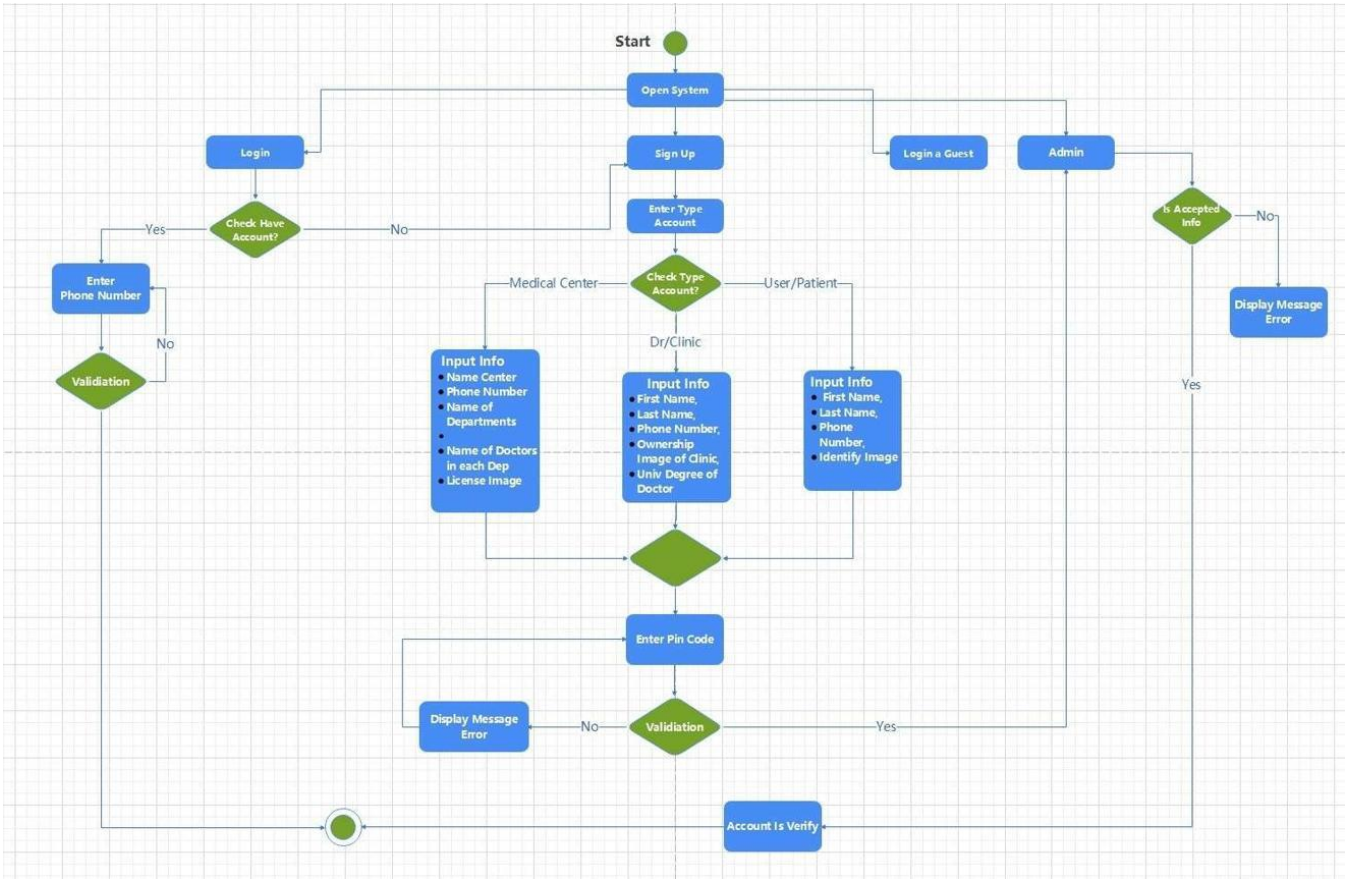
Use Cases

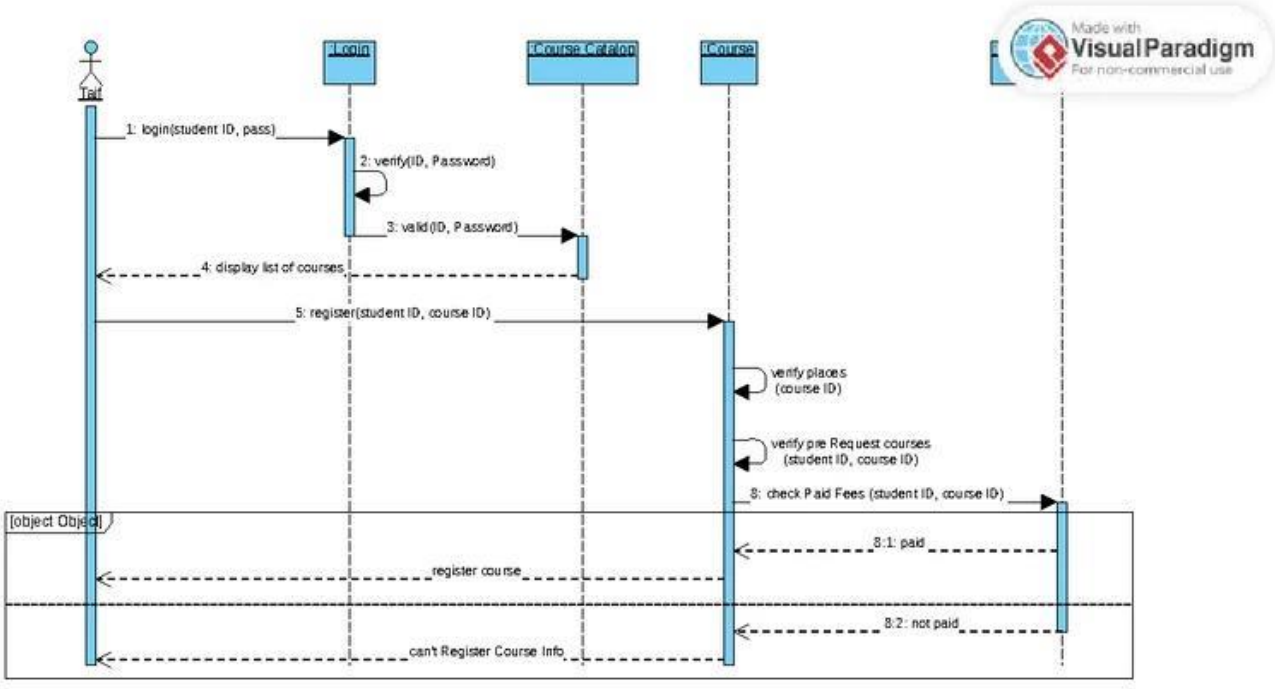
- Login
- View Exam Details
- Register
- Acknowledgement
- Fee Processing

---

2. ACTIVITY DIAGRAM


Fee payment system
Add student
Update Student Information
Delete Student
Input tuition fee data
Log in / Log out
<<Extend>>
Invalid Account
<<Include>>
Update payment data
<<Include>>
Pay fee
<<Include>>
payment report
Grey Quest
<<Include>>
Request
Approve
school Administrator
student
Bank


Start
Open System
Login
Sign Up
Login a Guest
Admin
Check Have Account?
Yes
No
Enter Type Account
Enter Phone Number
Validiation
No
Check Type Account?
Medical Center
User/Patient
Dr/Clinic
Input Info
Name Center
Phone Number
Name of Departments
Name of Doctors in each Dep
License Image
Input Info
First Name,
Last Name,
Phone Number,
Ownership Image of Clinic,
Univ Degree of Doctor
Input Info
First Name,
Last Name,
Phone Number,
Identify Image
Is Accepted Info
No
Display Message Error
Yes
Enter Pin Code
Display Message Error
No
Validiation
Yes
Account Is Verify



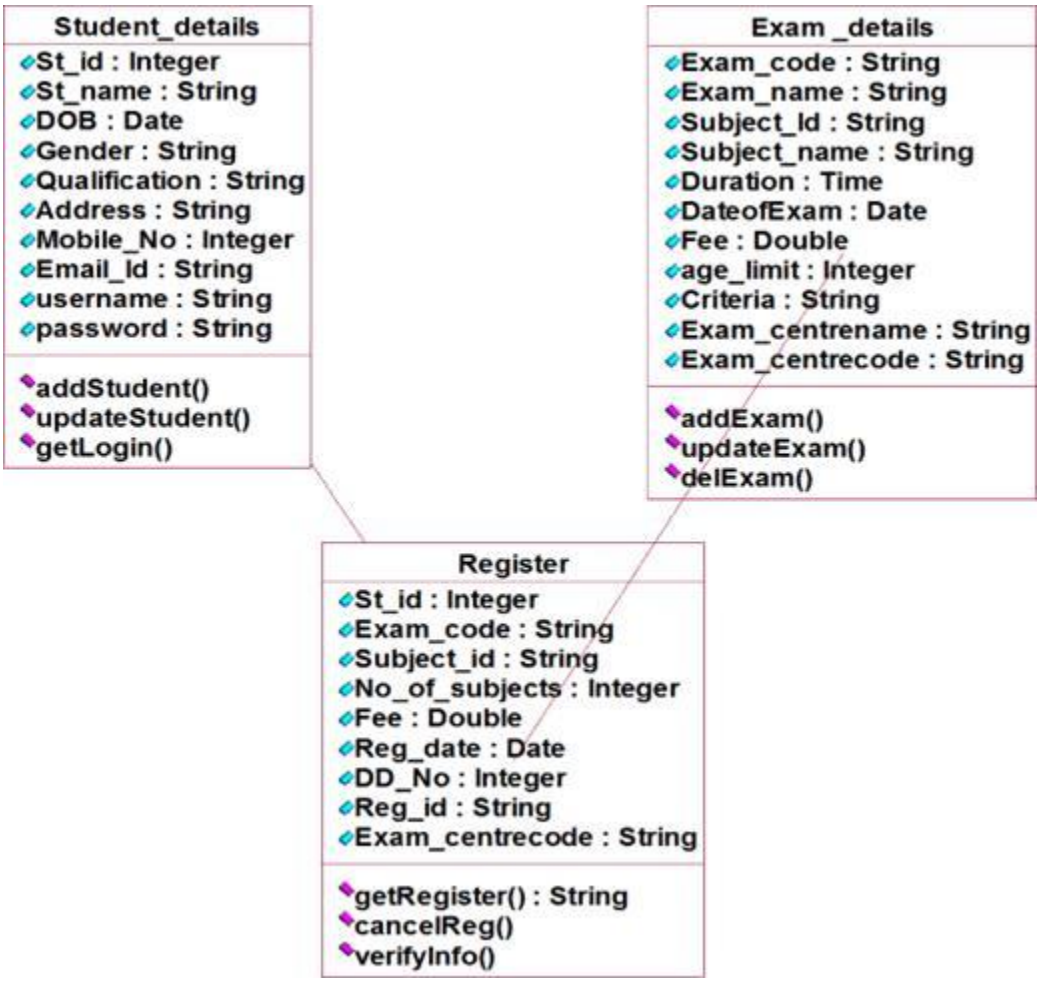
6

Flow

- Login
- View exam details
- Fill registration form
- Pay fees
- Verification
- Generate hall ticket

---

3. CLASS DIAGRAM


Student_details
St_id : Integer
St_name : String
DOB : Date
Gender : String
Qualification : String
Address : String
Mobile_No : Integer
Email_Id : String
username : String
password : String
addStudent()
updateStudent()
getLogin()
Exam _details
Exam_code : String
Exam_name : String
Subject_Id : String
Subject_name : String
Duration : Time
DateofExam : Date
Fee : Double
age_limit : Integer
Criteria : String
Exam_centrename : String
Exam_centrecode : String
addExam()
updateExam()
delExam()
Register
St_id : Integer
Exam_code : String
Subject_id : String
No_of_subjects : Integer
Fee : Double
Reg_date : Date
DD_No : Integer
Reg_id : String
Exam_centrecode : String
getRegister() : String
cancelReg()
verifyInfo()

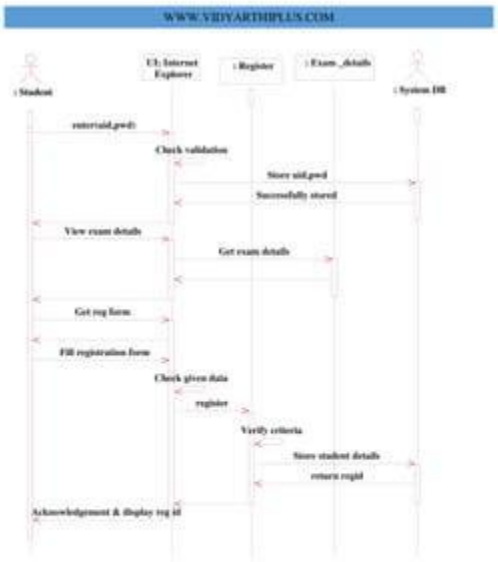
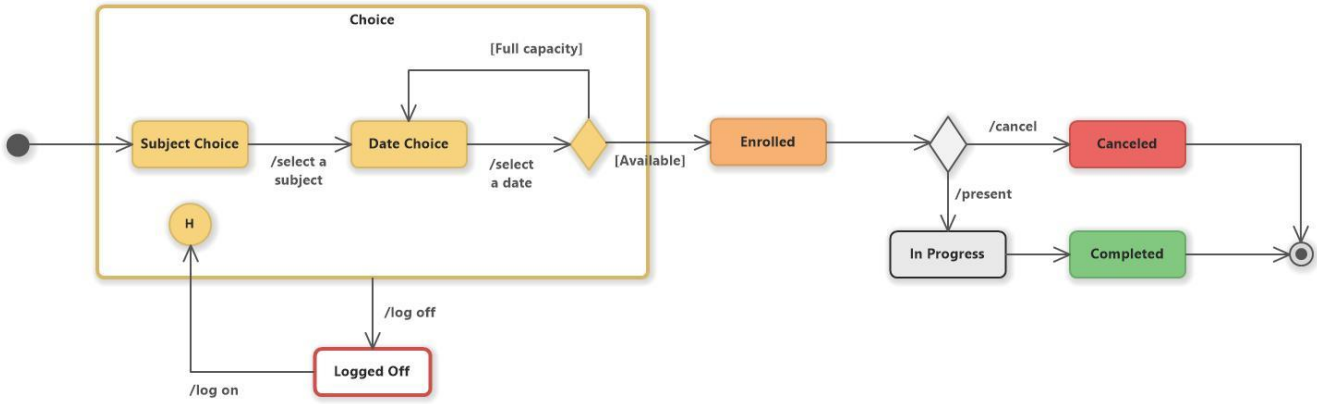


Fig. 4.1. SEQUENCE DIAGRAM FOR REGISTRATION SYSTEM

The sequence and collaboration diagram represents that the student enter the information to get the hall ticket and the exam controller issues the hall ticket after verifying the necessary items and this data are stored in the database.



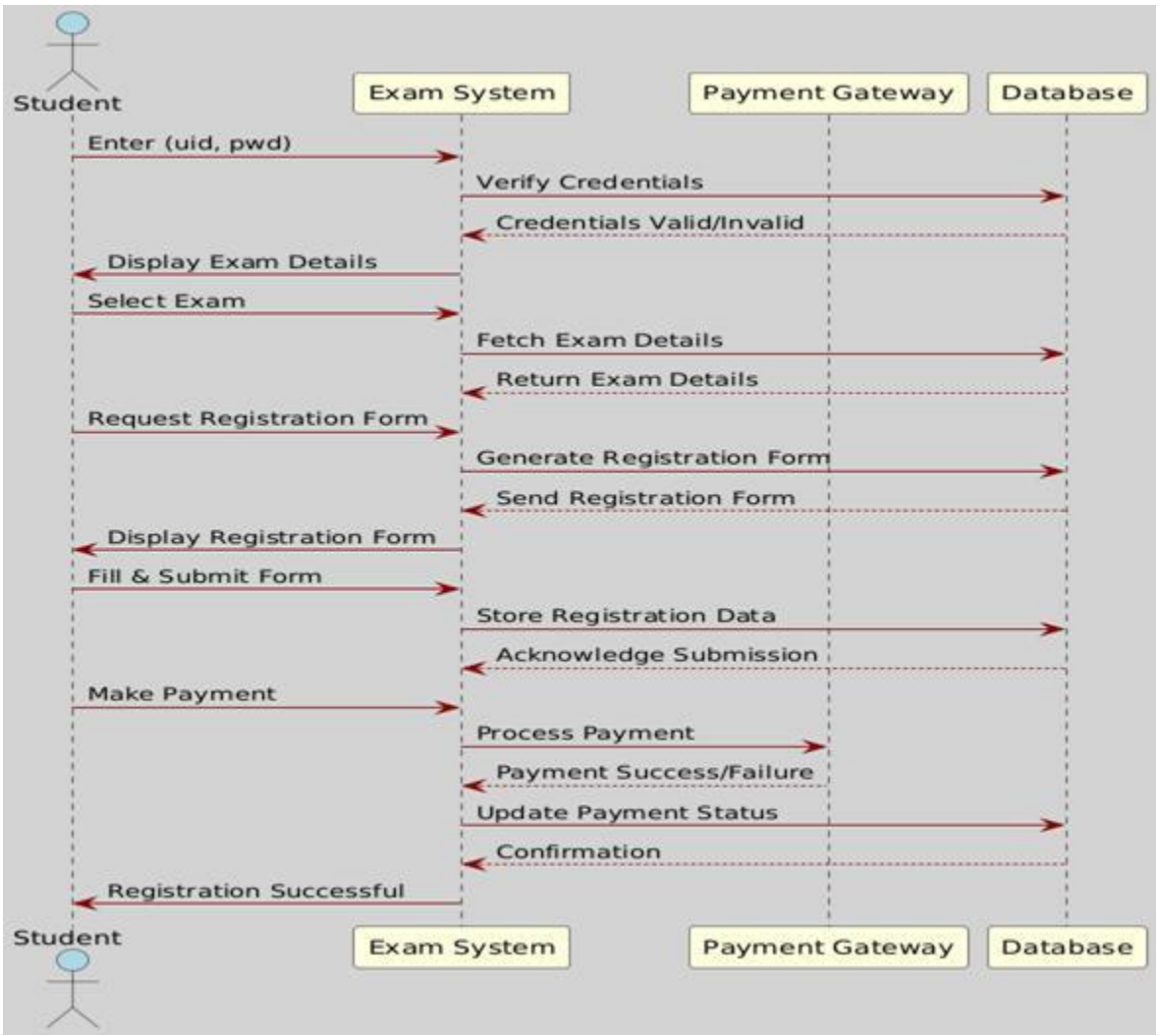
6

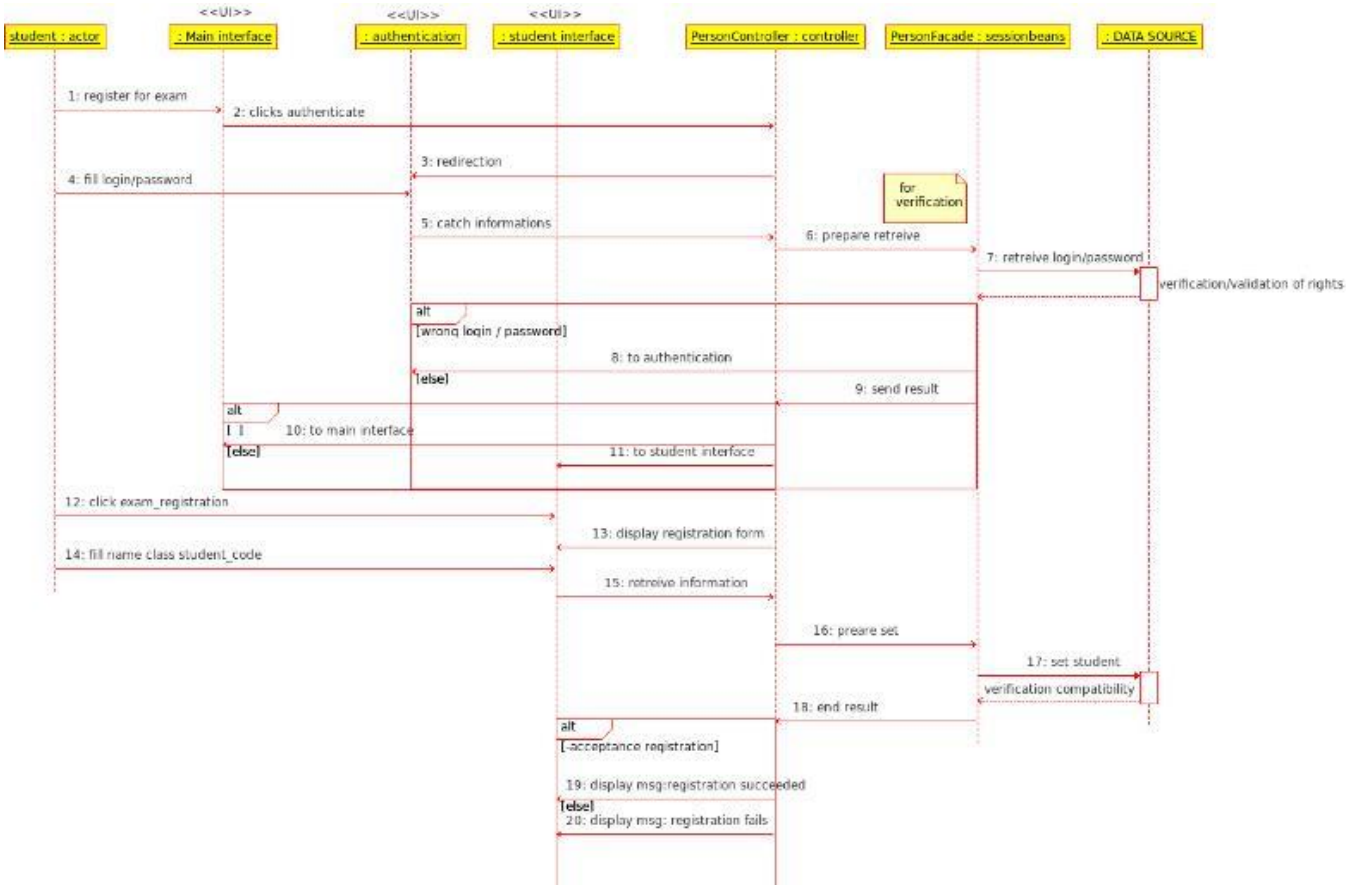
Classes

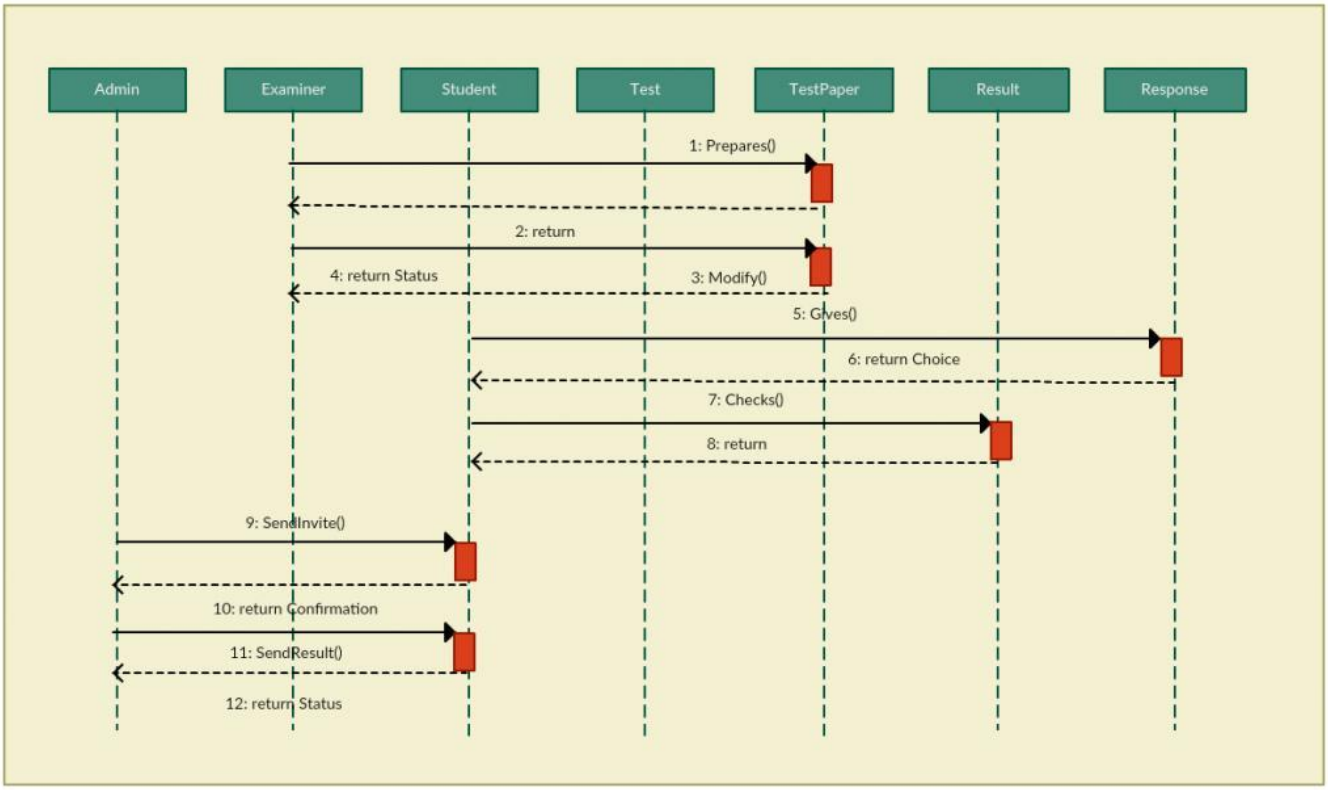
1. Student_Details
2. Exam_Details
3. Register

---

4. SEQUENCE DIAGRAM


Student
Exam System
Payment Gateway
Database
Enter (uid, pwd)
Verify Credentials
Credentials Valid/Invalid
Display Exam Details
Select Exam
Fetch Exam Details
Return Exam Details
Request Registration Form
Generate Registration Form
Send Registration Form
Display Registration Form
Fill & Submit Form
Store Registration Data
Acknowledge Submission
Make Payment
Process Payment
Payment Success/Failure
Update Payment Status
Confirmation
Registration Successful
Student
Exam System
Payment Gateway
Database


student : actor
<<UI>>
: Main interface
<<UI>>
: authentication
<<UI>>
: student interface
PersonController : controller
PersonFacade : sessionbeans
: DATA SOURCE
1: register for exam
2: clicks authenticate
3: redirection
4: fill login/password
for verification
5: catch informations
6: prepare retreive
7: retreive login/password
verification/validation of rights
alt
[wrong login / password]
8: to authentication
[else]
9: send result
alt
10: to main interface
[else]
11: to student interface
12: click exam_registration
13: display registration form
14: fill name class student_code
15: retreive information
16: preare set
17: set student
verification compatibility
18: end result
alt
[-acceptance registration]
19: display msg:registration succeeded
[else]
20: display msg: registration fails


Admin
Examiner
Student
Test
TestPaper
Result
Response
1: Prepares()
2: return
4: return Status
3: Modify()
5: Gives()
6: return Choice
7: Checks()
8: return
9: SendInvite()
10: return Confirmation
11: SendResult()
12: return Status

5

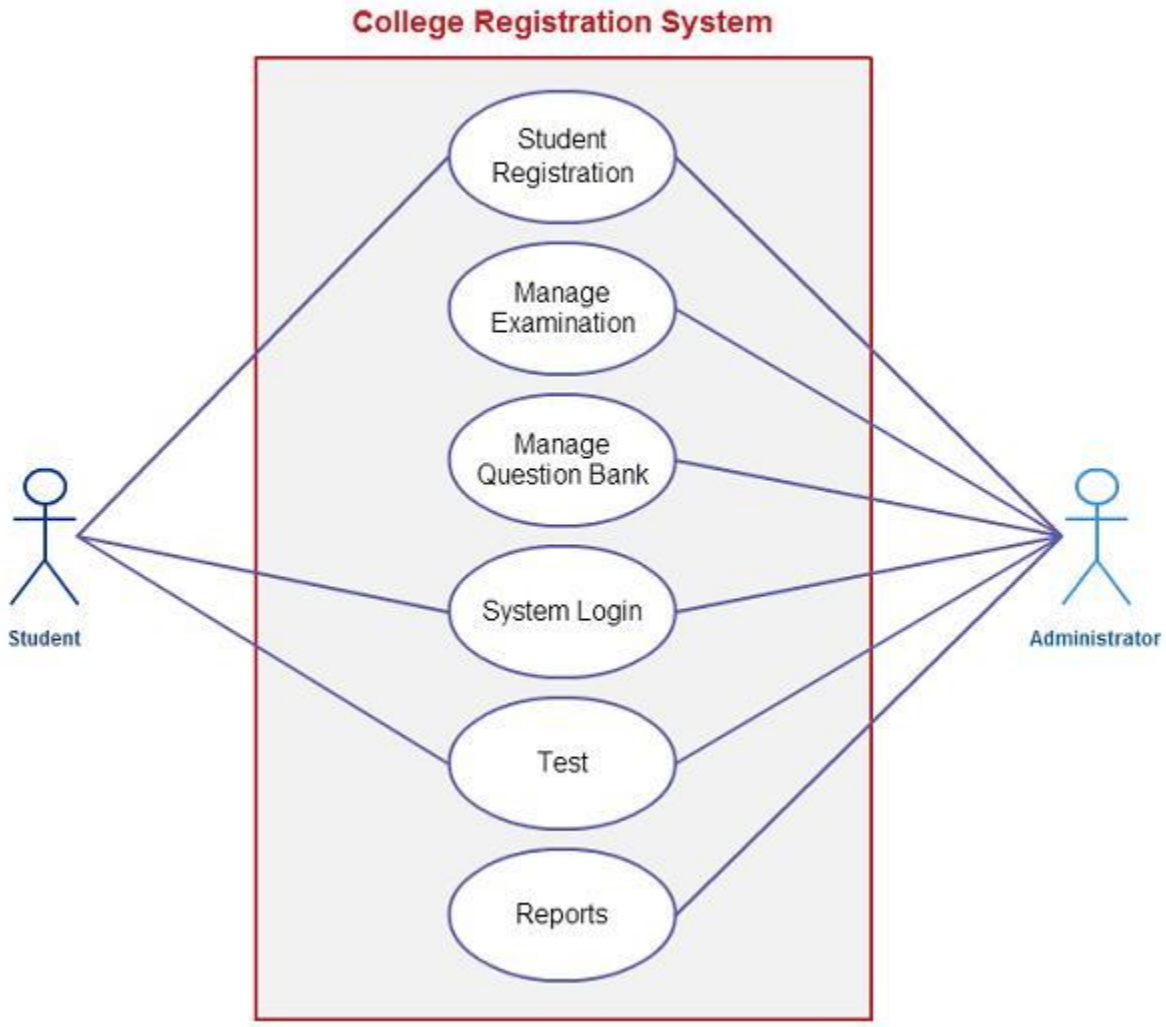
Description

Shows interaction between:

- Student
- System
- Database

---

5. COLLABORATION DIAGRAM



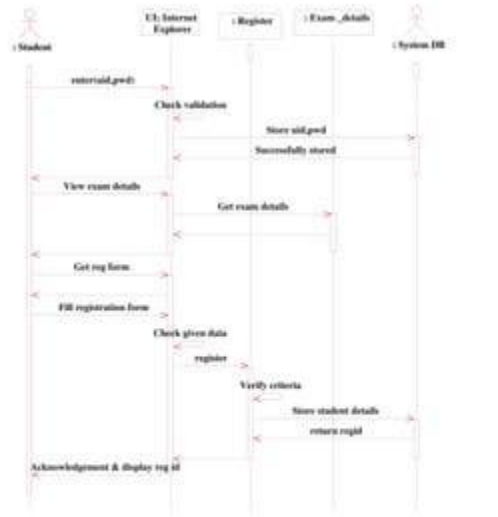
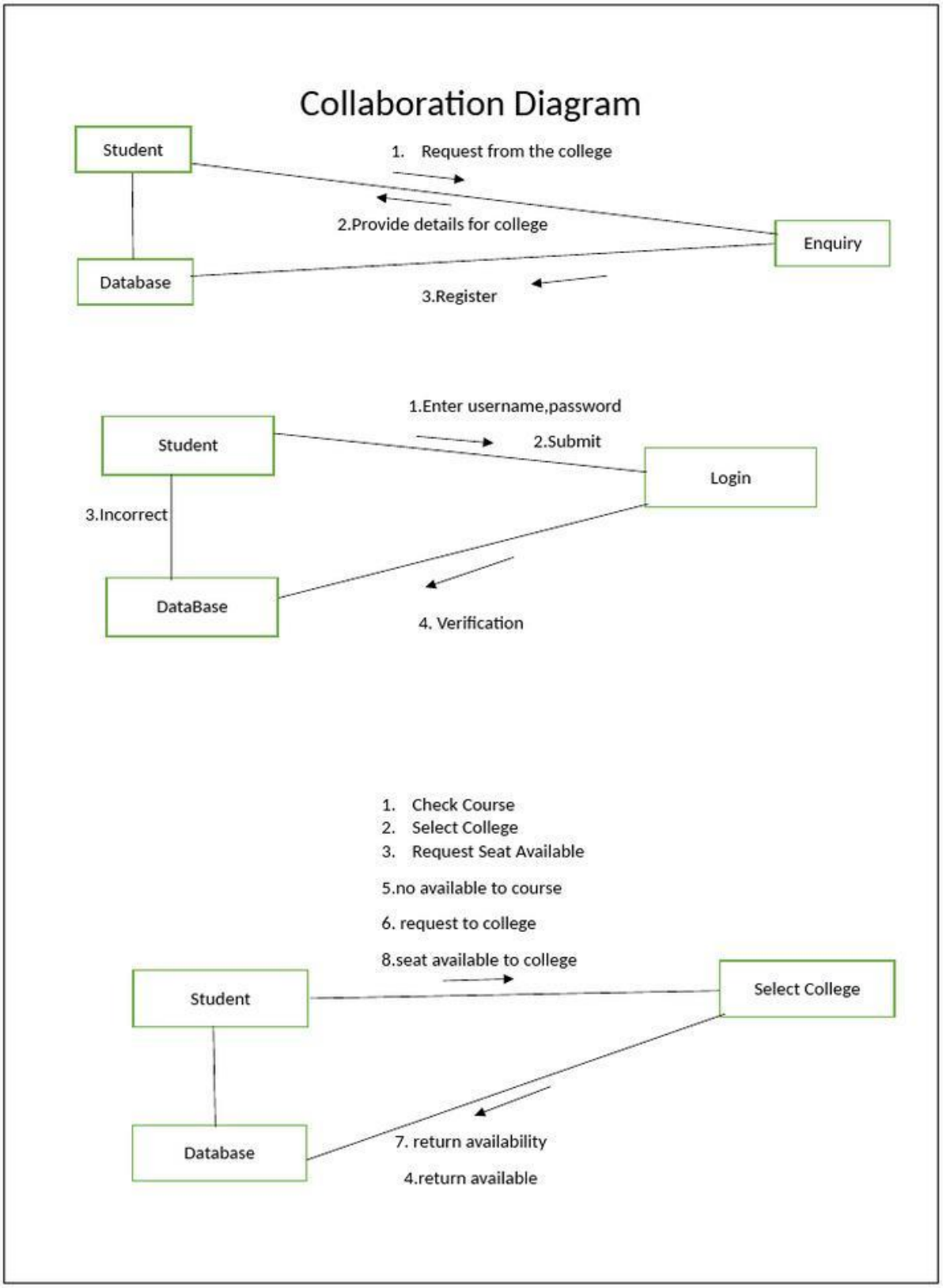
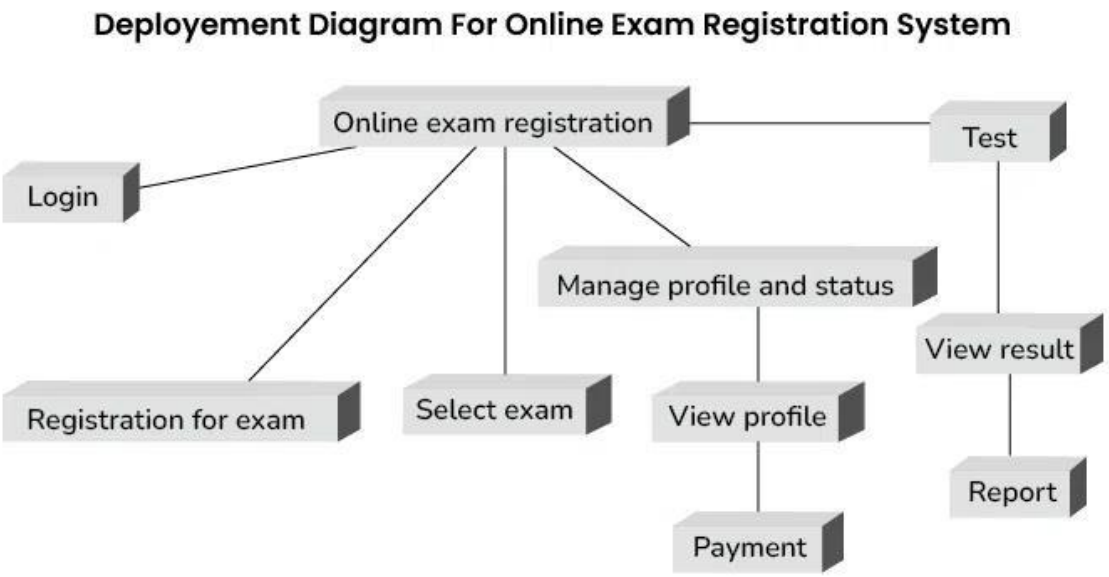


Fig. 6.1. SEQUENCE DIAGRAM FOR REGISTRATION SYSTEM

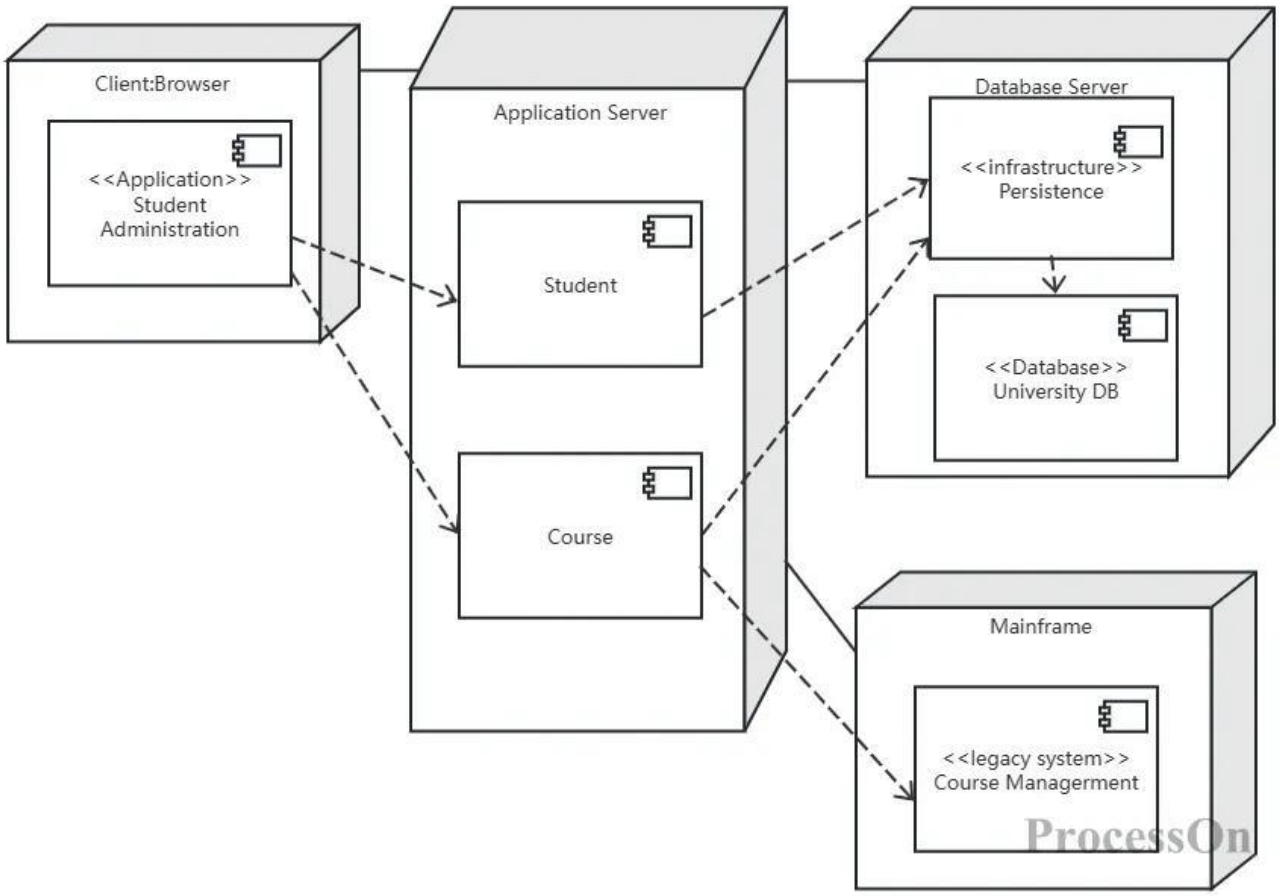
The sequence and collaboration diagram represents that the student enter the information to get the hall ticket and the exam controller issues the hall ticket after verifying the necessary items and this data are stored in the database.

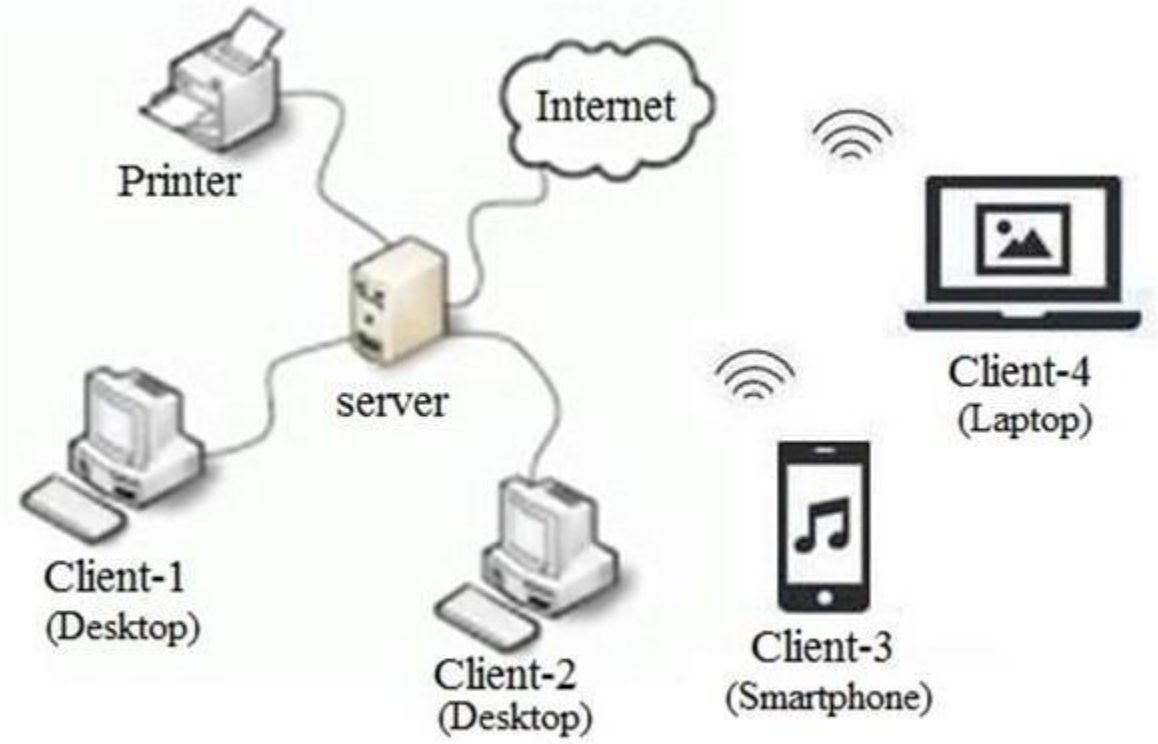

Collaboration Diagram
Student
1. Request from the college
2.Provide details for college
Enquiry
Database
3.Register
1.Enter username,password
Student
2.Submit
Login
3.Incorrect
DataBase
4. Verification
1. Check Course
2. Select College
3. Request Seat Available
5.no available to course
6. request to college
8.seat available to college
Student
Select College
Database
7. return availability
4.return available

6. DEPLOYMENT DIAGRAM



Système de sélection de cours – Diagramme de déploiement





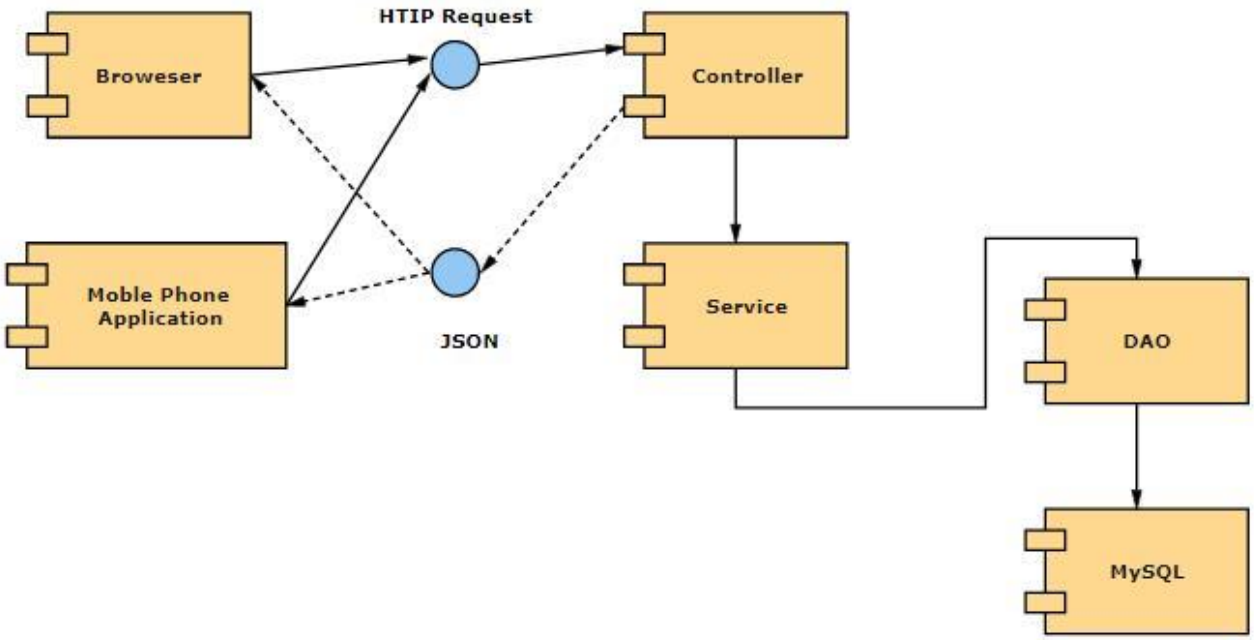
Client Server Architecture

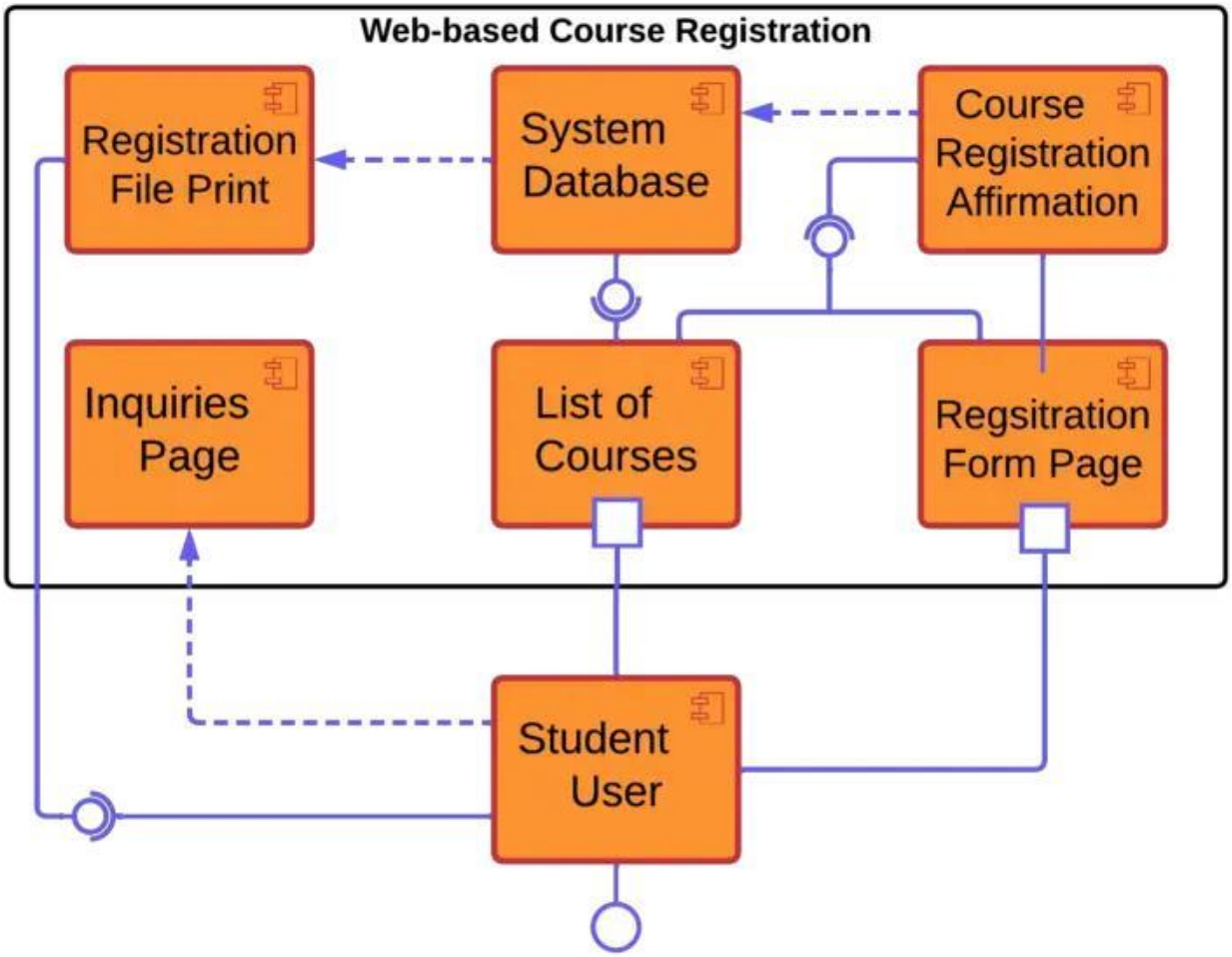
5

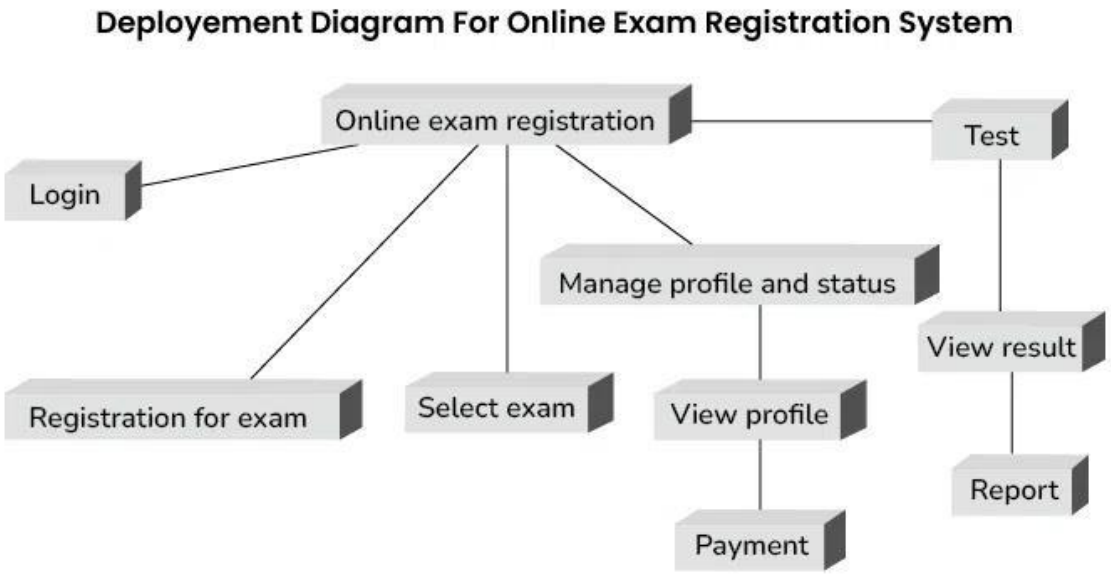
---

7. COMPONENT DIAGRAM

System Component Diagram




Web-based Course Registration
Registration File Print
System Database
Course Registration Affirmation
Inquiries Page
List of Courses
Regsitration Form Page
Student User



CONCLUSION

The Exam Registration System:

- Automates the exam registration process
- Reduces manual workload
- Ensures faster hall ticket generation
- Provides real-time updates to students

EXPERIMENT –4: STOCK MAINTENANCE SYSTEM

AIM

To create a system to perform stock maintenance operations efficiently.

PROCEDURE

(I) PROBLEM STATEMENT

The Stock Maintenance System is designed to manage and track stock details in an organization.

- Maintains records of items added and removed
- Tracks sales and purchase details
- Allows salesperson to update and view stock information
- Helps analyze business performance

The system ensures efficient handling of inventory and improves decision-making.

(II) SOFTWARE REQUIREMENT SPECIFICATION (SRS)

1. INTRODUCTION

Stock Maintenance System acts as an interface between the **Customer** and the **Salesperson**, improving efficiency in stock management.

---

2. PURPOSE

Manual stock handling is inefficient. This system:

- Reduces manual effort
- Improves accuracy
- Handles increasing customer demand

---

3. SCOPE

- Customer order placement
- Stock tracking
- Sales and purchase management
- Communication between customer and salesperson

4. DEFINITIONS & ACRONYMS

- **Customer**: Person who places orders
- **Salesperson**: Maintains stock details
- **Market Data Provider**: Analyzes product demand

5. TECHNOLOGIES USED

- Visual Studio
- VBScript

TOOLS USED

- Eclipse IDE
- Rational Rose (UML Tool)

6. OVERVIEW

SRS includes:

- Overall Description
- Specific Requirements

7. OVERALL DESCRIPTION

The system provides a simple and secure interface between customer and salesperson for managing stock efficiently.

8. SYSTEM FUNCTIONS

- Order placement by customer
- Stock update by salesperson
- Delivery scheduling
- Sales tracking

9. USER CHARACTERISTICS

- **Customer**: Places orders

- **Salesperson**: Manages stock and delivery
- **Company**: Maintains overall stock data

---

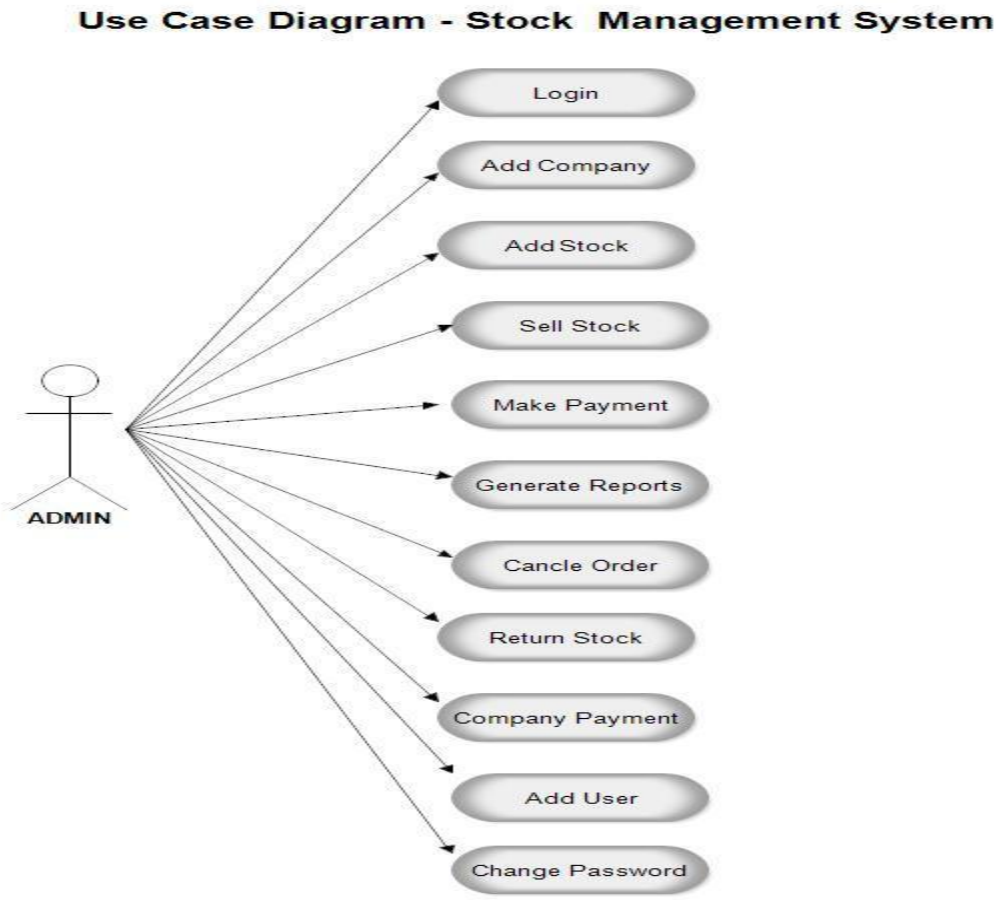
10. CONSTRAINTS

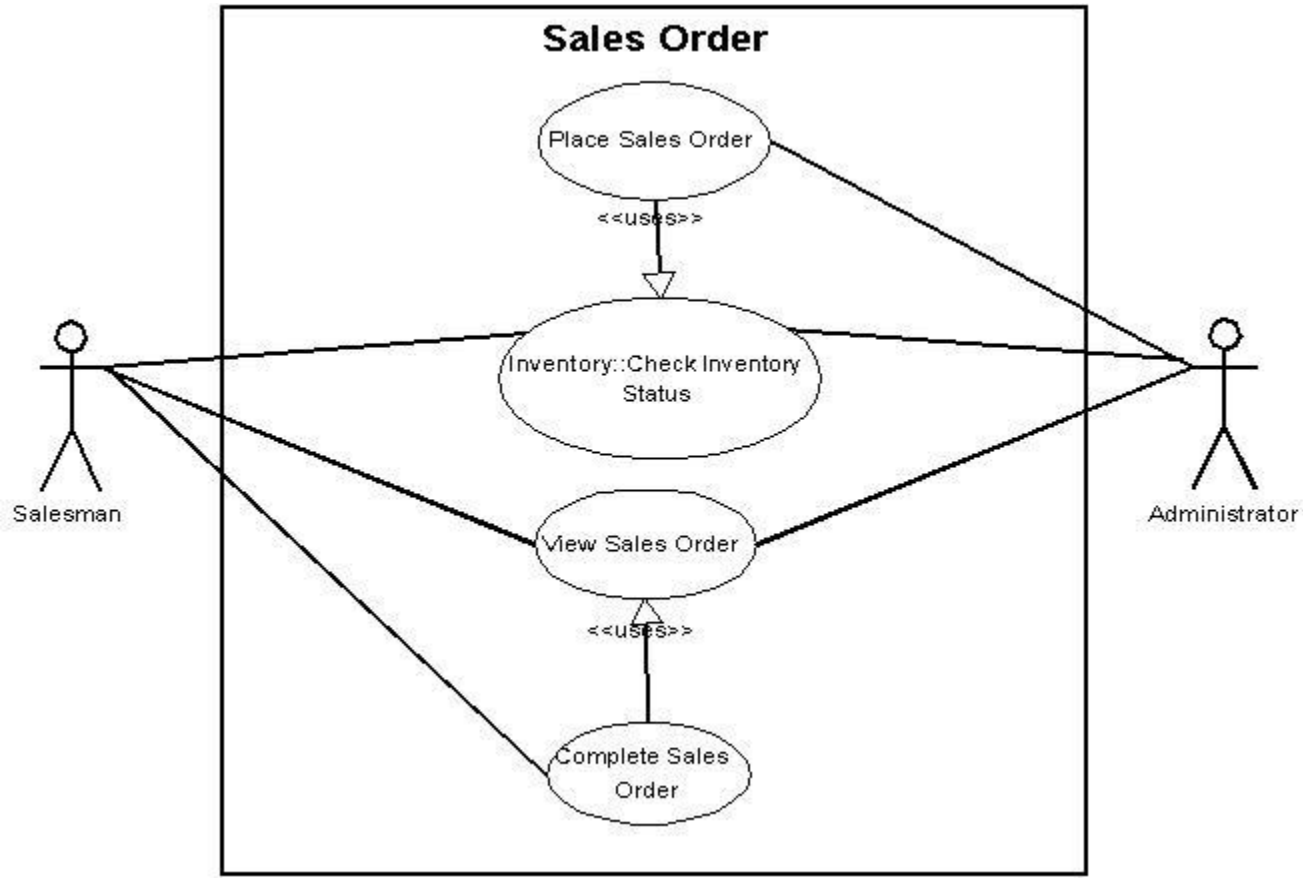
- Customer must wait for availability
- Stock depends on supplier updates
- Manual errors possible if data incorrect

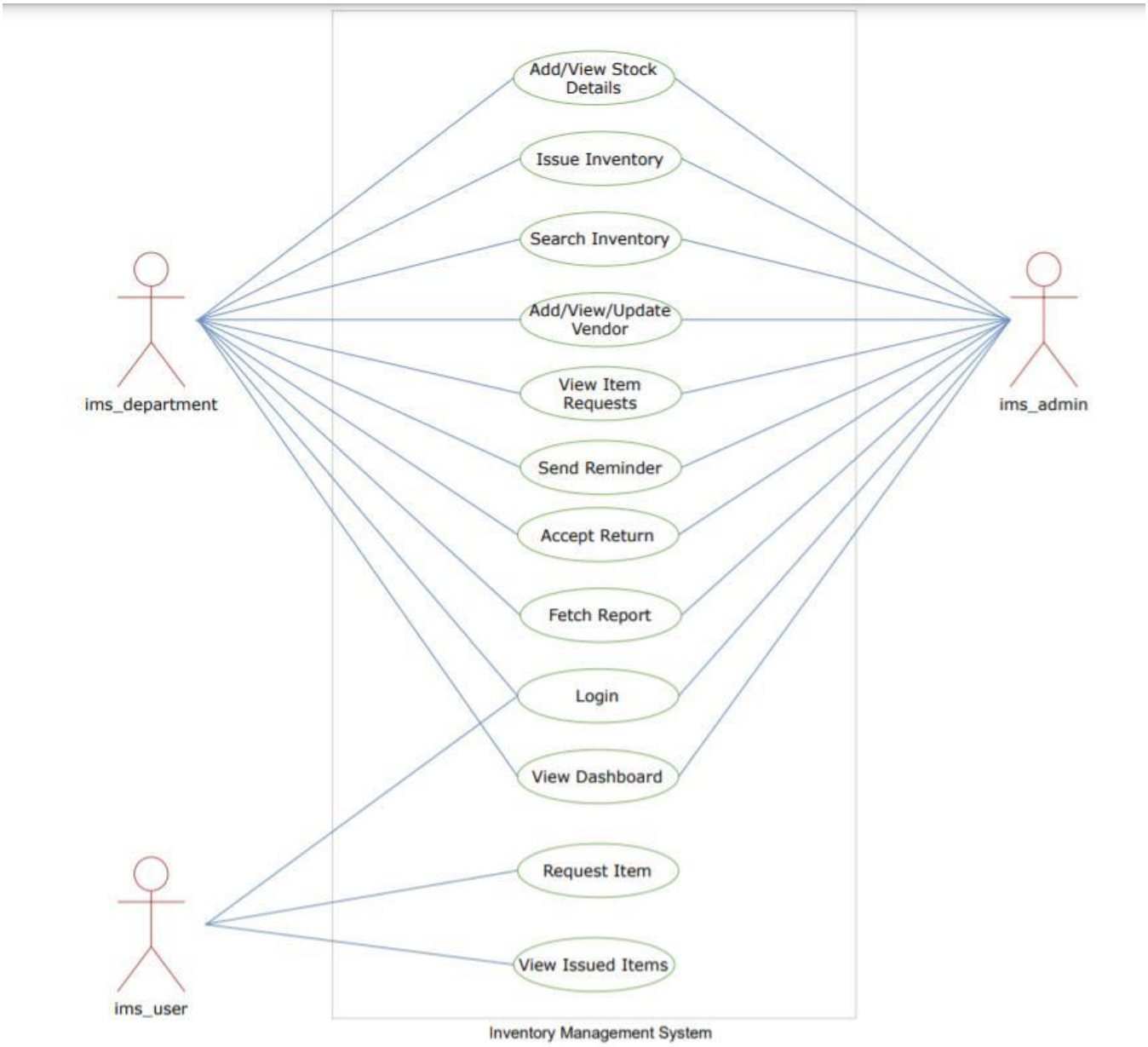
(III) UML DIAGRAMS

---

1. USE CASE DIAGRAM




Sales Order
Place Sales Order
<<uses>>
Inventory::Check Inventory Status
View Sales Order
<<uses>>
Complete Sales Order
Salesman
Administrator



6

Actors

- Customer
- Shopkeeper (Salesperson)
- Company

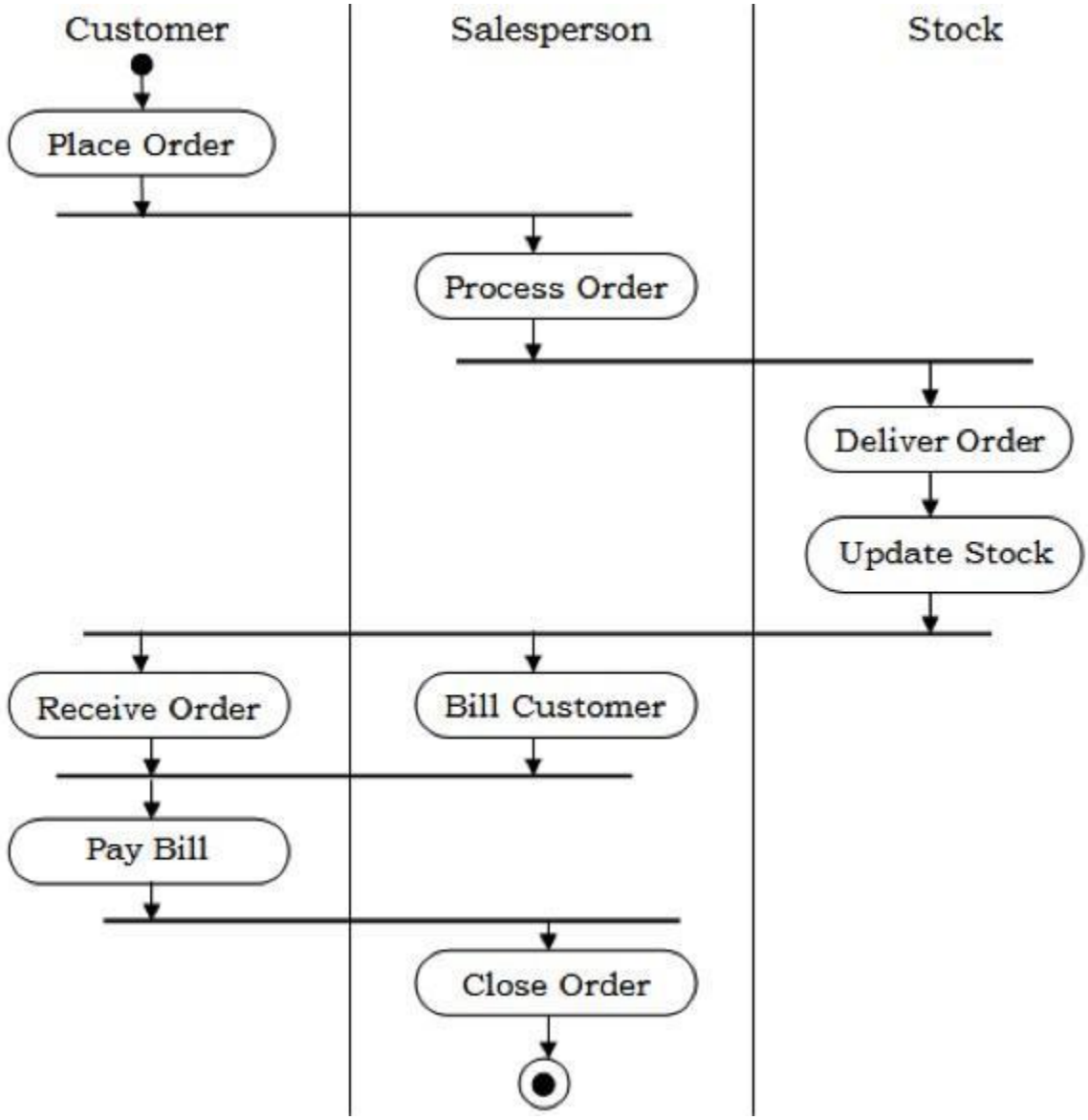
Use Cases

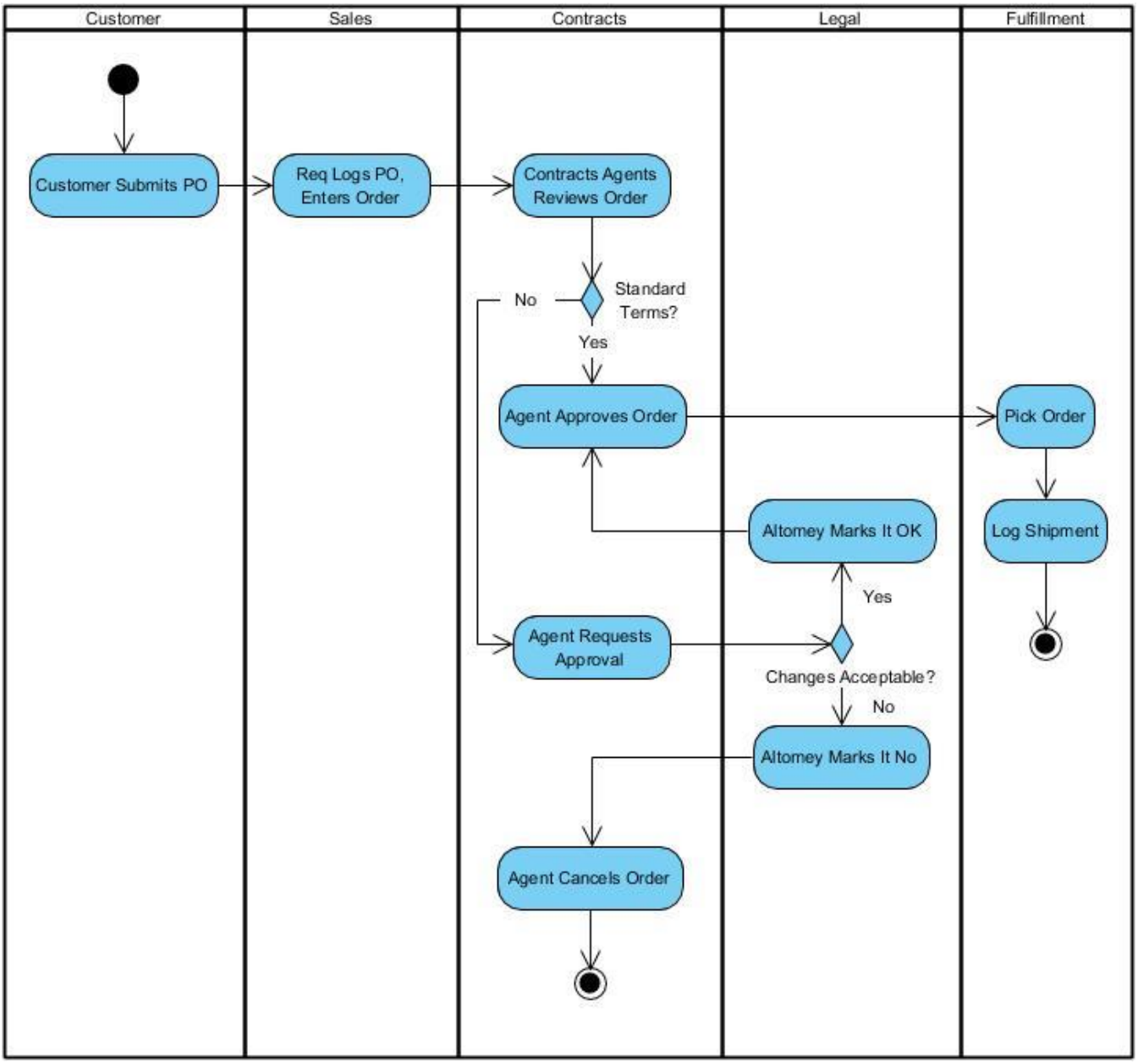
- Product Details
- Purchase Details

- Sales Details
- Stock Details
- Purchase Product
- Supply Product

---

2. ACTIVITY DIAGRAM




Customer
Sales
Contracts
Legal
Fulfillment
Customer Submits PO
Req Logs PO, Enters Order
Contracts Agents Reviews Order
No
Standard Terms?
Yes
Agent Approves Order
Pick Order
Altorney Marks It OK
Log Shipment
Yes
Agent Requests Approval
Changes Acceptable?
No
Altorney Marks It No
Agent Cancels Order

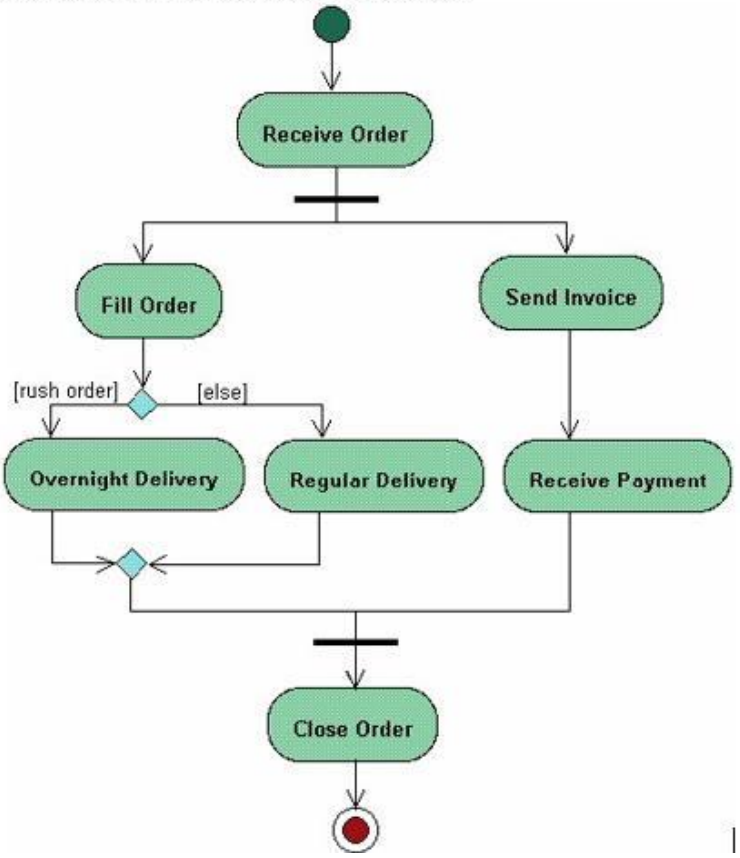
Draw the activity diagram for processing an order.

The diagram shows the flow of actions in the system's workflow.

Once the order is received the activities split into two parallel sets of activities: Order and Billing

On the Fill Order side, the method of delivery is decided conditionally, depending on the condition, either the Overnight Delivery activity or, the Regular Delivery activity is performed.

Finally the parallel activities combine to close the order.



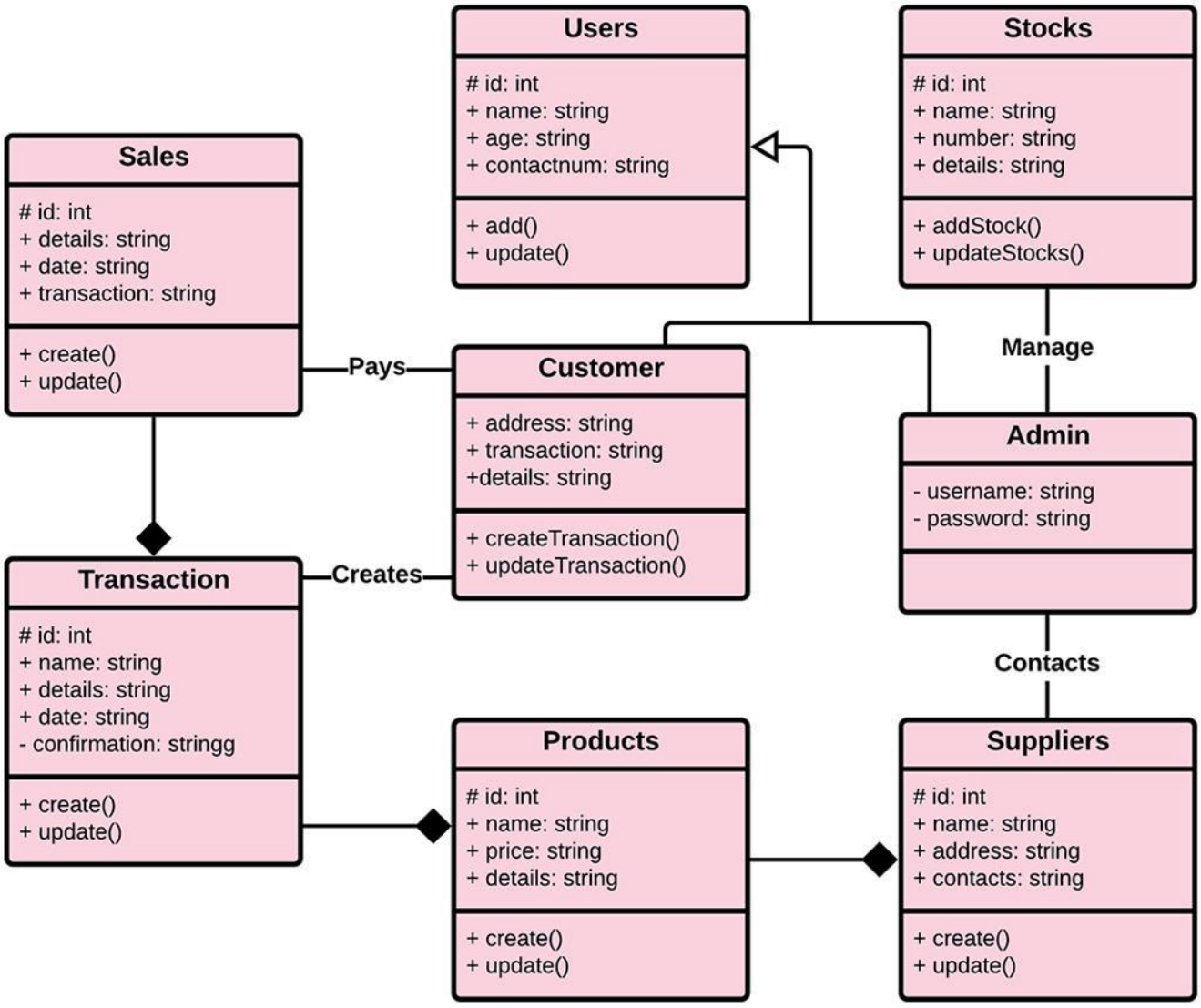
7

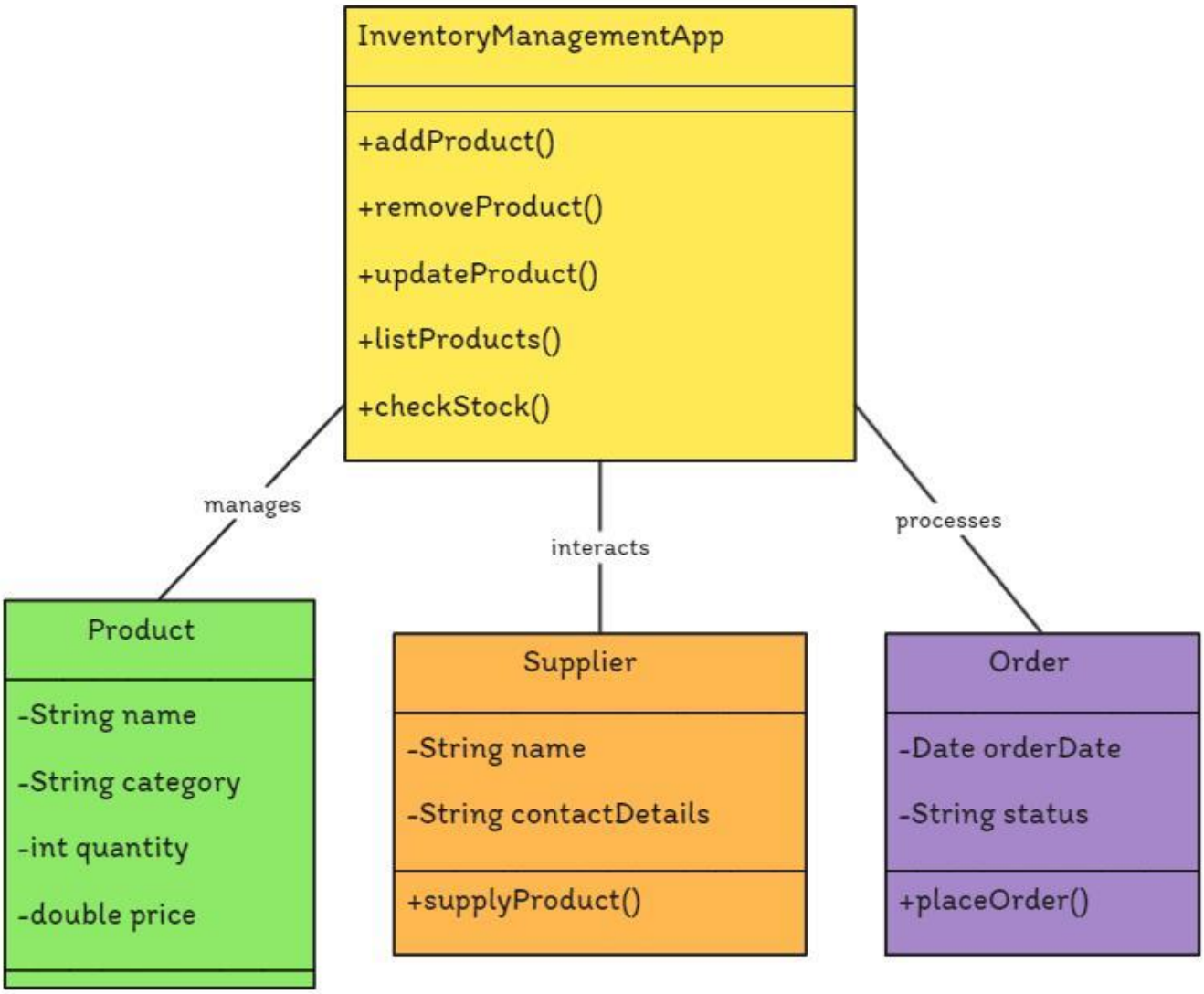
Flow

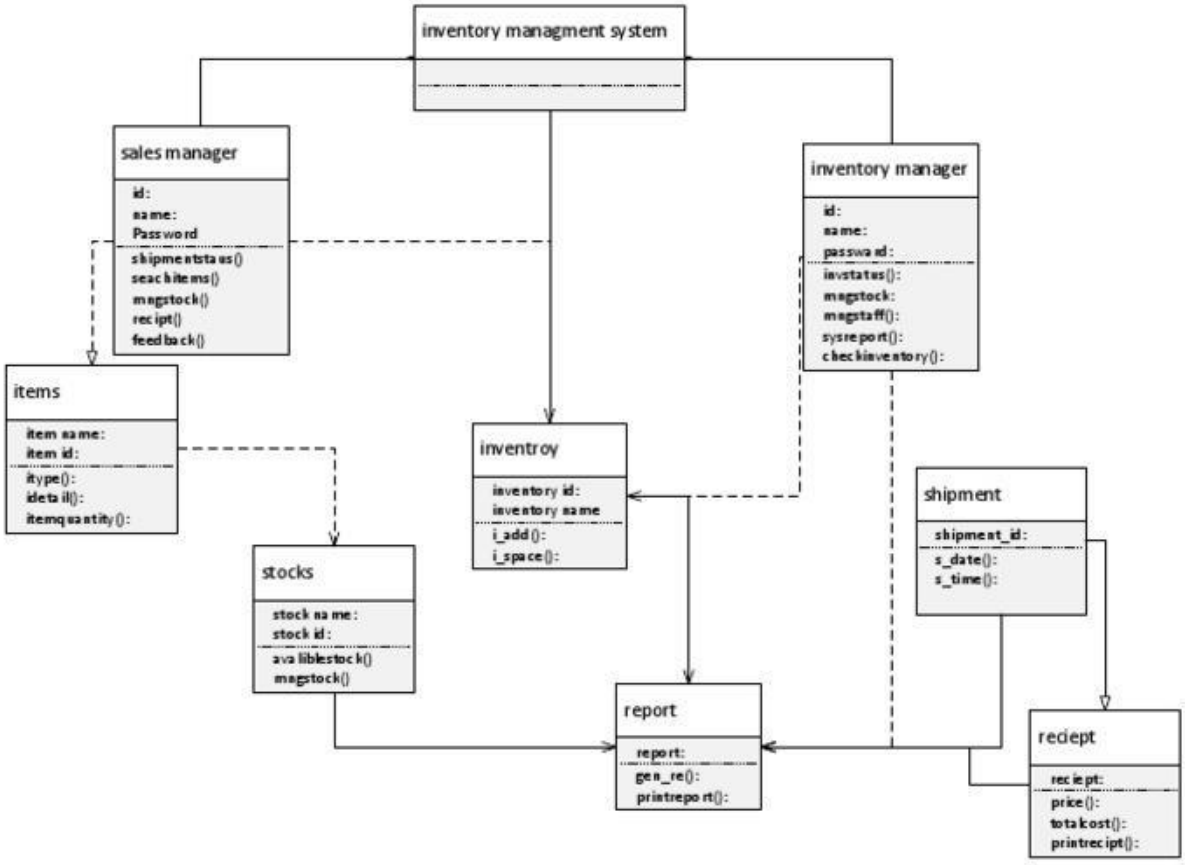
- Customer places order
- Check availability
- Process order
- Update stock
- Deliver product

---

3. CLASS DIAGRAM


Users
id: int
+ name: string
+ age: string
+ contactnum: string
+ add()
+ update()
Stocks
id: int
+ name: string
+ number: string
+ details: string
+ addStock()
+ updateStocks()
Sales
id: int
+ details: string
+ date: string
+ transaction: string
+ create()
+ update()
Pays
Customer
+ address: string
+ transaction: string
+details: string
+ createTransaction()
+ updateTransaction()
Manage
Admin
- username: string
- password: string
Transaction
id: int
+ name: string
+ details: string
+ date: string
- confirmation: stringg
+ create()
+ update()
Creates
Contacts
Products
id: int
+ name: string
+ price: string
+ details: string
+ create()
+ update()
Suppliers
id: int
+ name: string
+ address: string
+ contacts: string
+ create()
+ update()


InventoryManagementApp
+addProduct()
+removeProduct()
+updateProduct()
+listProducts()
+checkStock()
manages
interacts
processes
Product
-String name
-String category
-int quantity
-double price
Supplier
-String name
-String contactDetails
+supplyProduct()
Order
-Date orderDate
-String status
+placeOrder()



CLASS DIAGRAM OF INVENTORY MANAGEMENT SYSTEM

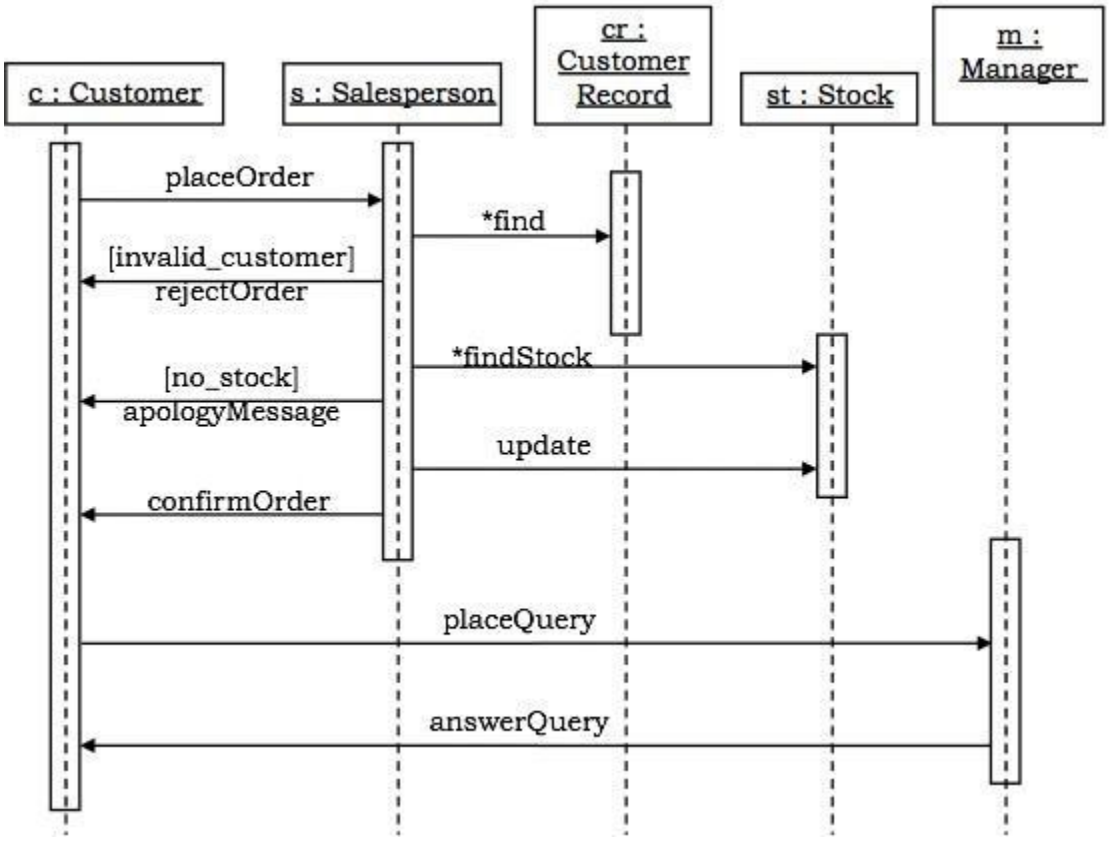
7

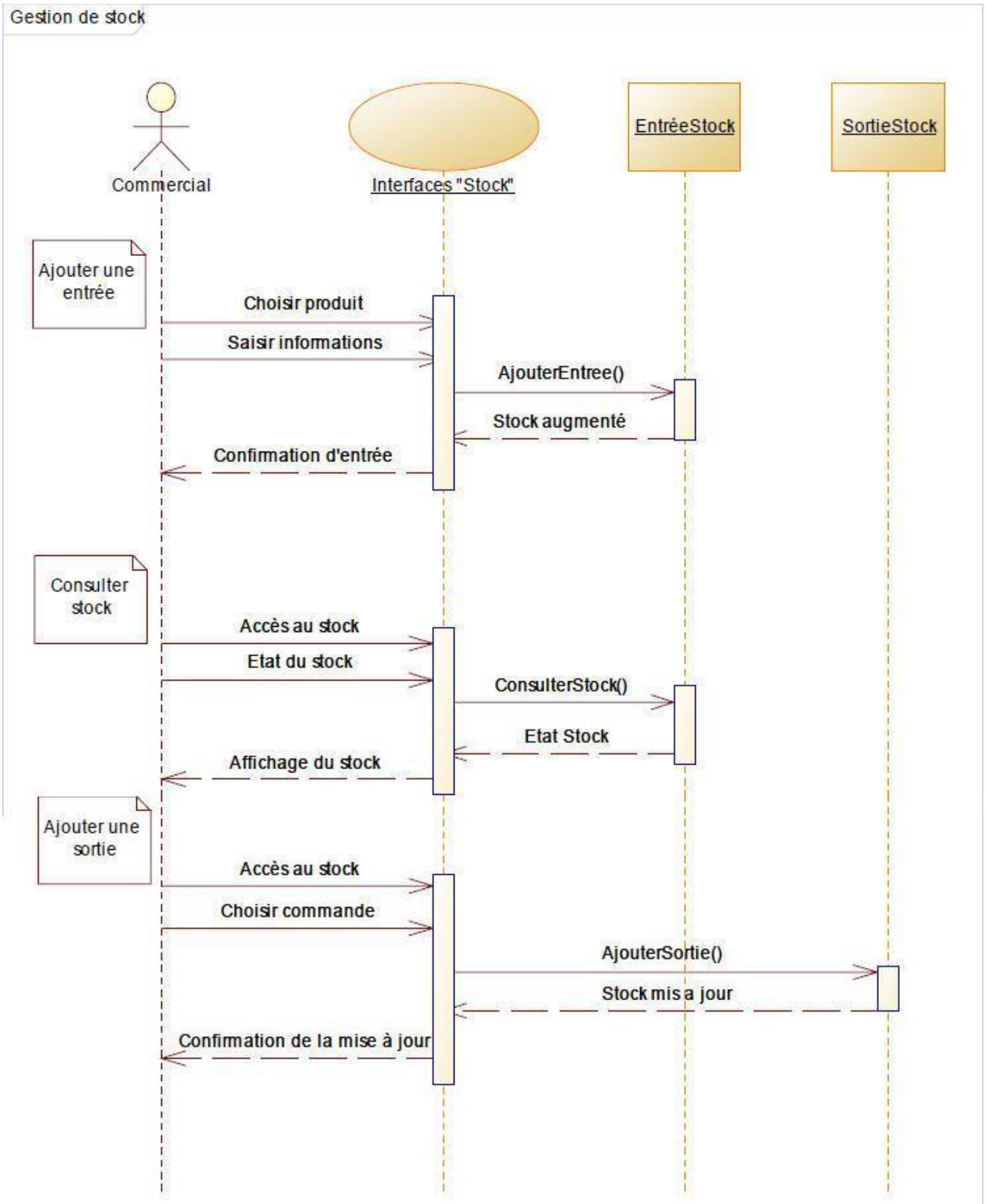
Classes

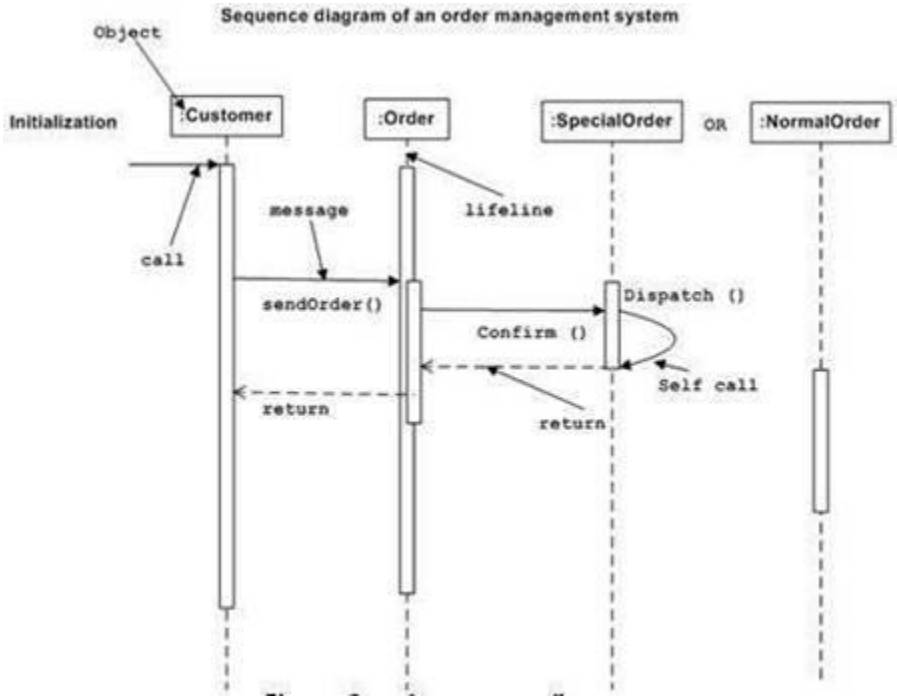
1. Product Details
2. Purchase Details
3. Sales Details
4. Stock Details
5. Customer
6. Supplier/Company

---

4. SEQUENCE DIAGRAM


c : Customer
s : Salesperson
cr : Customer Record
st : Stock
m : Manager
placeOrder
*find
[invalid_customer]
rejectOrder
*findStock
[no_stock]
apologyMessage
update
confirmOrder
placeQuery
answerQuery


Gestion de stock
Commercial
Interfaces "Stock"
EntréeStock
SortieStock
Ajouter une entrée
Choisir produit
Saisir informations
AjouterEntree()
Stock augmenté
Confirmation d'entrée
Consulter stock
Accès au stock
Etat du stock
ConsulterStock()
Etat Stock
Affichage du stock
Ajouter une sortie
Accès au stock
Choisir commande
AjouterSortie()
Stock mis a jour
Confirmation de la mise à jour



6

Description

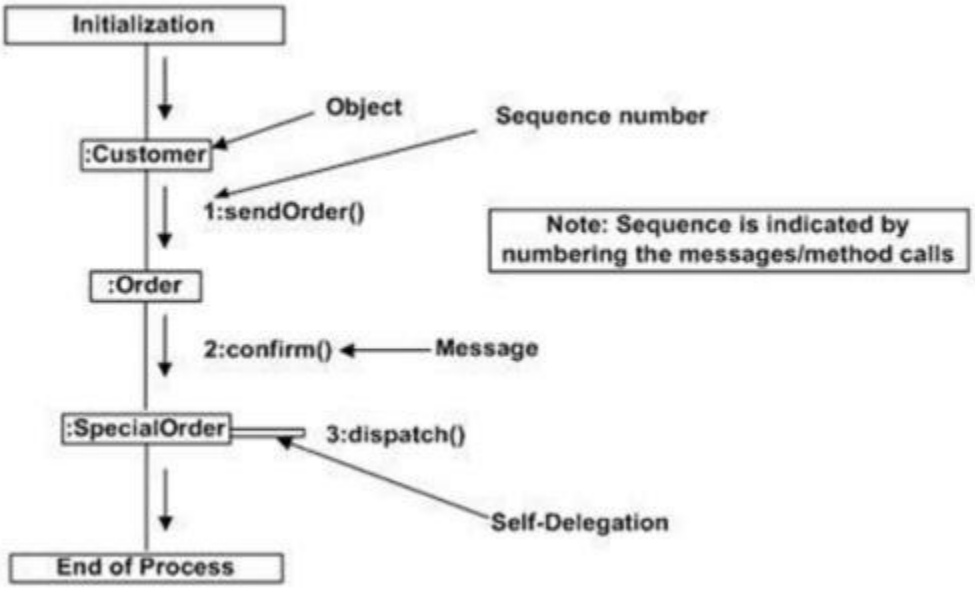
Shows interaction between:

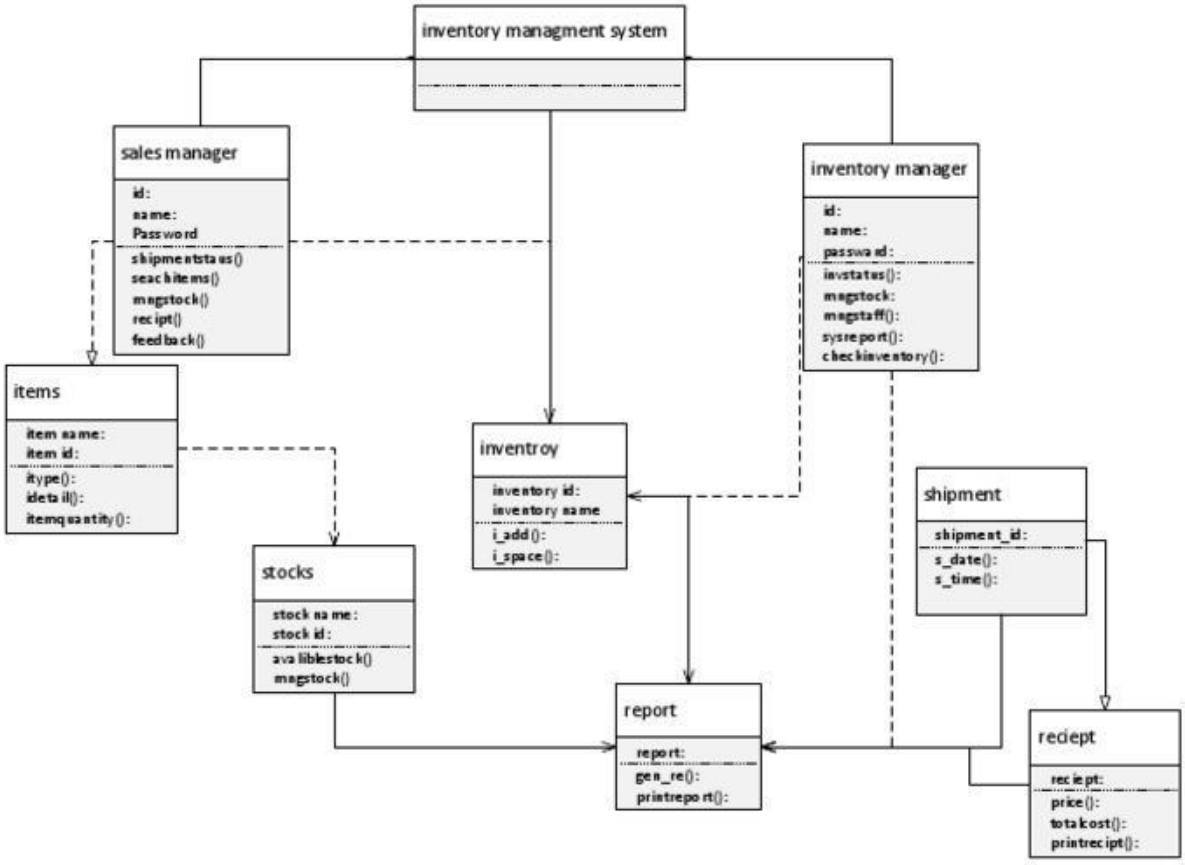
- Customer
- Salesperson
- Database

---

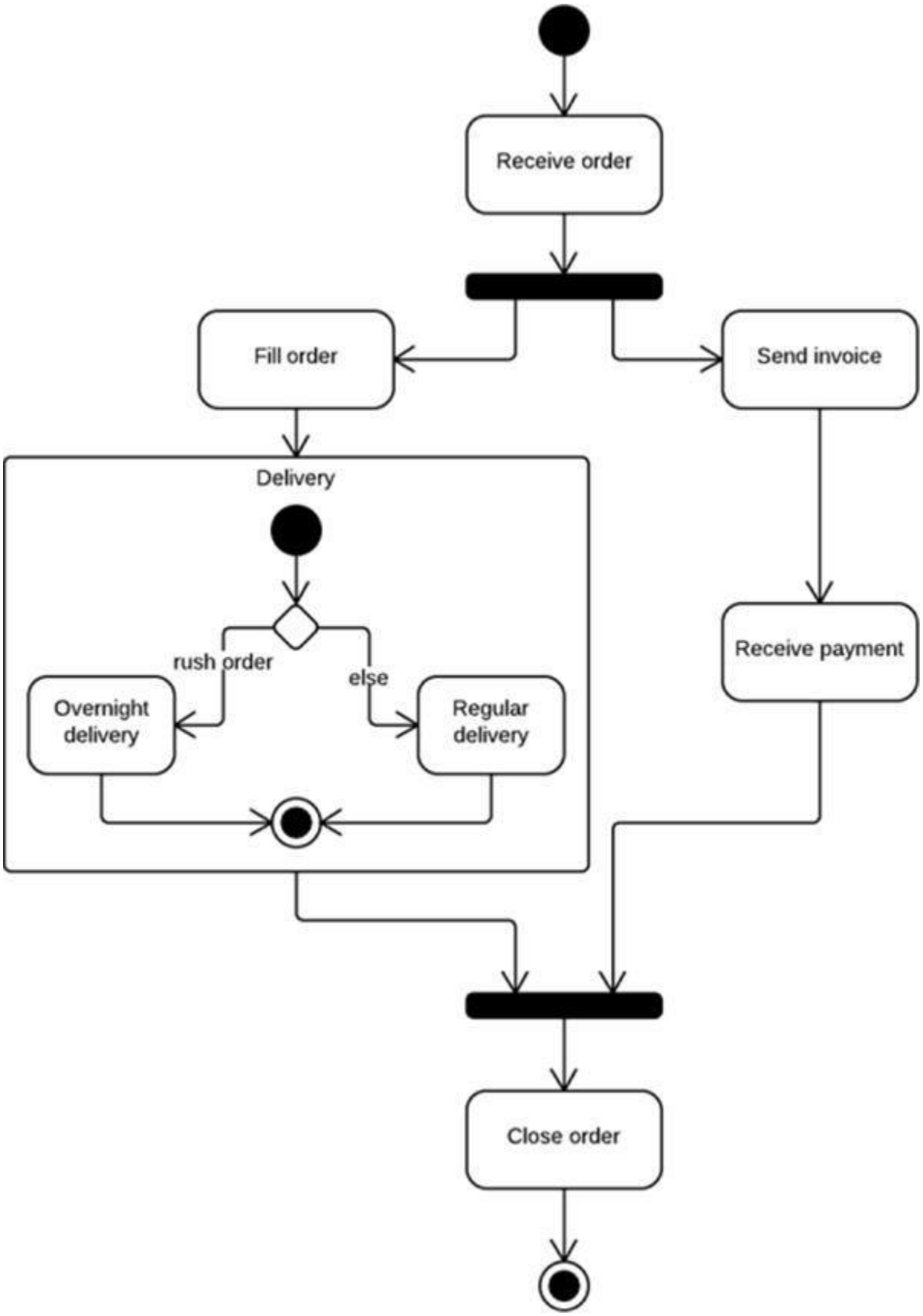
5. COLLABORATION DIAGRAM

Collaboration diagram of an order management system

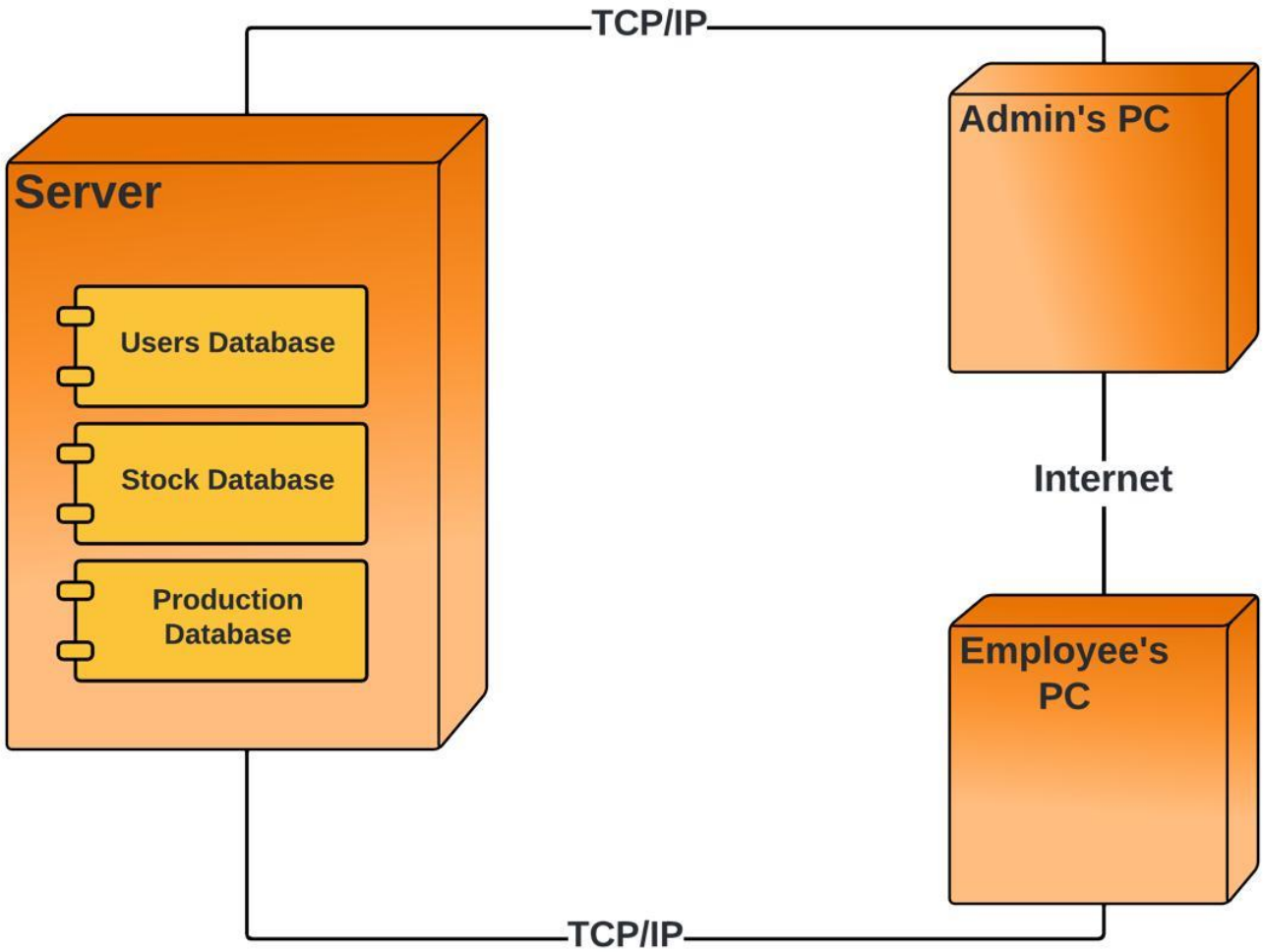


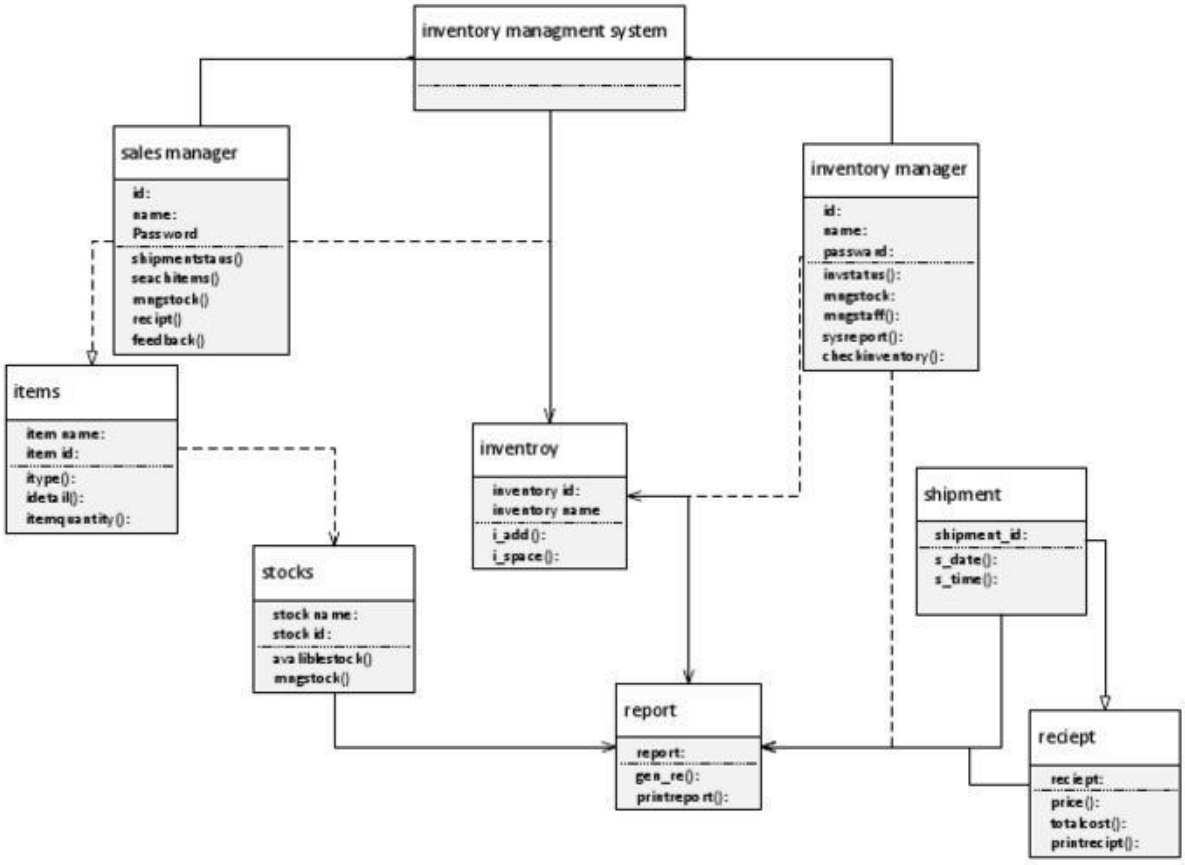


CLASS DIAGRAM OF INVENTORY MANAGEMENT SYSTEM

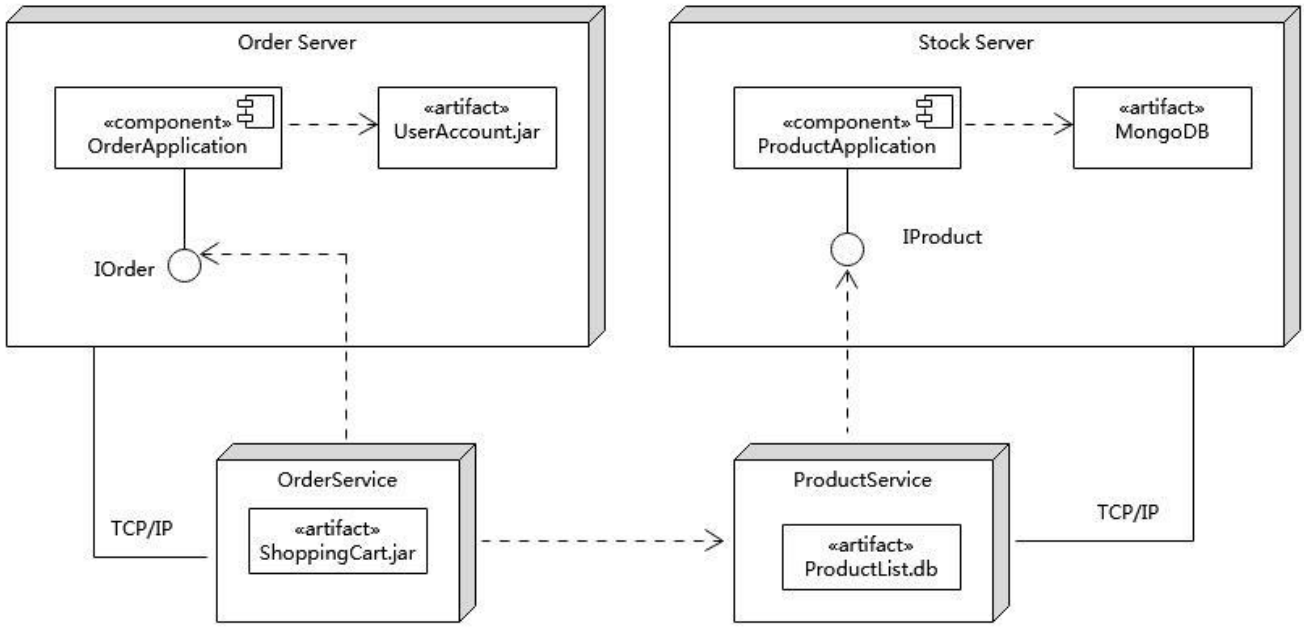

Receive order
Fill order
Send invoice
Delivery
rush order
else
Overnight delivery
Regular delivery
Receive payment
Close order

6. DEPLOYMENT DIAGRAM





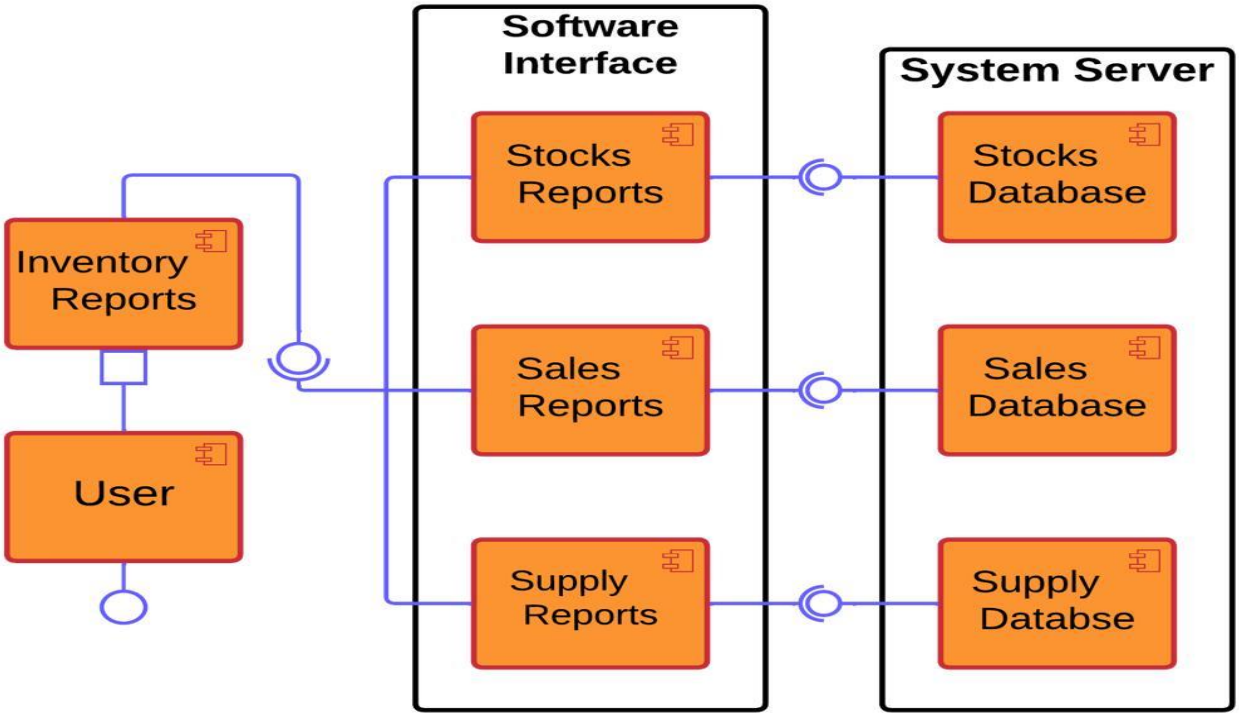
CLASS DIAGRAM OF INVENTORY MANAGEMENT SYSTEM

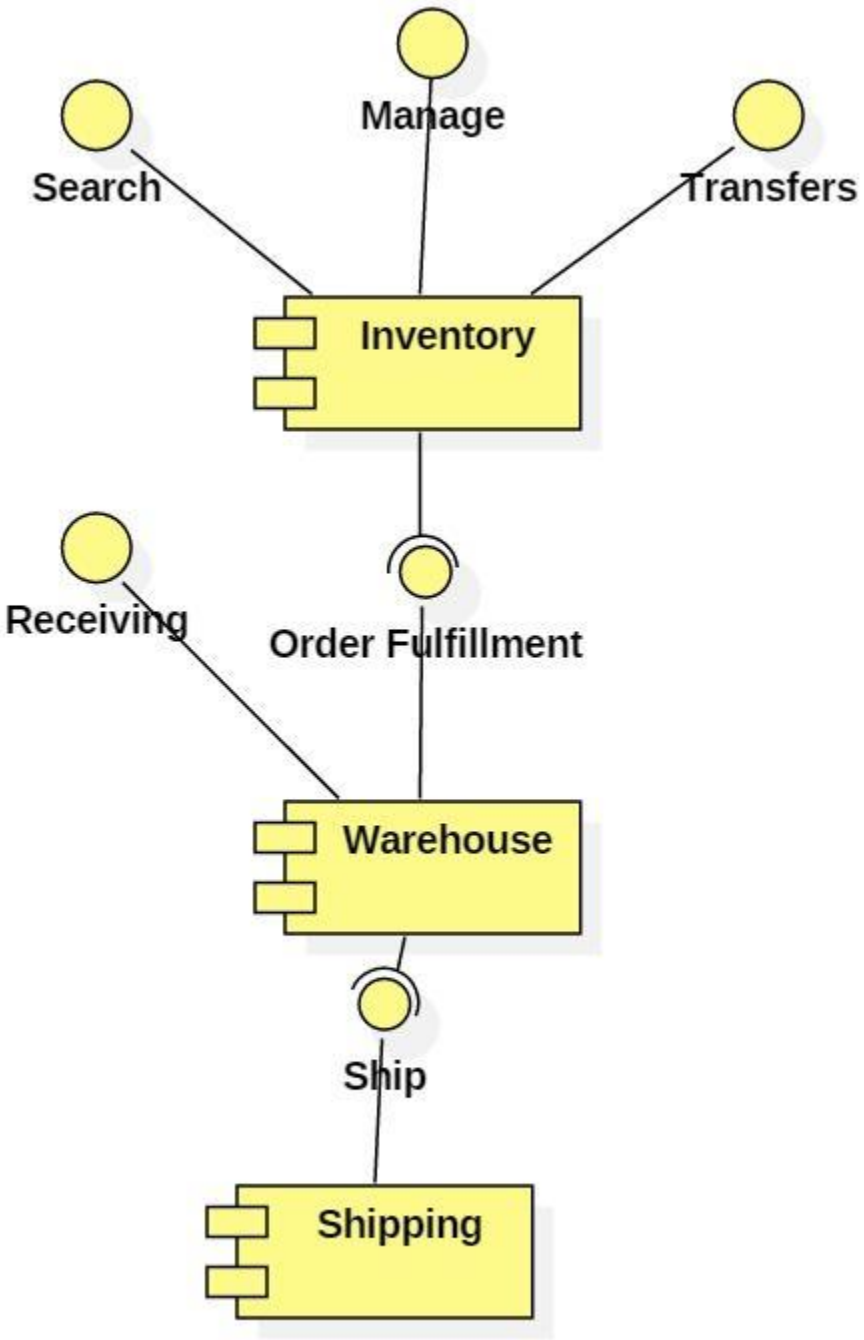


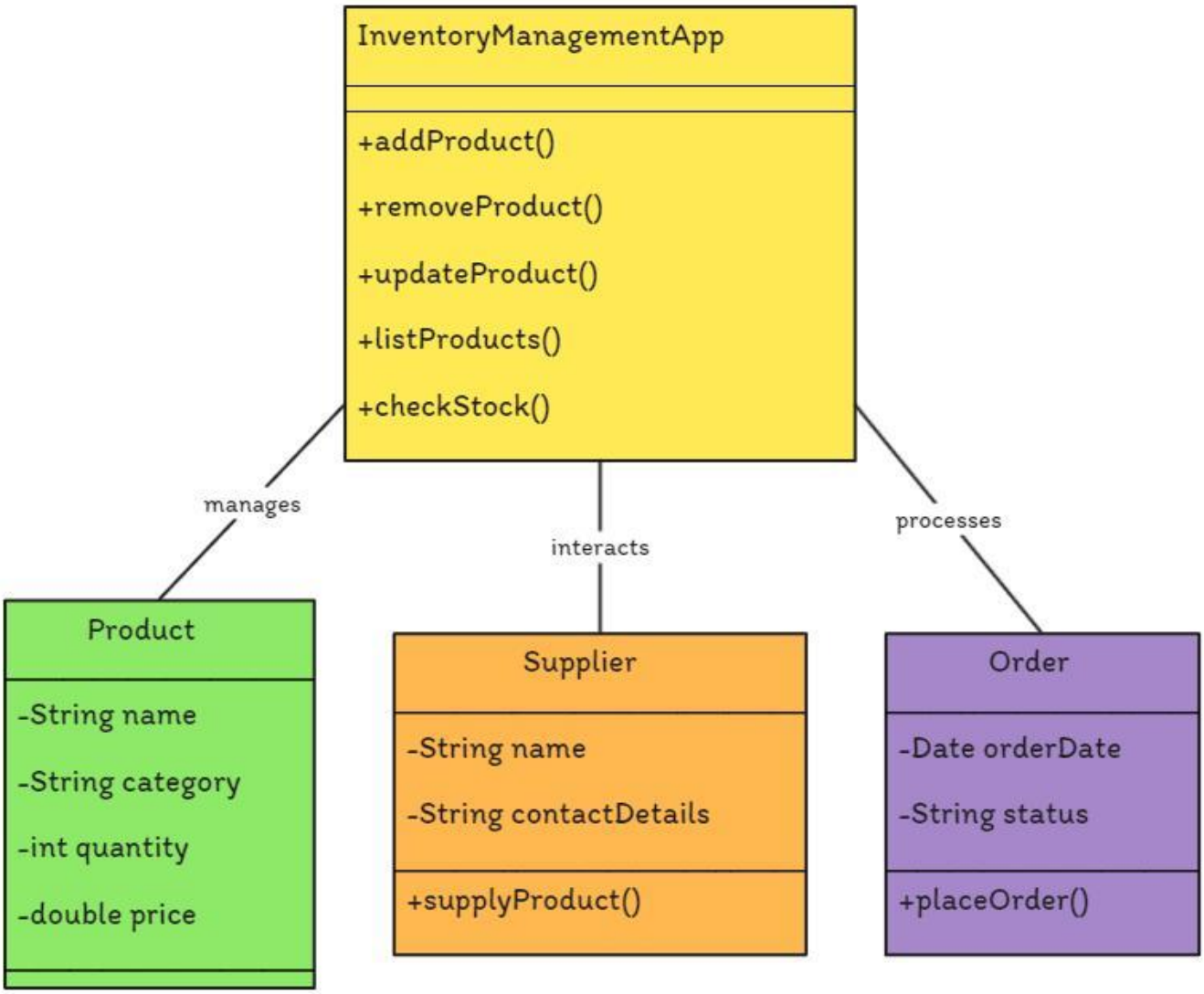
7

---

7. COMPONENT DIAGRAM




Manage
Search
Transfers
Inventory
Receiving
Order Fulfillment
Warehouse
Ship
Shipping



CONCLUSION

The Stock Maintenance System:

- Automates stock tracking
- Reduces manual errors
- Improves efficiency
- Helps in better inventory control

EXPERIMENT–5: ONLINE COURSE RESERVATION SYSTEM

AIM

Study and usage of any Design phase CASE tool

PROCEDURE

(I) PROBLEM STATEMENT

The Online Course Reservation System is designed to allow students/users to browse, select, and reserve courses online.

- Users can register and log in
- Browse available courses
- Reserve or enroll in courses
- Make payments (if required)
- Receive confirmation and notifications

The system reduces manual registration effort and improves accessibility.

(II) SOFTWARE REQUIREMENT SPECIFICATION (SRS)

1. INTRODUCTION

The system acts as an interface between **Users (Students)** and the **Course Provider/Admin**, improving efficiency in course enrollment.

---

2. PURPOSE

Manual course registration is time-consuming. This system:

- Automates course enrollment
- Provides real-time course availability
- Improves user experience

---

3. SCOPE

- User registration and login
- Course browsing and search

- Course reservation/enrollment
- Payment processing
- Notification system

---

4. DEFINITIONS & ACRONYMS

- **User/Student**: Person enrolling in courses
- **Admin**: Manages courses
- **OCRS**: Online Course Reservation System
- **HTML, HTTP, TCP/IP**: Web technologies

---

5. TECHNOLOGIES USED

- HTML
- CSS
- JavaScript
- Java / PHP

TOOLS USED

- Eclipse / VS Code
- Rational Rose (UML Tool)

---

6. OVERVIEW

SRS includes:

- Overall Description
- Specific Requirements

---

7. OVERALL DESCRIPTION

Product Perspective

The system connects users and course providers through a web interface.

Software Interface

- Front End: HTML, CSS, JavaScript

- Back End: Database (MySQL/Oracle)

Hardware Interface

- Client–Server architecture

---

8. SYSTEM FUNCTIONS

- User registration and login
- Course listing
- Enrollment management
- Payment processing
- Notifications

---

9. USER CHARACTERISTICS

- **User**: Enrolls in courses
- **Admin**: Manages courses and users

---

10. CONSTRAINTS

- Requires internet access
- Security must be ensured
- Payment gateway dependency

---

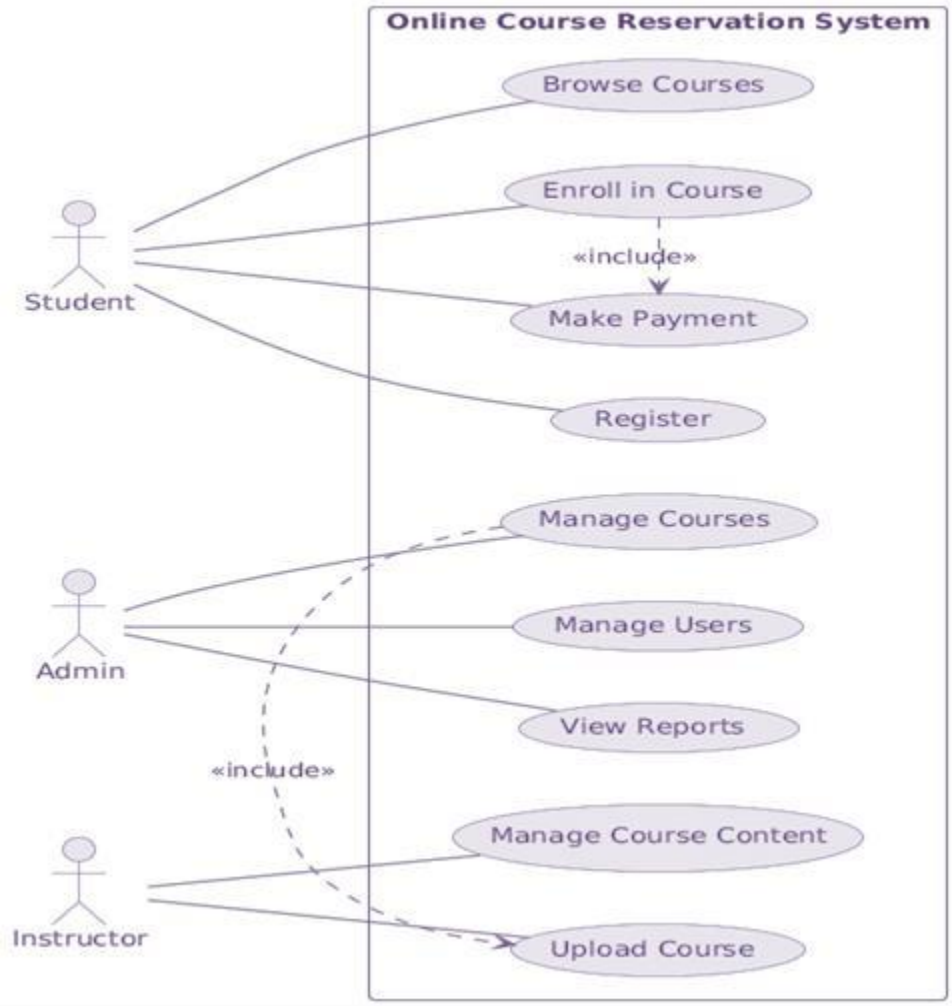
11. ASSUMPTIONS

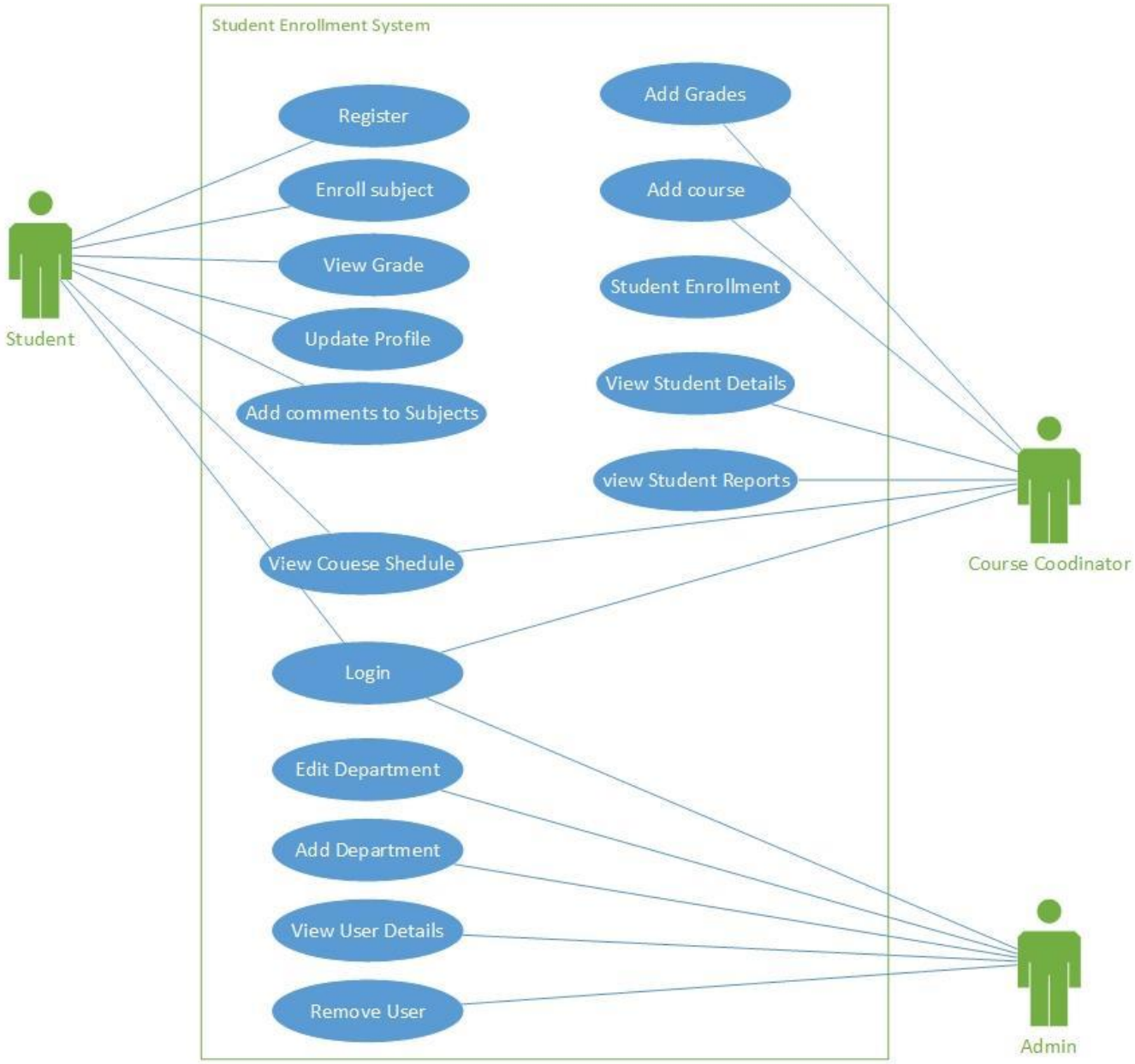
- Users have basic computer knowledge
- Online payment system available

(III) UML DIAGRAMS

---

1. USE CASE DIAGRAM


Online Course Reservation System
Browse Courses
Enroll in Course
«include»
Make Payment
Register
Manage Courses
Manage Users
View Reports
Manage Course Content
Upload Course
Student
Admin
Instructor
«include»



6

Actors

- Student/User
- Admin

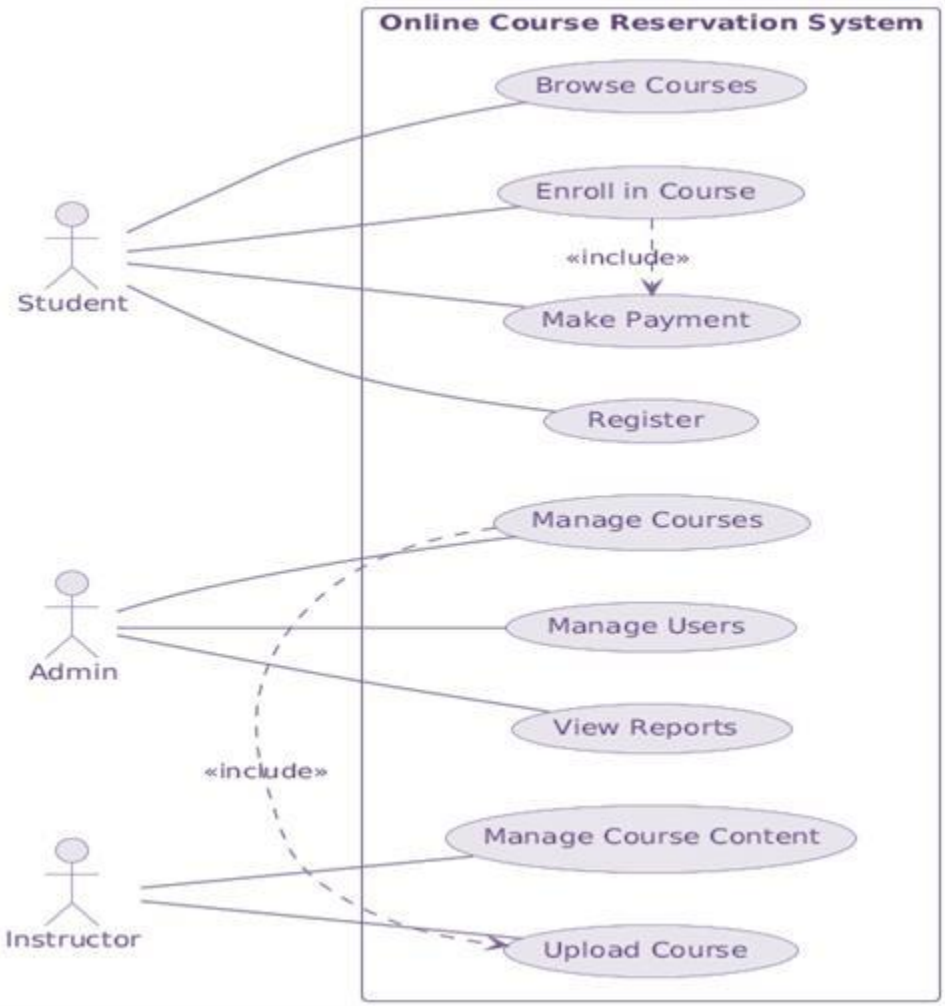
Use Cases

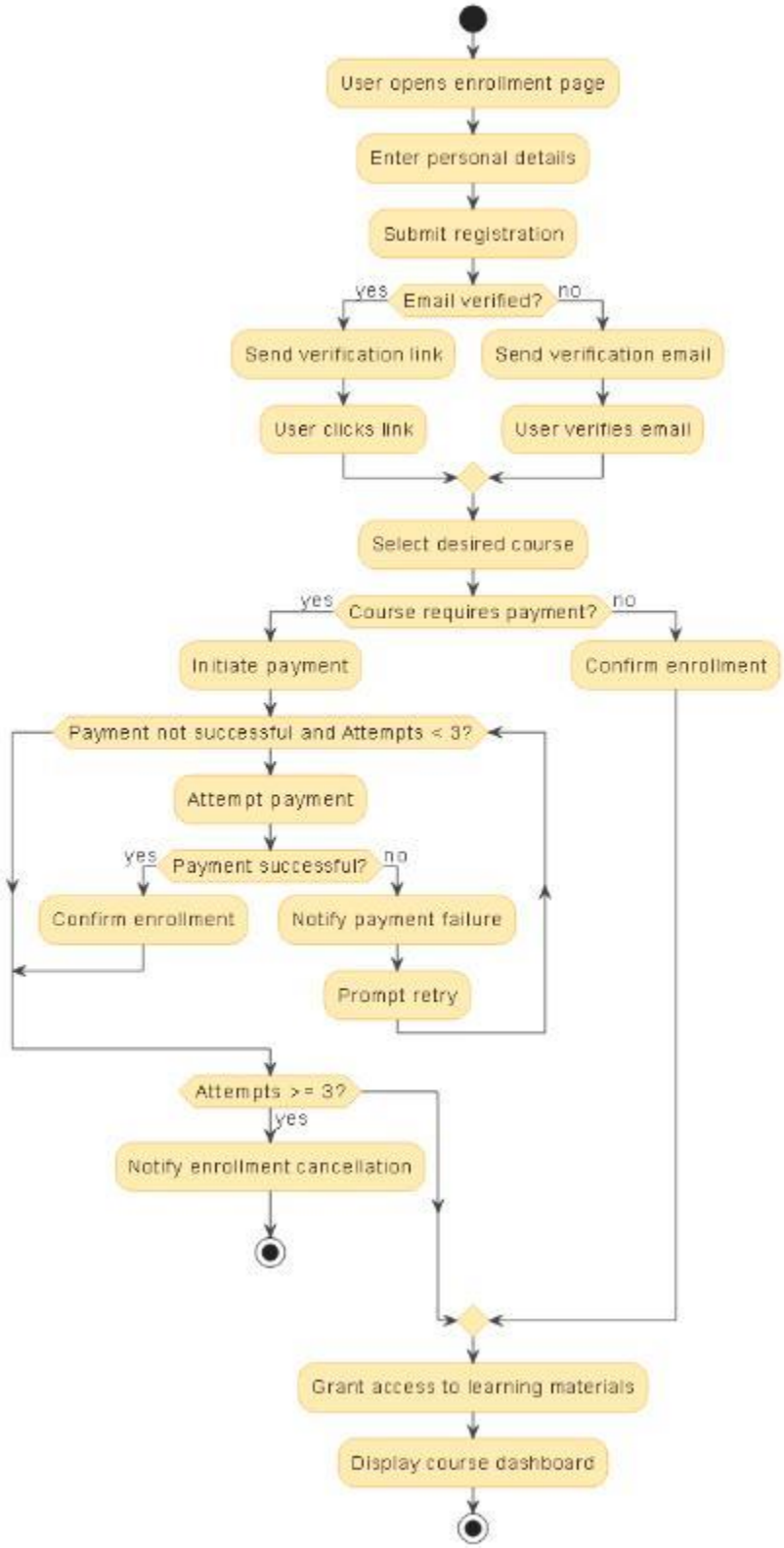
- Login
- Register
- Browse Courses

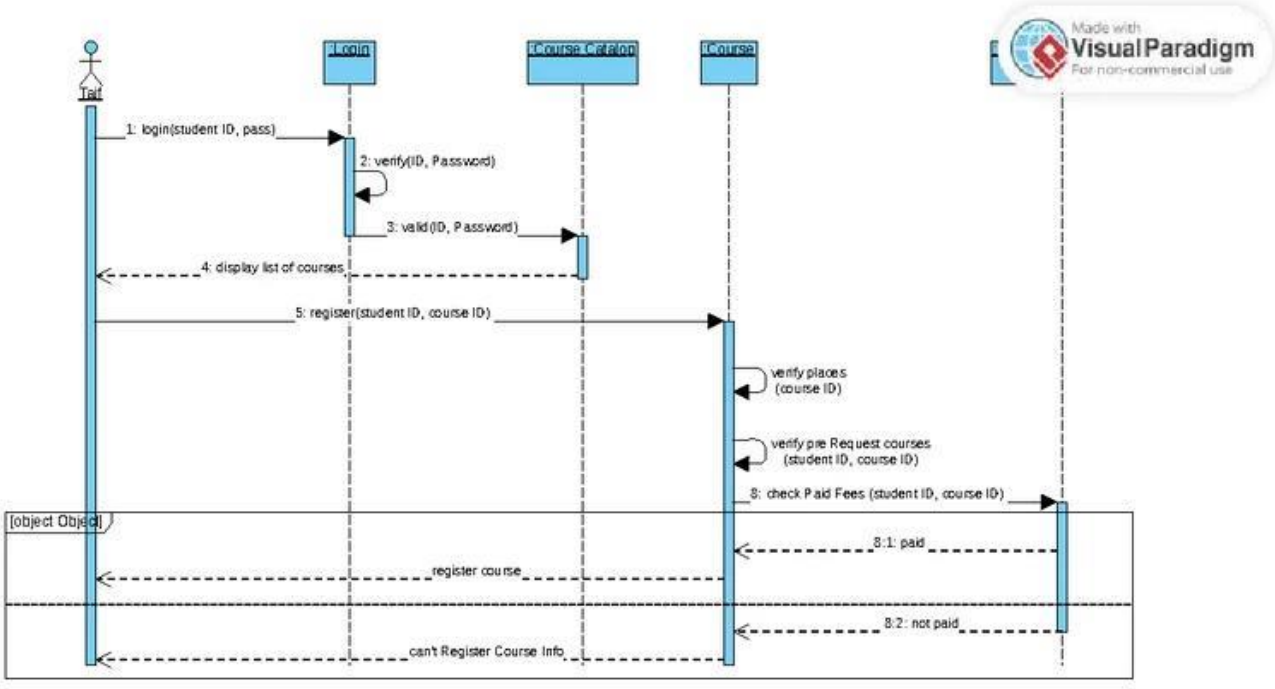
- Enroll Course
- Payment
- View Status

---

2. ACTIVITY DIAGRAM




User opens enrollment page
Enter personal details
Submit registration
Email verified?
yes
no
Send verification link
Send verification email
User clicks link
User verifies email
Select desired course
Course requires payment?
yes
no
Initiate payment
Confirm enrollment
Payment not successful and Attempts < 3?
Attempt payment
Payment successful?
yes
no
Confirm enrollment
Notify payment failure
Prompt retry
Attempts >= 3?
yes
Notify enrollment cancellation
Grant access to learning materials
Display course dashboard



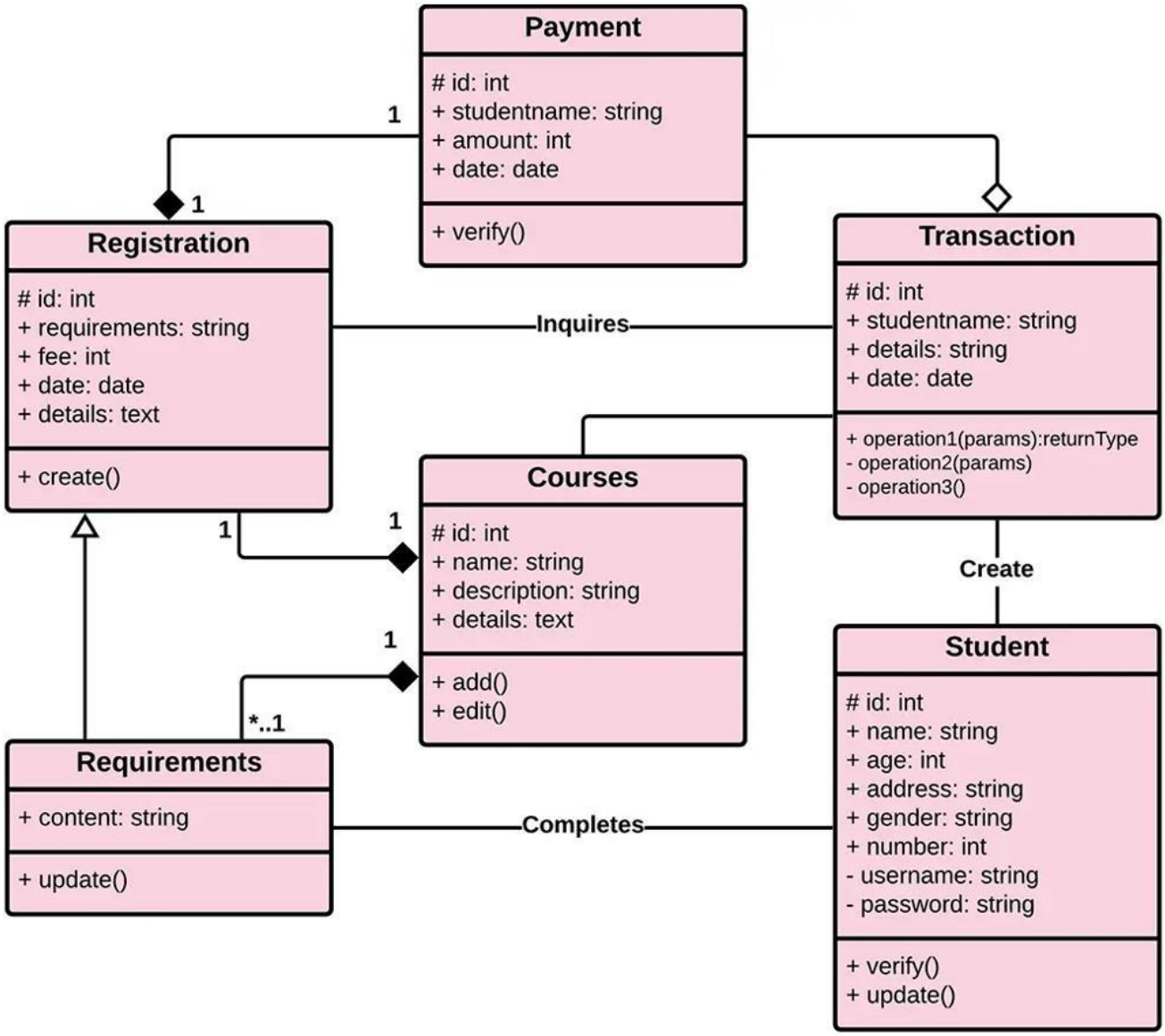
6

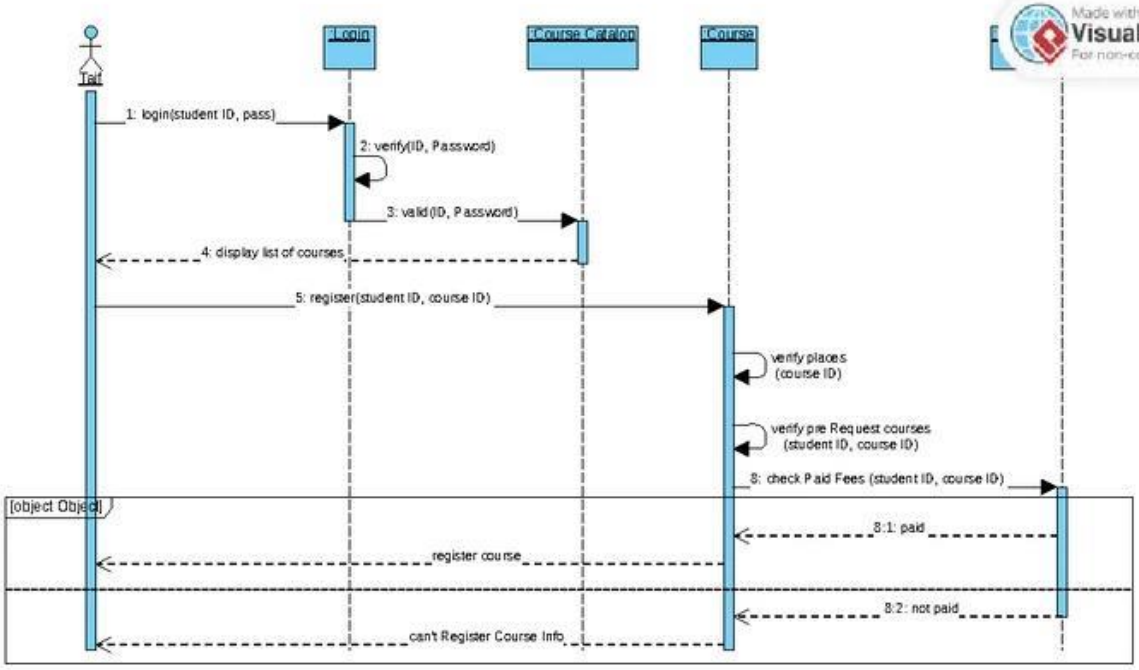
Flow

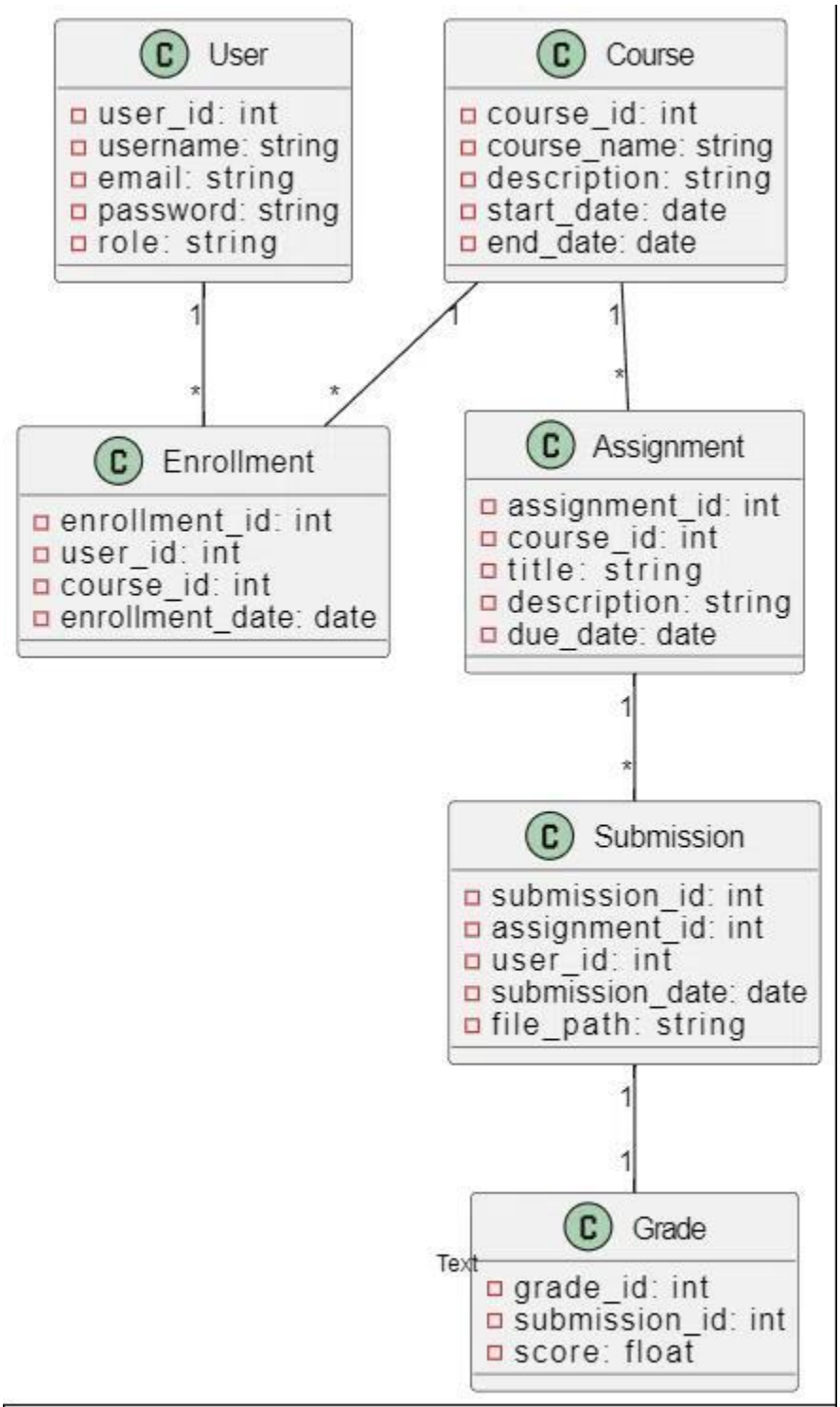
- User login
- Browse courses
- Select course
- Make payment
- Confirm enrollment

---

3. CLASS DIAGRAM


Payment
id: int
+ studentname: string
+ amount: int
+ date: date
+ verify()
1
1
Registration
id: int
+ requirements: string
+ fee: int
+ date: date
+ details: text
+ create()
Inquires
Transaction
id: int
+ studentname: string
+ details: string
+ date: date
+ operation1(params):returnType
- operation2(params)
- operation3()
Courses
id: int
+ name: string
+ description: string
+ details: text
+ add()
+ edit()
1
1
1
1
Create
Student
id: int
+ name: string
+ age: int
+ address: string
+ gender: string
+ number: int
- username: string
- password: string
+ verify()
+ update()
*..1
Requirements
+ content: string
+ update()
Completes


Made with VisualParadigm
For non-commercial use
Tail
:Login
:Course Creation
:Course
1: login(student ID, pass)
2: verify(ID, Password)
3: valid(ID, Password)
4: display list of courses
5: register(student ID, course ID)
verify places (course ID)
verify pre Request courses (student ID, course ID)
8: check Paid Fees (student ID, course ID)
[object Object]
8:1: paid
register course
8:2: not paid
can't Register Course Info



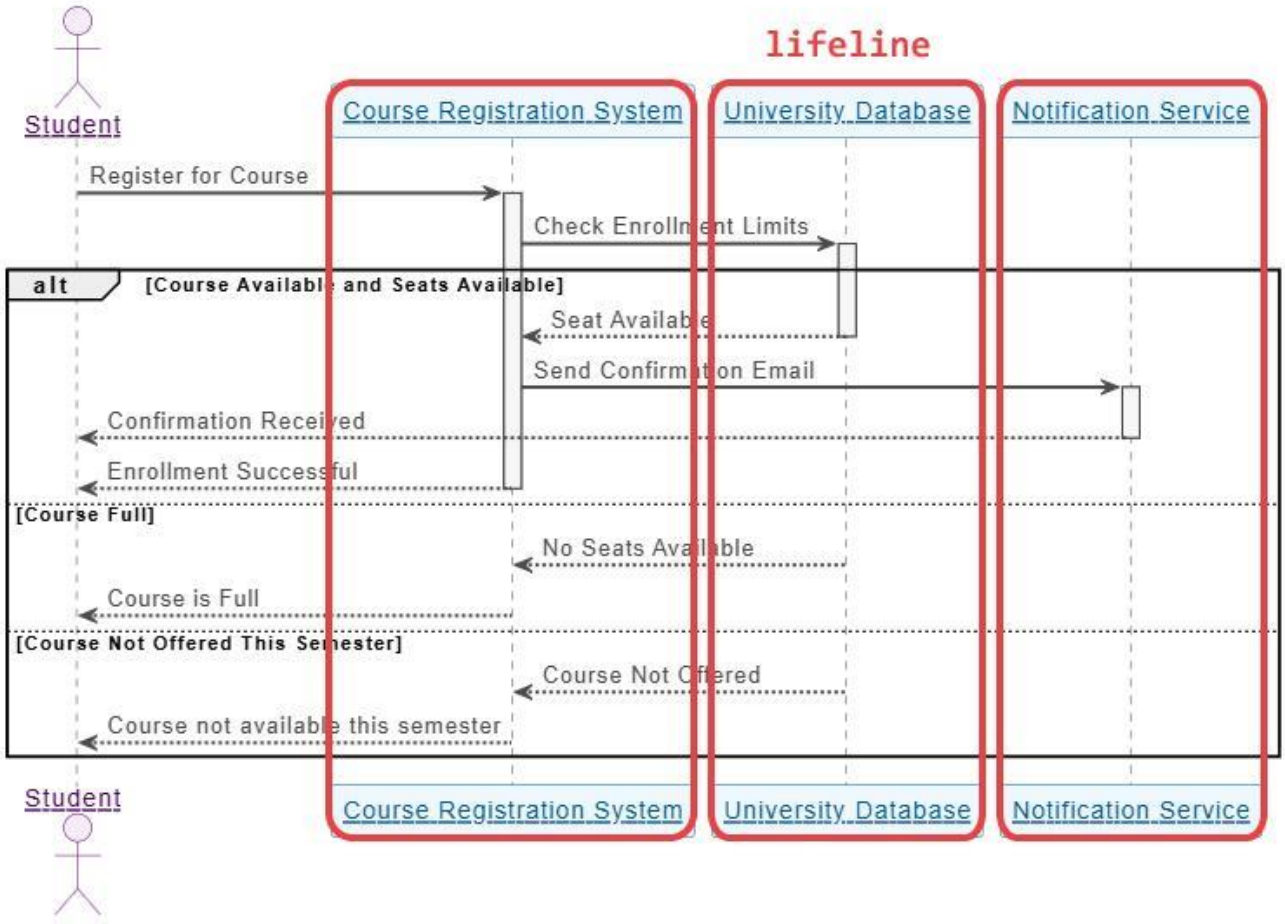
Class Diagram of Learning management System

Classes

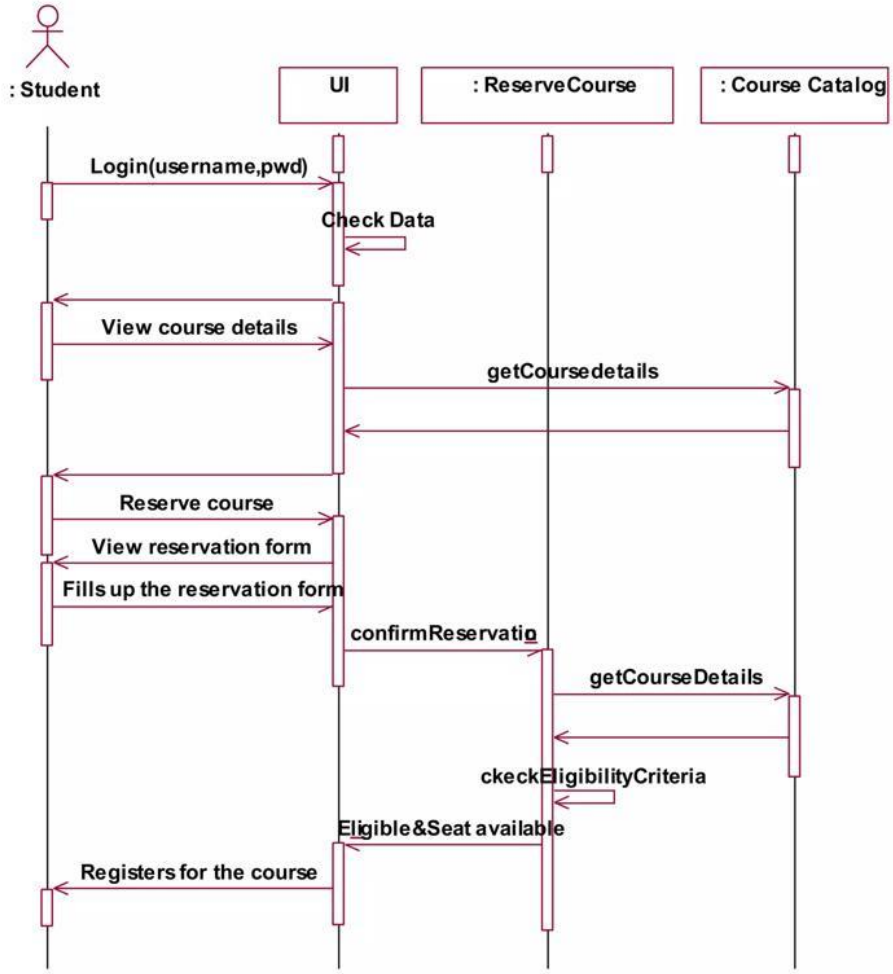
1. User
2. Course
3. Enrollment
4. Payment
5. Admin

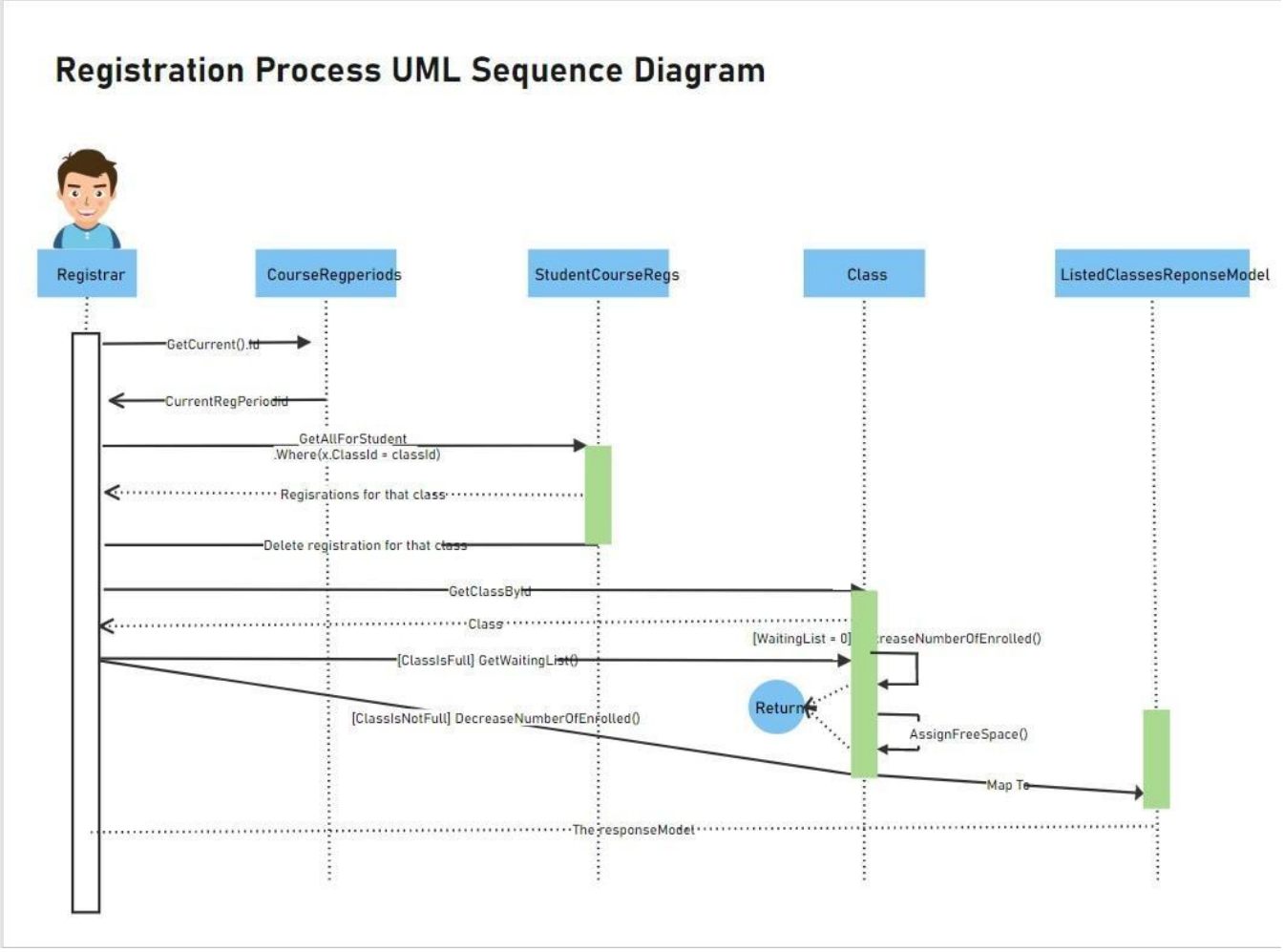
---

4. SEQUENCE DIAGRAM



Sequence diagram:





6

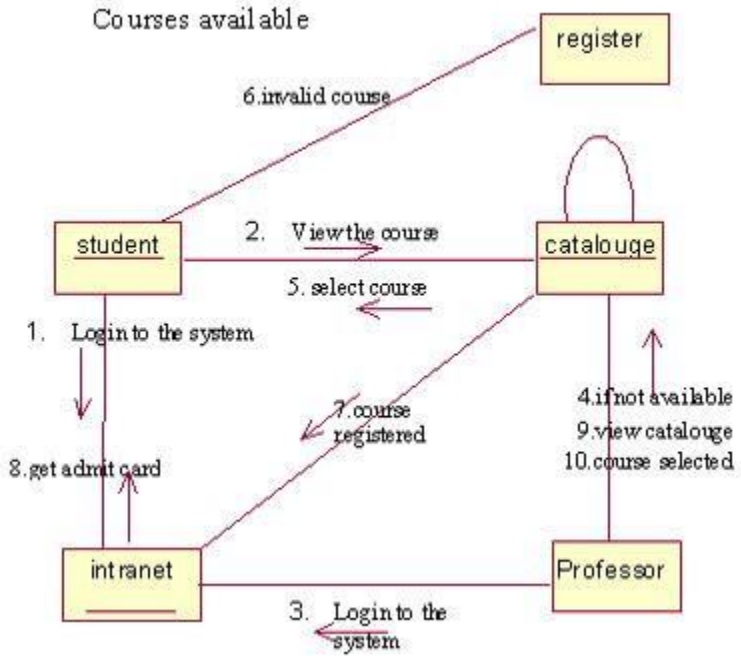
Description

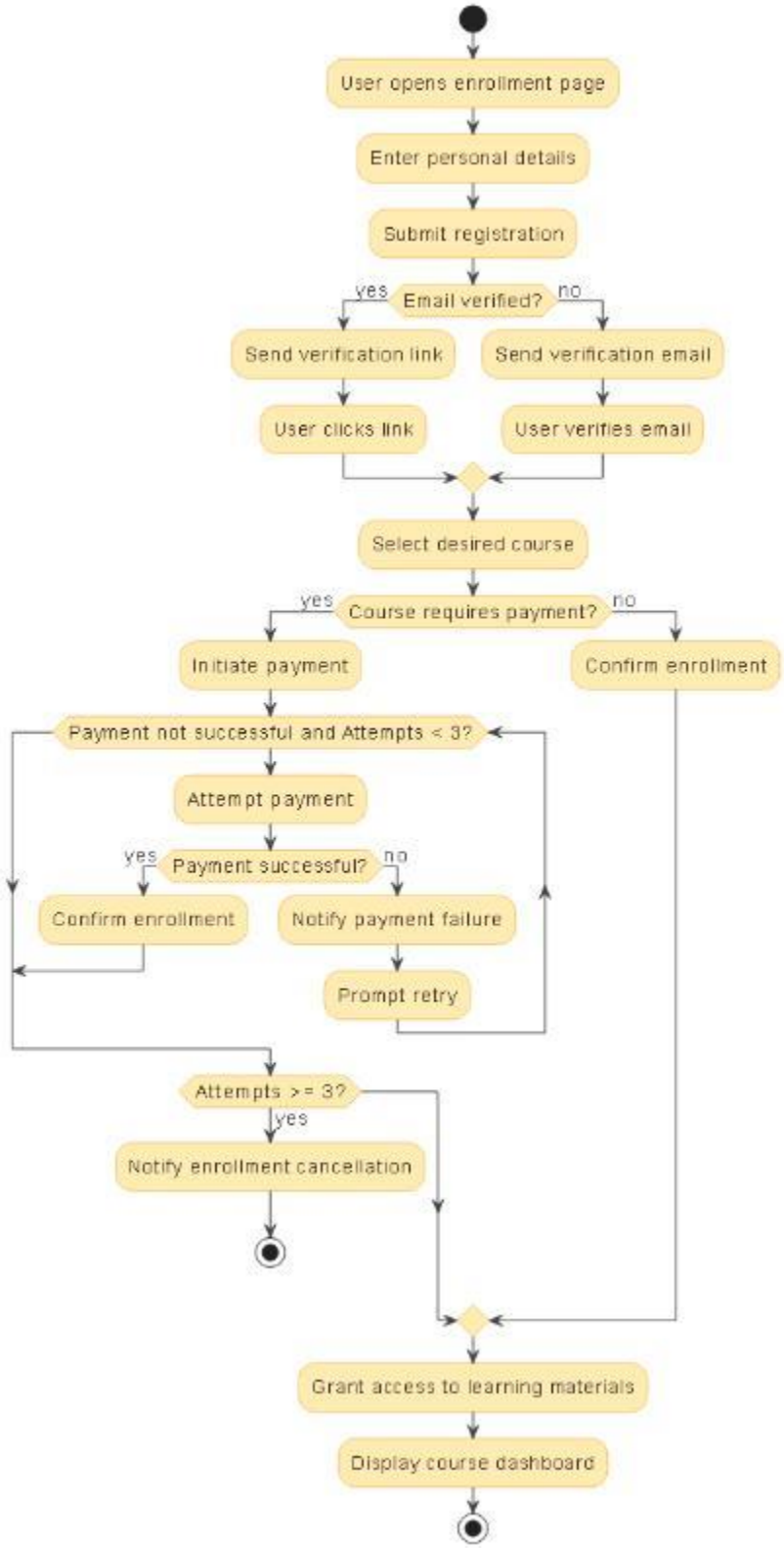
Shows interaction between:

- User
- System
- Database

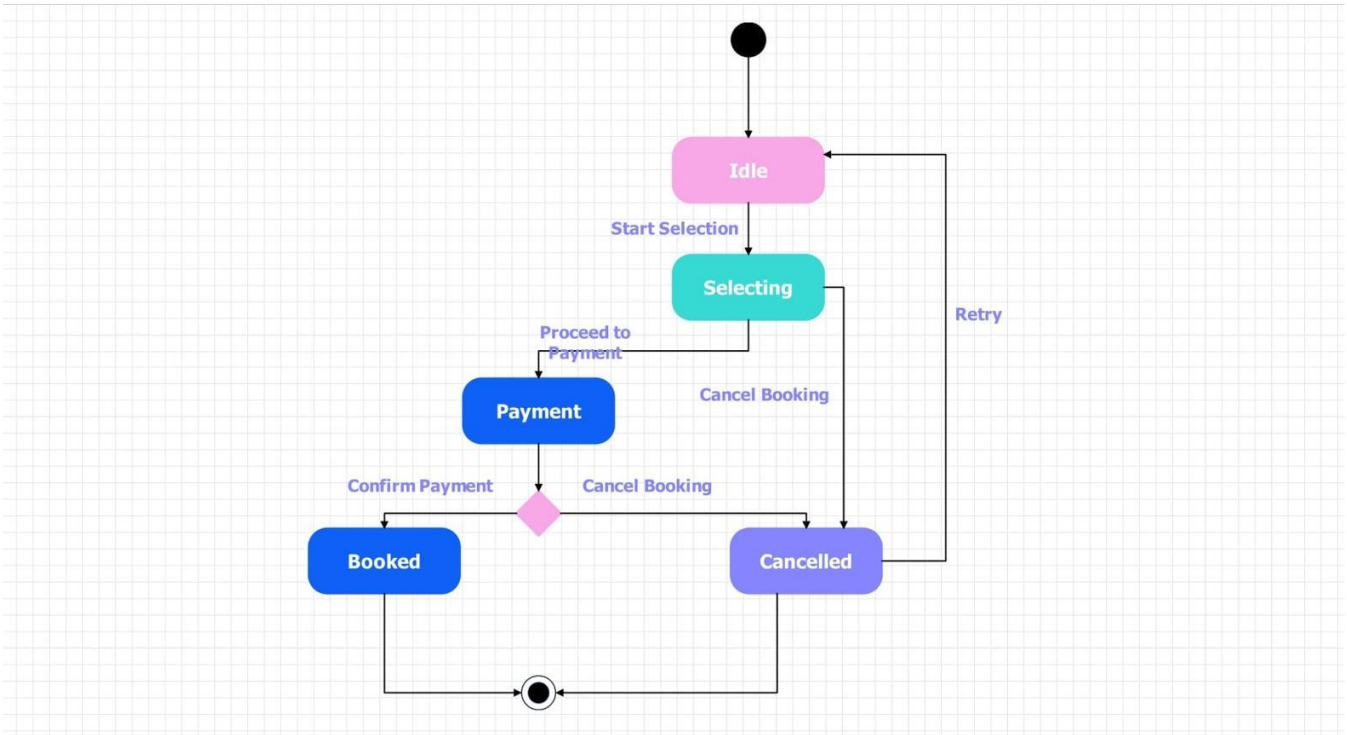
---

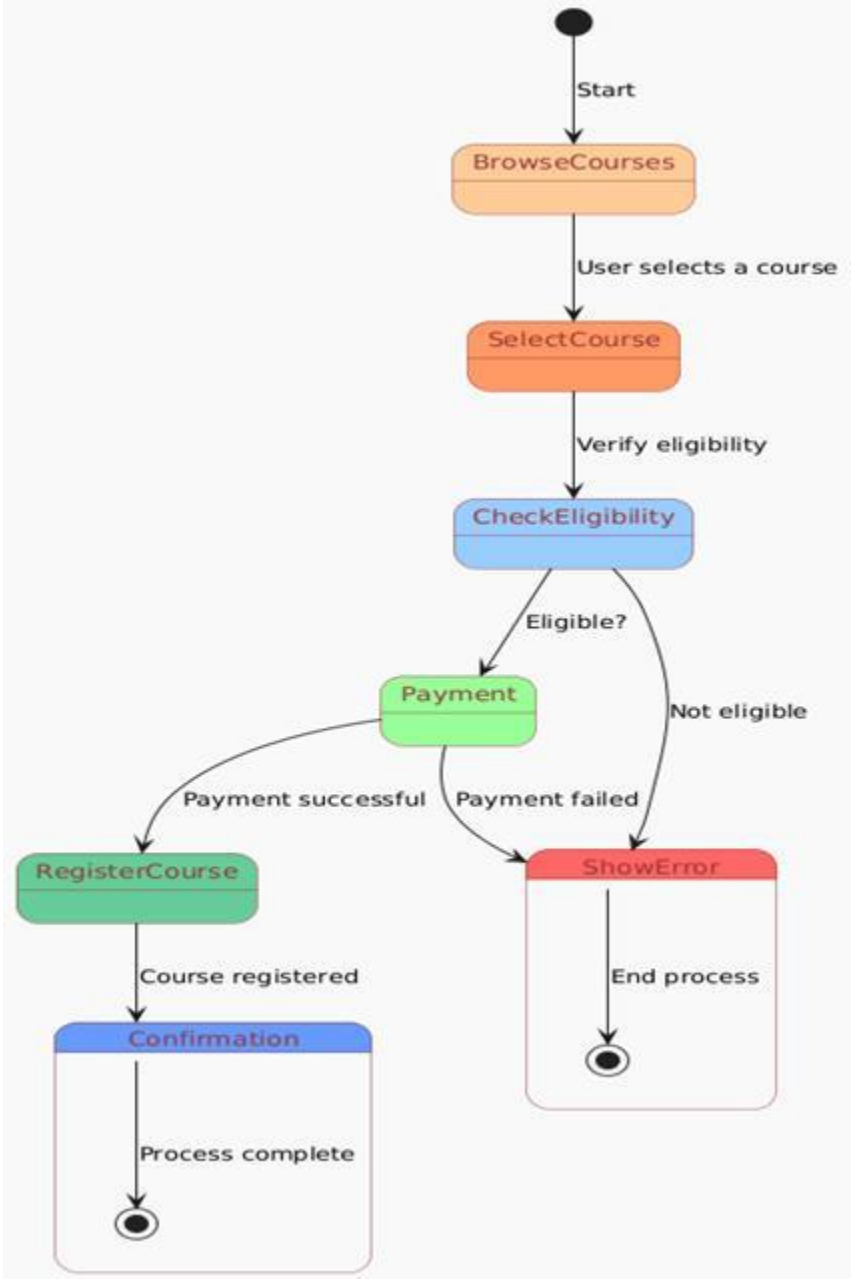
5. COLLABORATION DIAGRAM

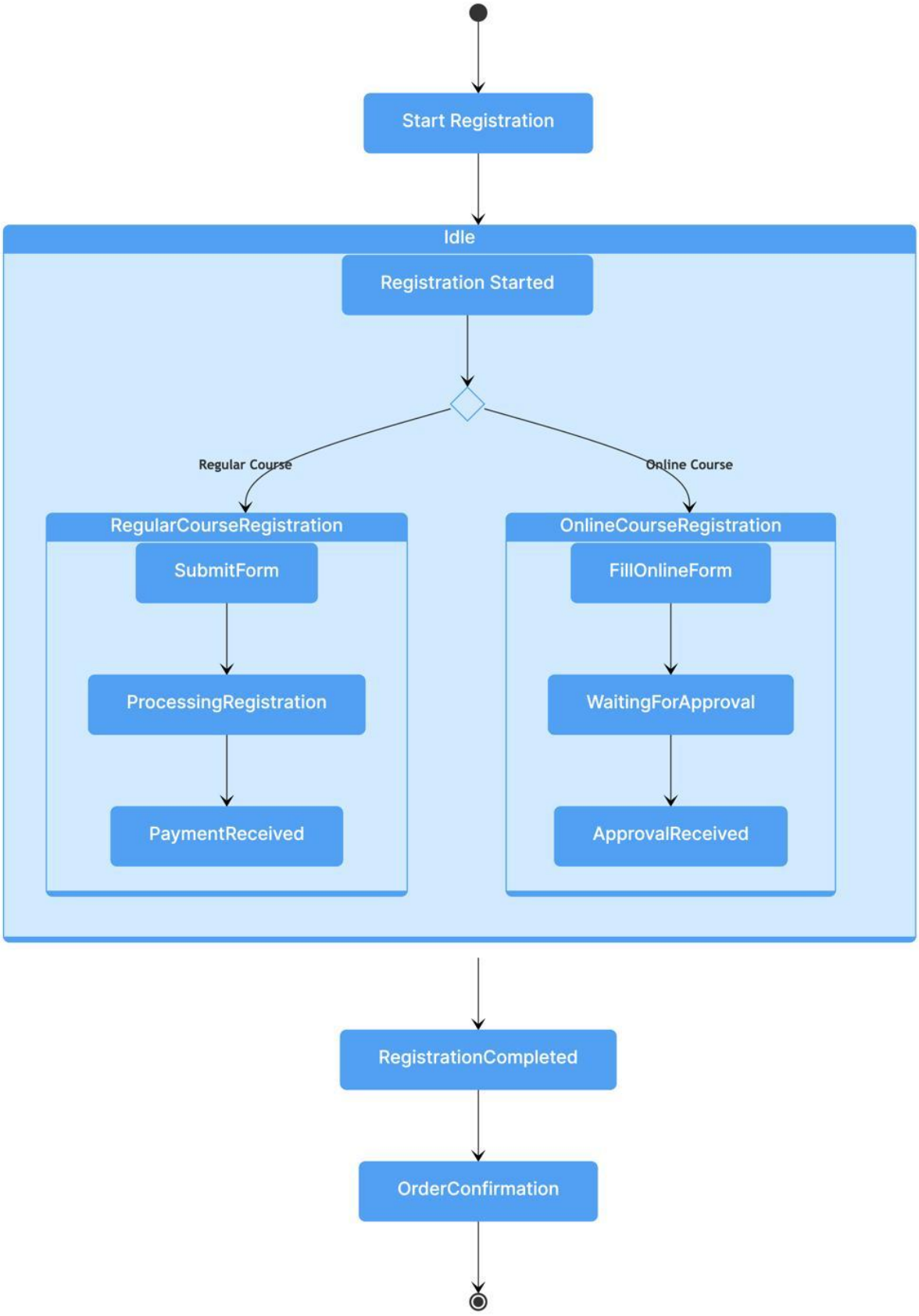

Courses available
register
6.invalid course
student
2. View the course
catalouge
5. select course
1. Login to the system
7.course registered
4.if not available
9.view catalouge
10.course selected
8.get admit card
intranet
Professor
3. Login to the system


User opens enrollment page
Enter personal details
Submit registration
Email verified?
yes
no
Send verification link
Send verification email
User clicks link
User verifies email
Select desired course
Course requires payment?
yes
no
Initiate payment
Confirm enrollment
Payment not successful and Attempts < 3?
Attempt payment
Payment successful?
yes
no
Confirm enrollment
Notify payment failure
Prompt retry
Attempts >= 3?
yes
Notify enrollment cancellation
Grant access to learning materials
Display course dashboard

6. STATECHART DIAGRAM




Start
BrowseCourses
User selects a course
SelectCourse
Verify eligibility
CheckEligibility
Eligible?
Not eligible
Payment
Payment successful
Payment failed
RegisterCourse
ShowError
Course registered
End process
Confirmation
Process complete


Start Registration
Idle
Registration Started
Regular Course
Online Course
RegularCourseRegistration
SubmitForm
ProcessingRegistration
PaymentReceived
OnlineCourseRegistration
FillOnlineForm
WaitingForApproval
ApprovalReceived
RegistrationCompleted
OrderConfirmation

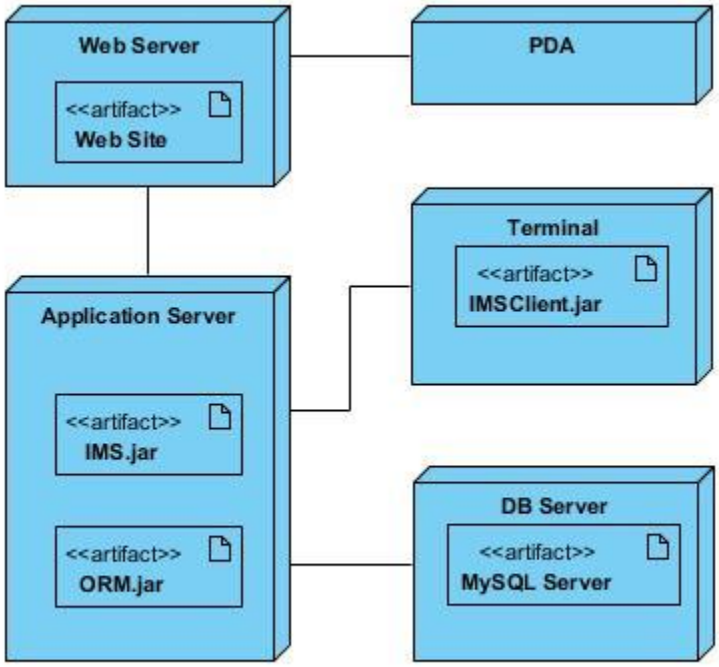
6

States

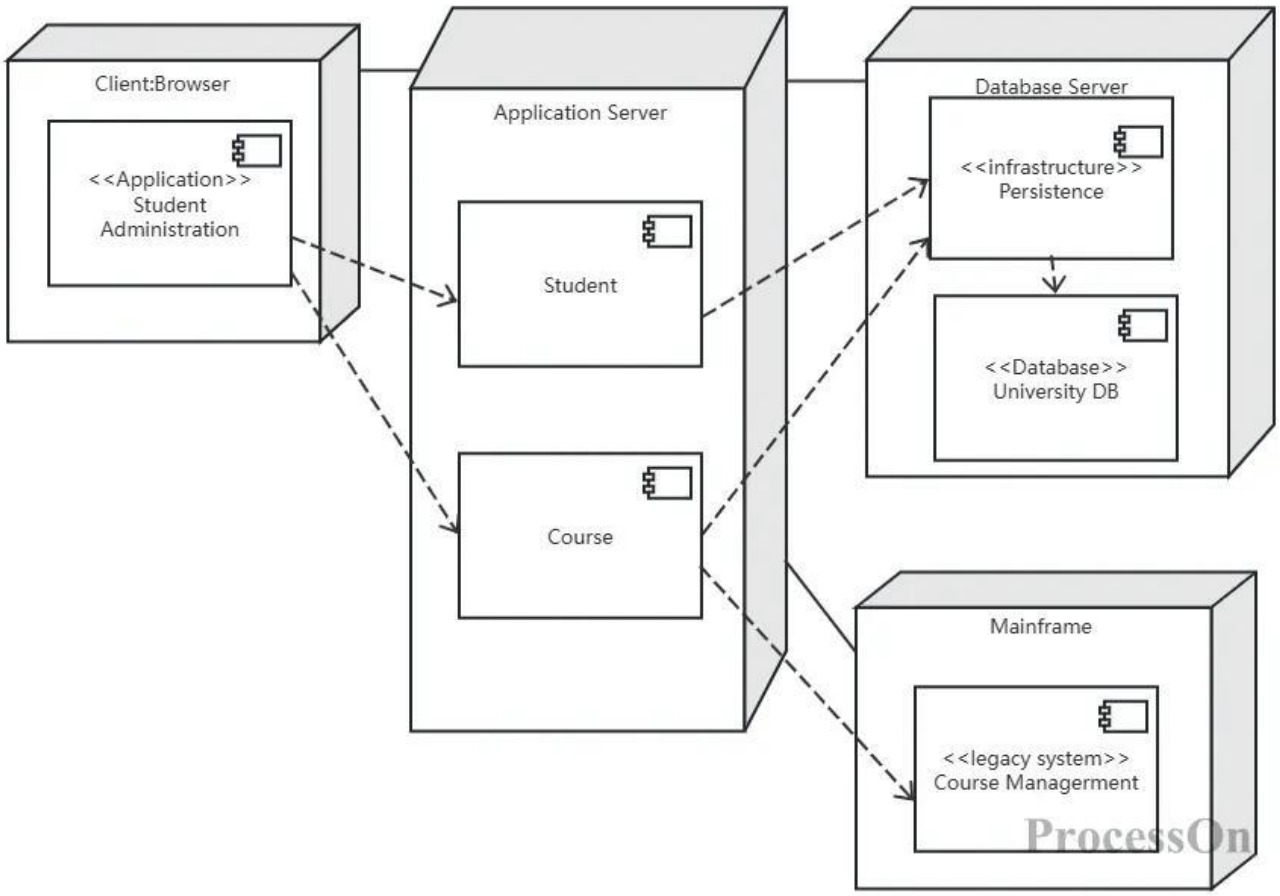
- Registered
- Browsing
- Enrolled
- Payment Pending
- Completed

---

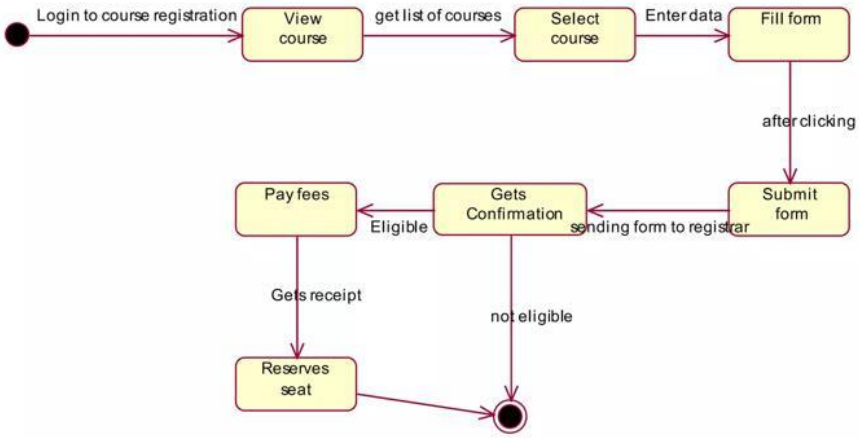
7. DEPLOYMENT DIAGRAM



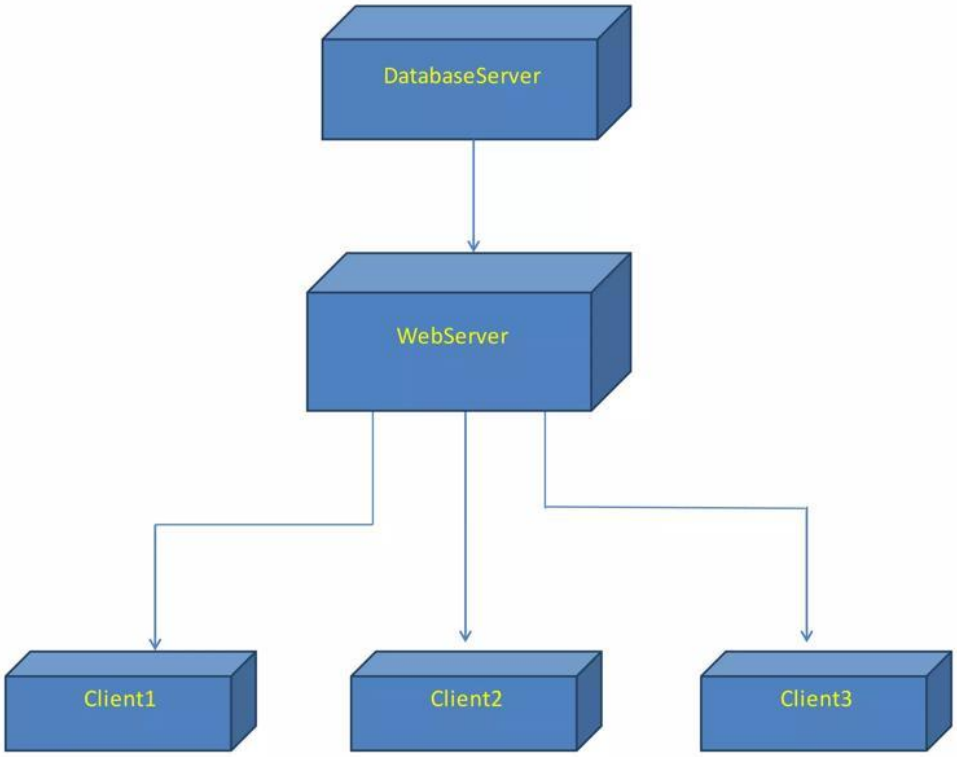
Système de sélection de cours – Diagramme de déploiement



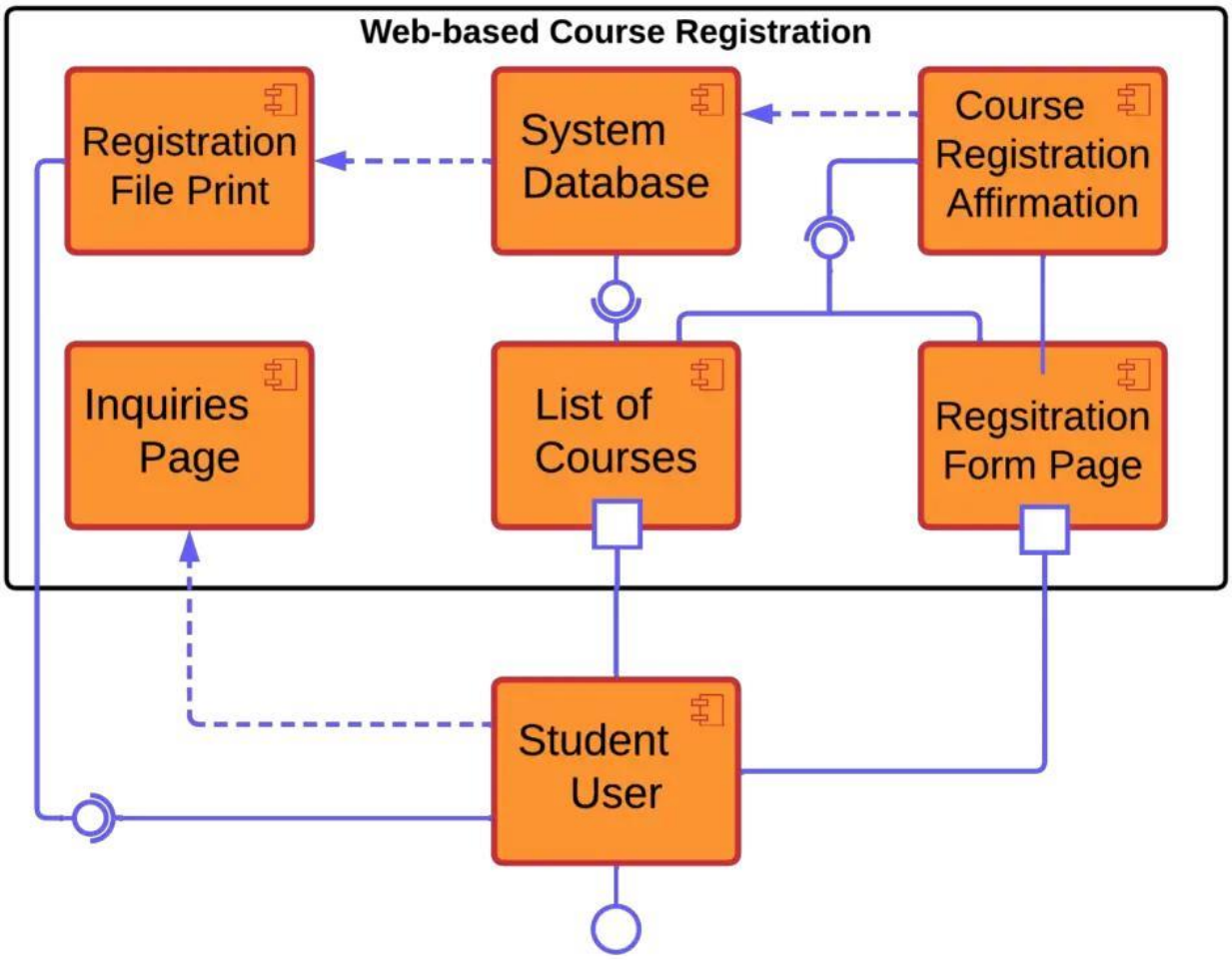
State-Chart diagram:

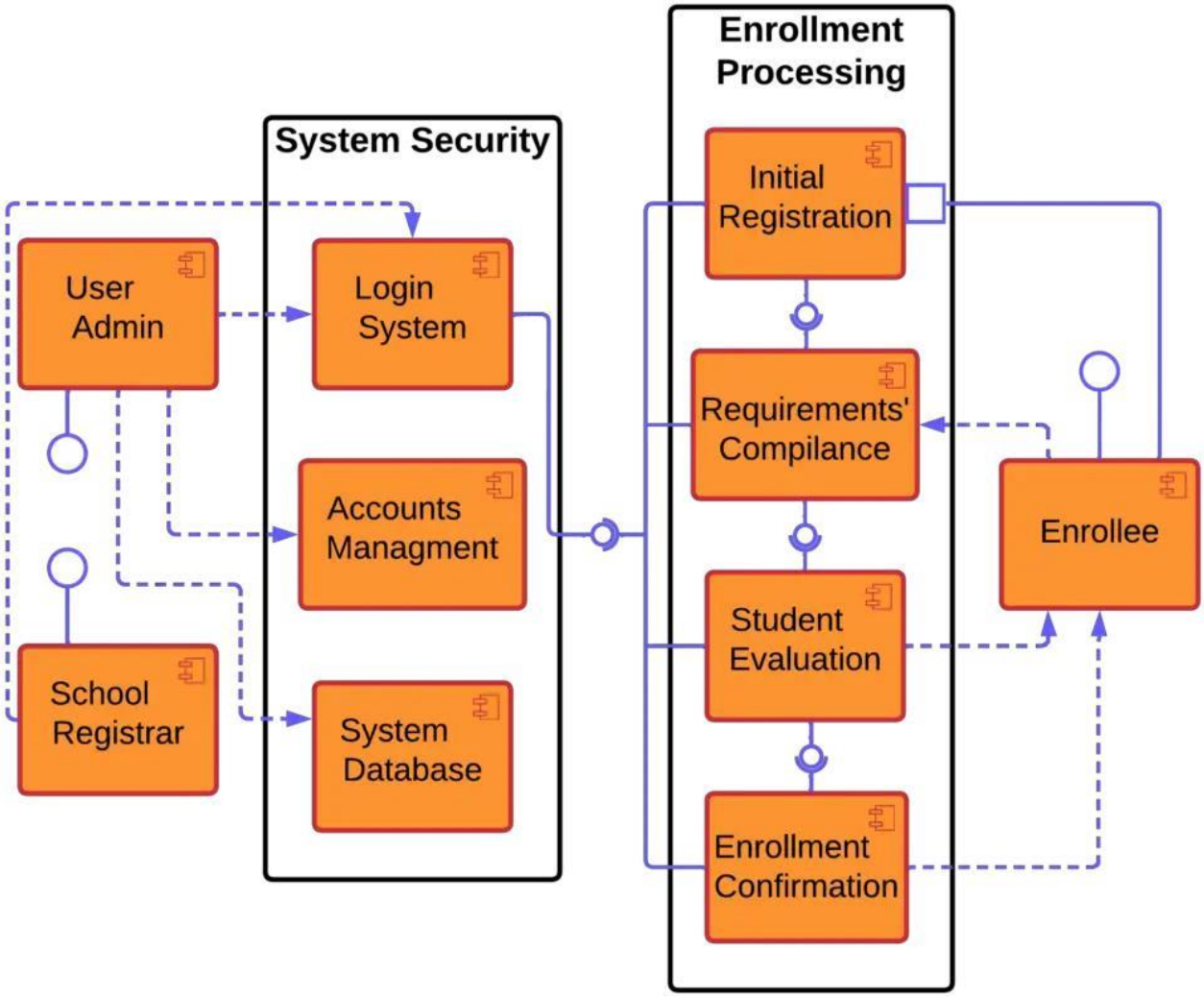


Deployment diagram:

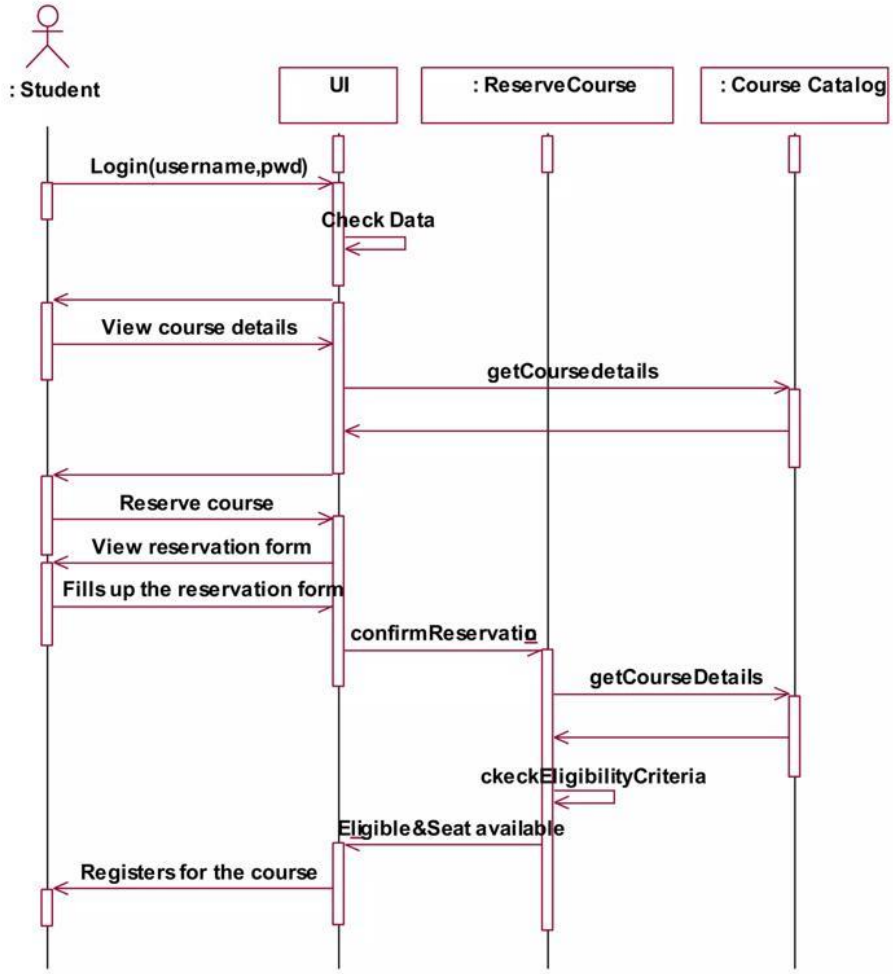


8. COMPONENT DIAGRAM




Enrollment Processing
System Security
Initial Registration
User Admin
Login System
Requirements' Compilance
Accounts Managment
Enrollee
Student Evaluation
School Registrar
System Database
Enrollment Confirmation

Sequence diagram:



CONCLUSION

The Online Course Reservation System:

- Simplifies course enrollment
- Provides real-time access to courses
- Reduces manual effort
- Improves user experience

EXPERIMENT–6: E-TICKETING SYSTEM

AIM

Performing the Design by using any Design phase CASE tools.

PROCEDURE

(I) PROBLEM STATEMENT

The E-Ticketing System is used for booking tickets online for services such as **bus, train, movie, or flight**.

- Users can register and log in
- Search available tickets
- Select seat and schedule
- Make payment
- Receive e-ticket confirmation

The system reduces manual booking effort and provides faster service.

(II) SOFTWARE REQUIREMENT SPECIFICATION (SRS)

1. INTRODUCTION

The E-Ticketing System acts as an interface between the **User** and the **Service Provider/Admin**, improving efficiency in ticket booking.

---

2. PURPOSE

Manual ticket booking is time-consuming and inefficient. This system:

- Enables online booking
- Reduces waiting time
- Provides real-time availability

---

3. SCOPE

- User registration and login
- Ticket search and booking
- Seat selection

- Payment processing
- Ticket cancellation
- Status tracking

---

4. DEFINITIONS & ACRONYMS

- **User**: Person booking tickets
- **Admin**: Manages system
- **ETS**: E-Ticketing System
- **HTML, HTTP, TCP/IP**: Web technologies

---

5. TECHNOLOGIES USED

- HTML
- CSS
- JavaScript
- Java / PHP

TOOLS USED

- Eclipse / VS Code
- Rational Rose (UML Tool)

---

6. OVERVIEW

SRS includes:

- Overall Description
- Specific Requirements

---

7. OVERALL DESCRIPTION

Product Perspective

The system connects users and service providers through an online platform.

Software Interface

- Front End: HTML, CSS, JavaScript

- Back End: Database (MySQL/Oracle)

Hardware Interface

- Client–Server architecture

---

8. SYSTEM FUNCTIONS

- User registration
- Ticket search
- Booking and cancellation
- Payment processing
- Notification system

---

9. USER CHARACTERISTICS

- **User**: Books tickets
- **Admin**: Manages services and bookings

---

10. CONSTRAINTS

- Requires internet connection
- Payment gateway dependency
- Security risks

---

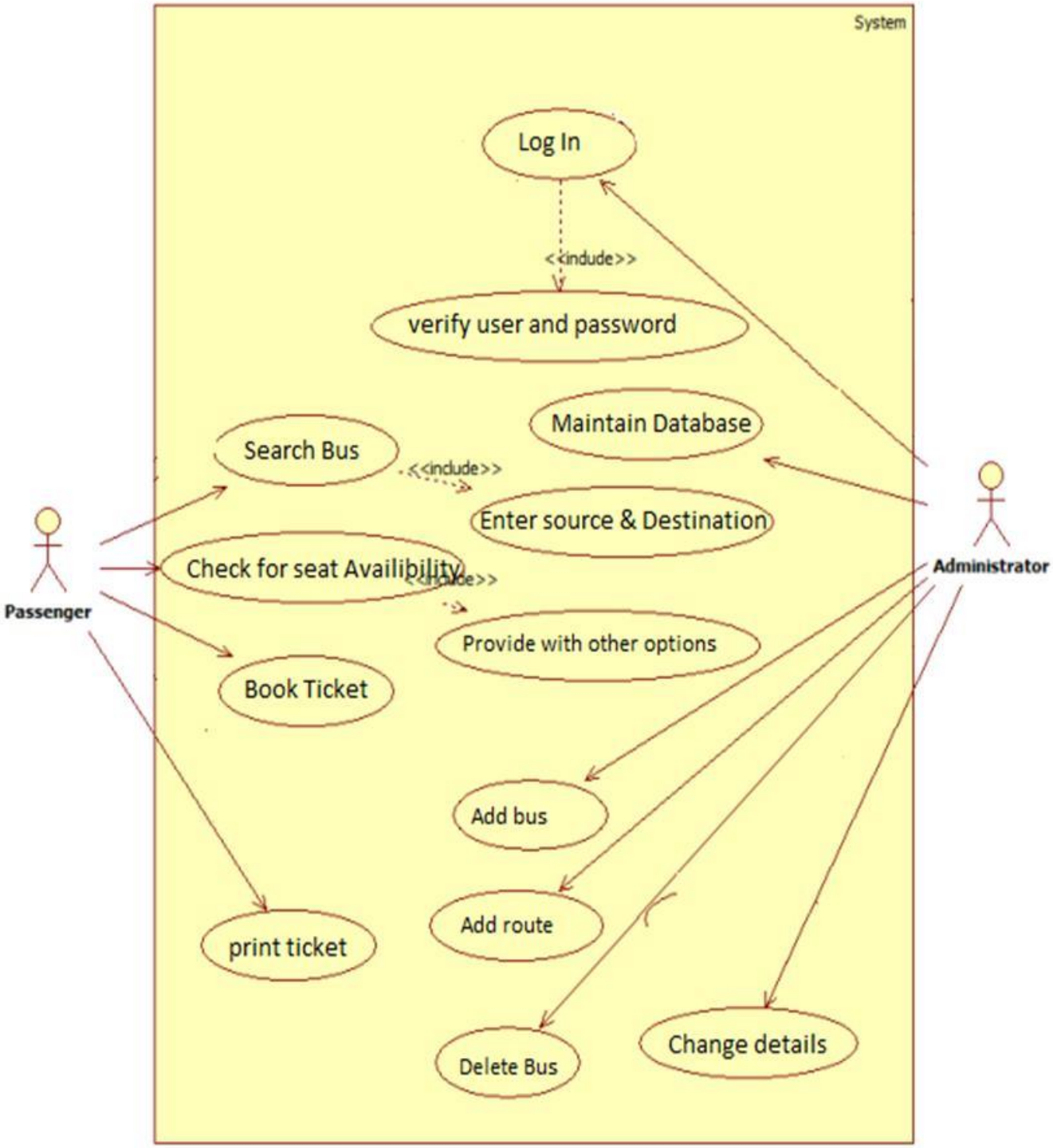
11. ASSUMPTIONS

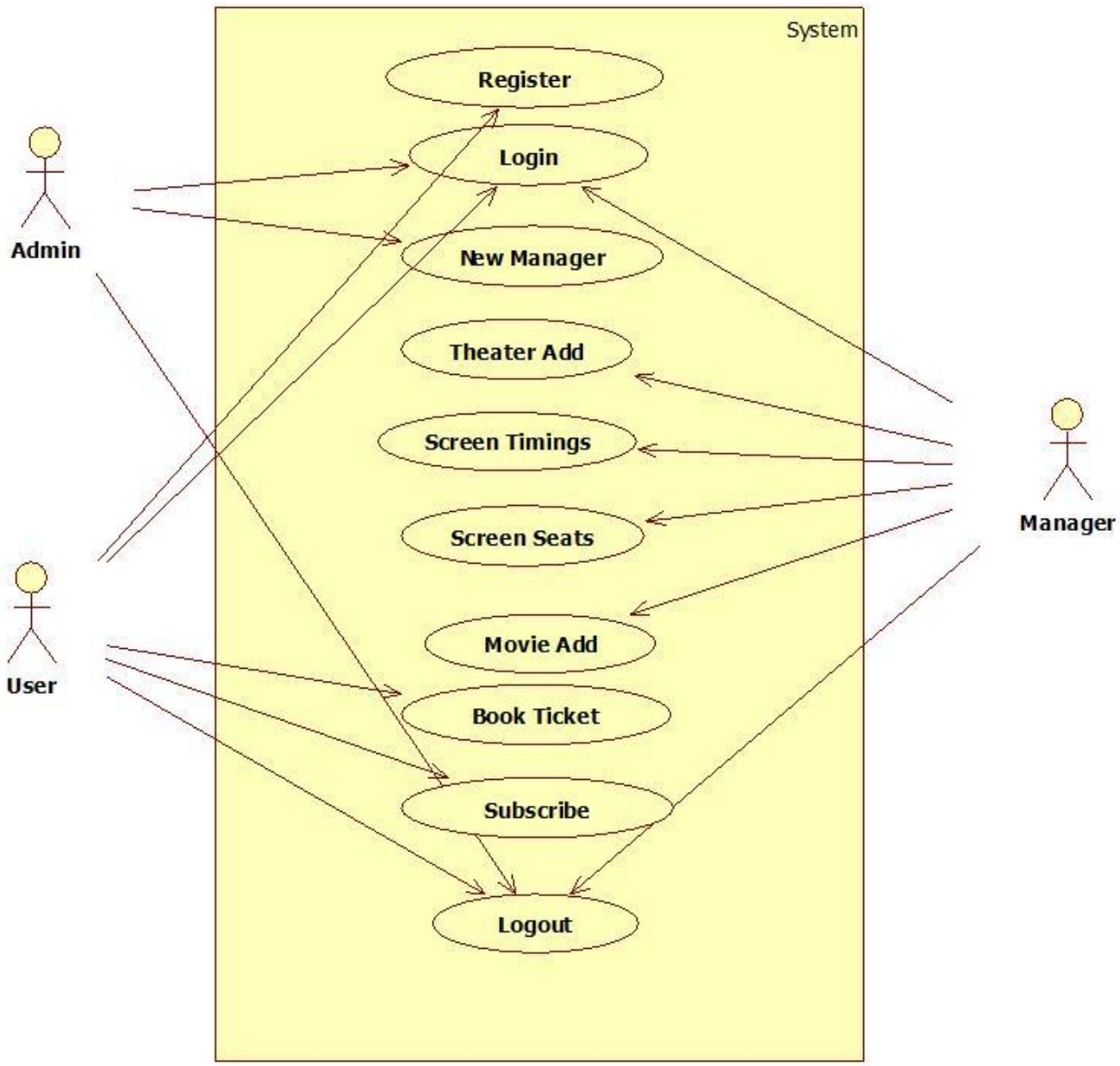
- Users have basic computer knowledge
- Online payment system available

(III) UML DIAGRAMS

---

1. USE CASE DIAGRAM


System
Log In
<<indude>>
verify user and password
Maintain Database
Search Bus
<<include>>
Enter source & Destination
Check for seat Availibility
<<include>>
Provide with other options
Book Ticket
Add bus
Add route
print ticket
Delete Bus
Change details
Passenger
Administrator


System
Register
Login
New Manager
Theater Add
Screen Timings
Screen Seats
Movie Add
Book Ticket
Subscribe
Logout
Admin
User
Manager

dig.Use Case Diagram for Booking Ticket (Online)

7

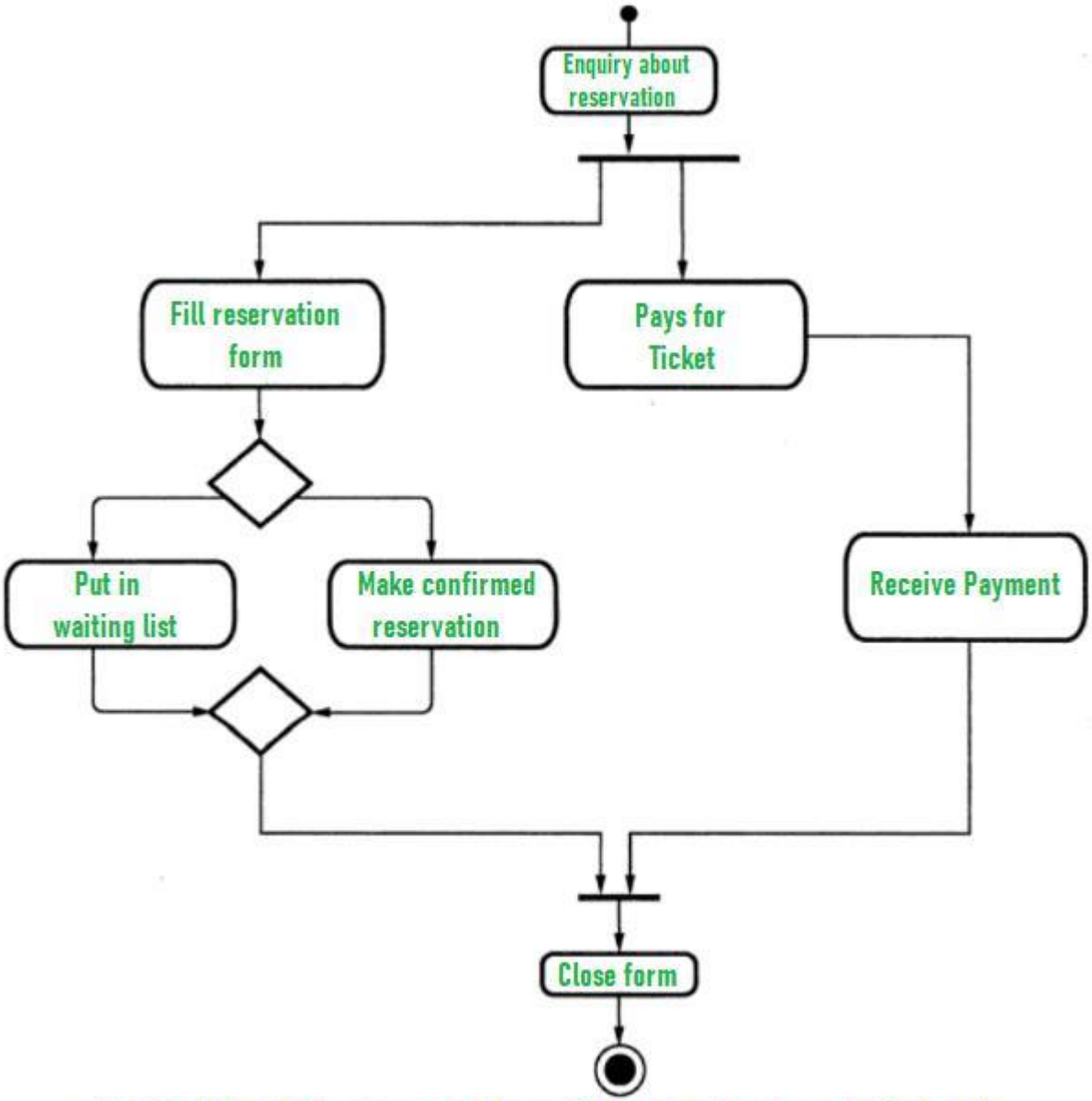
Actors

- User
- Admin

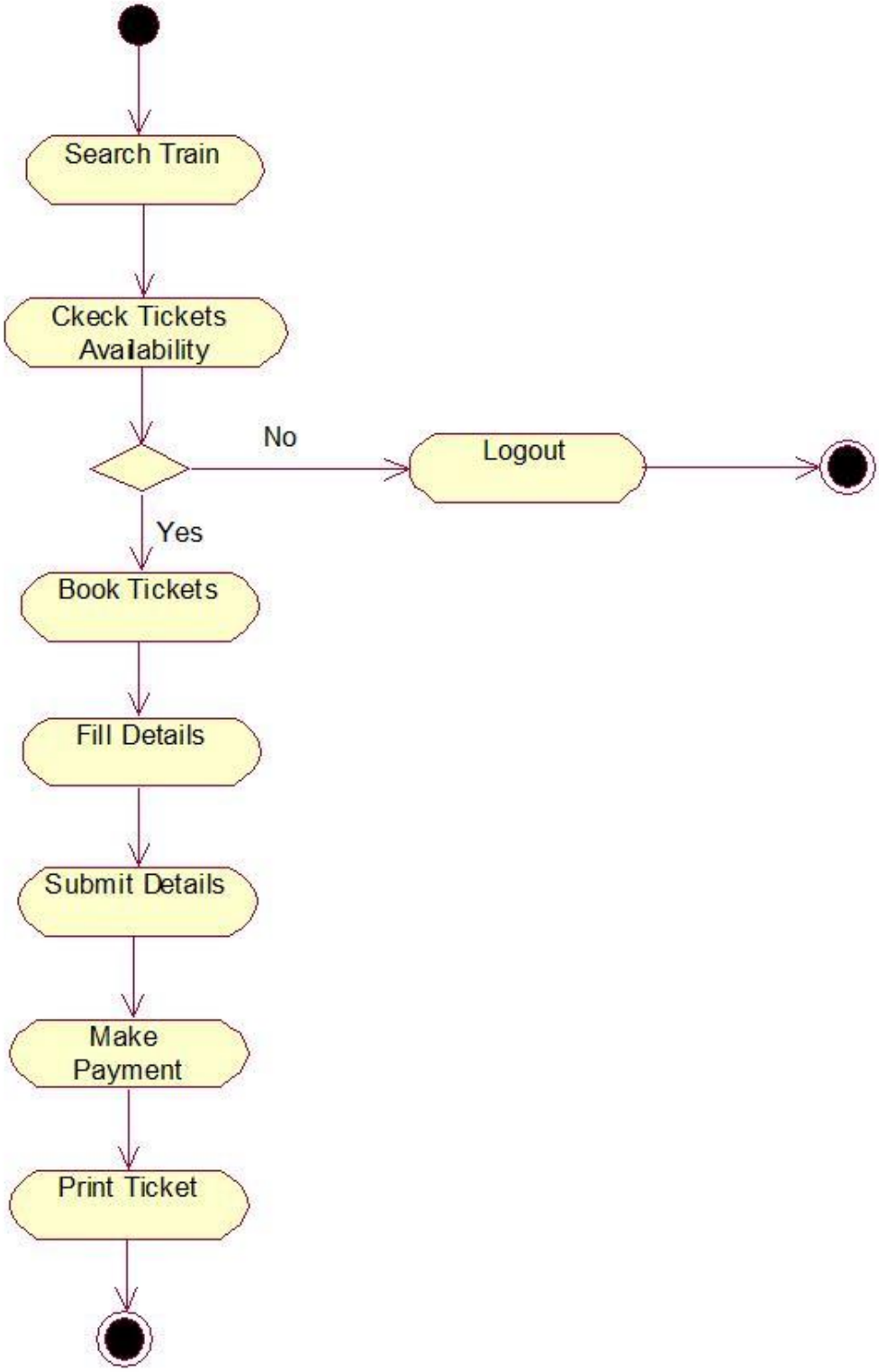
Use Cases

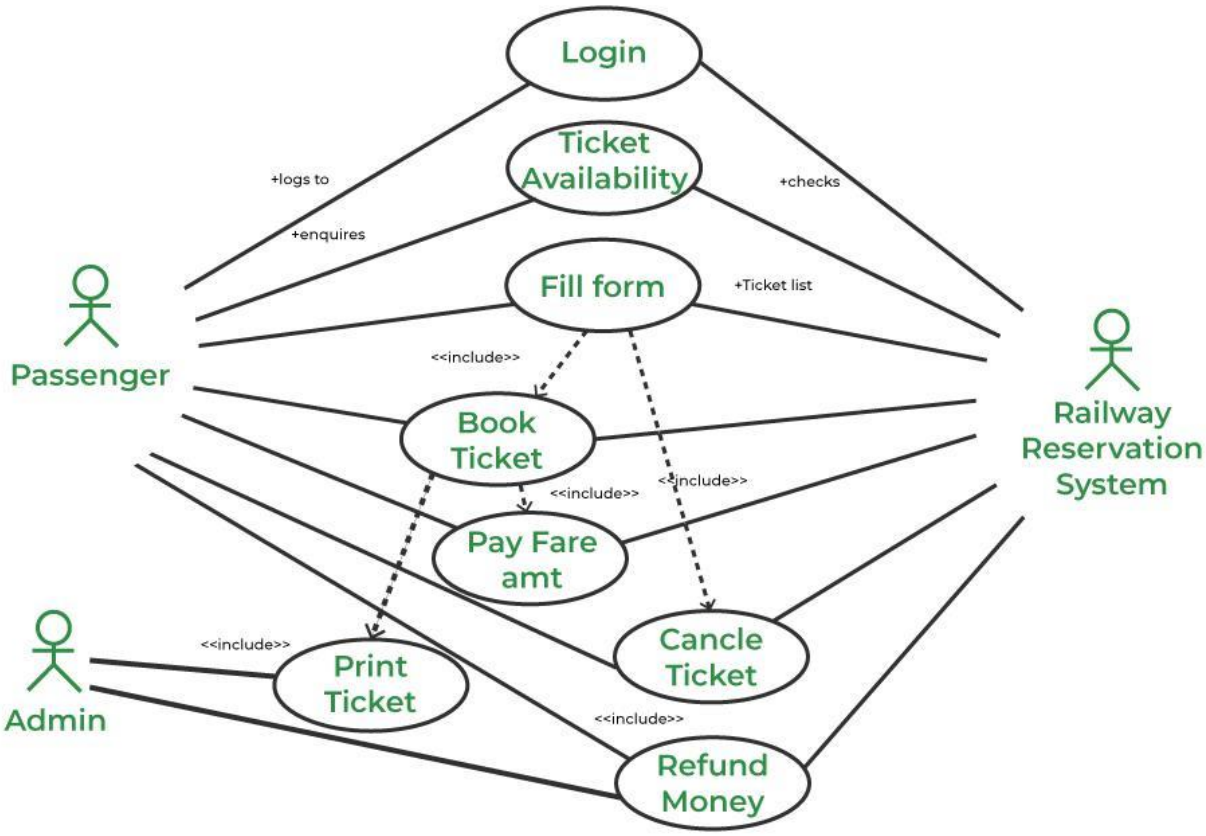
- Register
- Login
- Search Tickets
- Book Ticket
- Cancel Ticket
- Payment
- View Status

2. ACTIVITY DIAGRAM



Activity Diagram for Reserving a Ticket


Search Train
Ckeck Tickets Availability
No
Logout
Yes
Book Tickets
Fill Details
Submit Details
Make Payment
Print Ticket



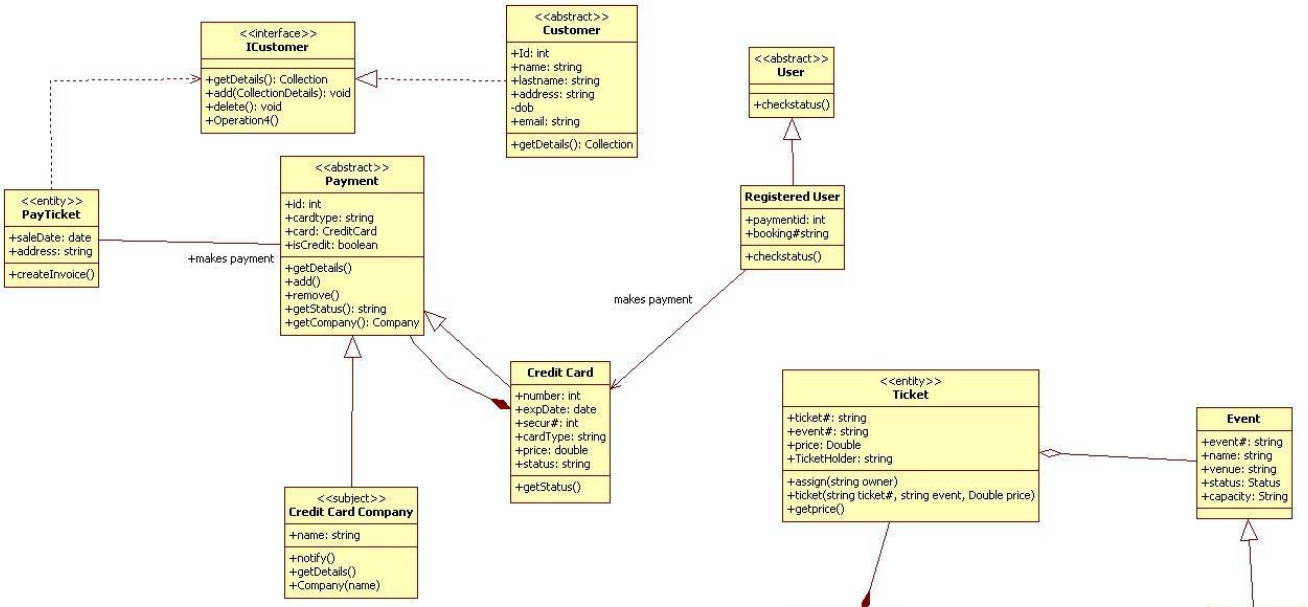
6

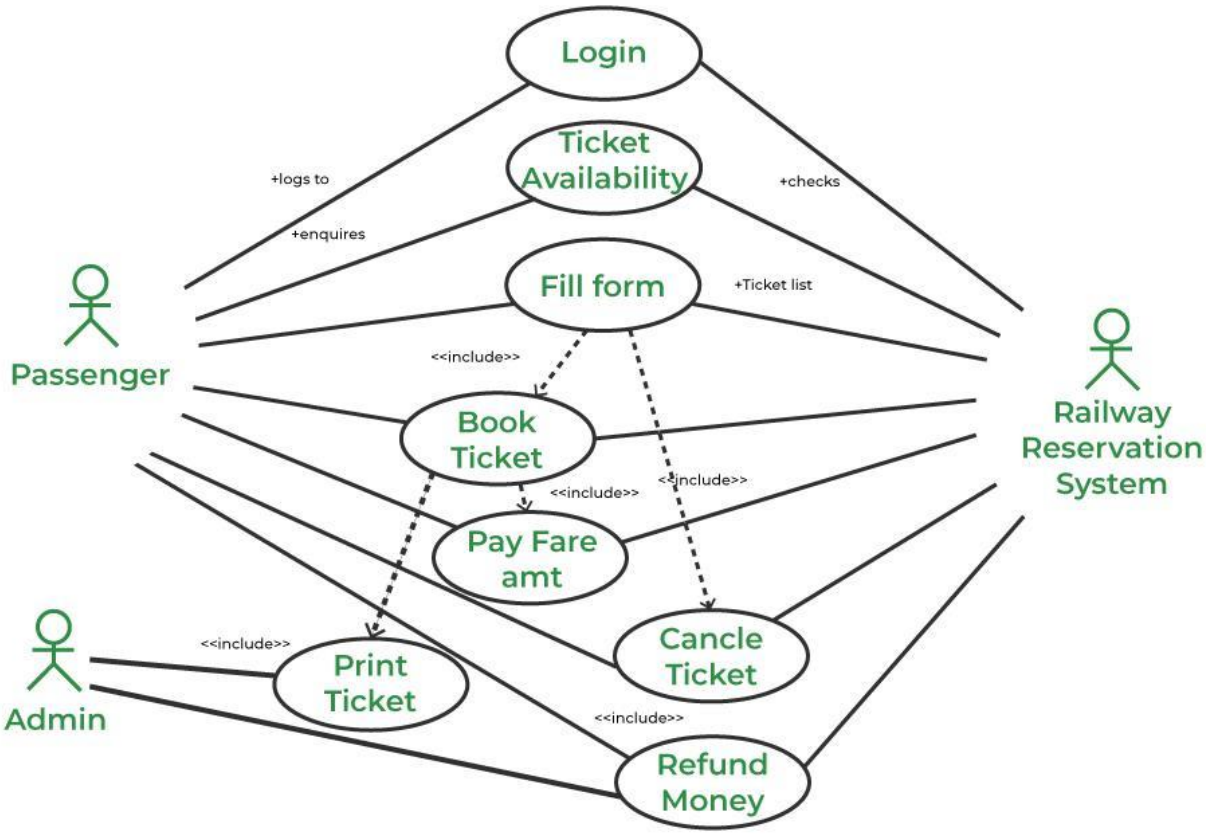
Flow

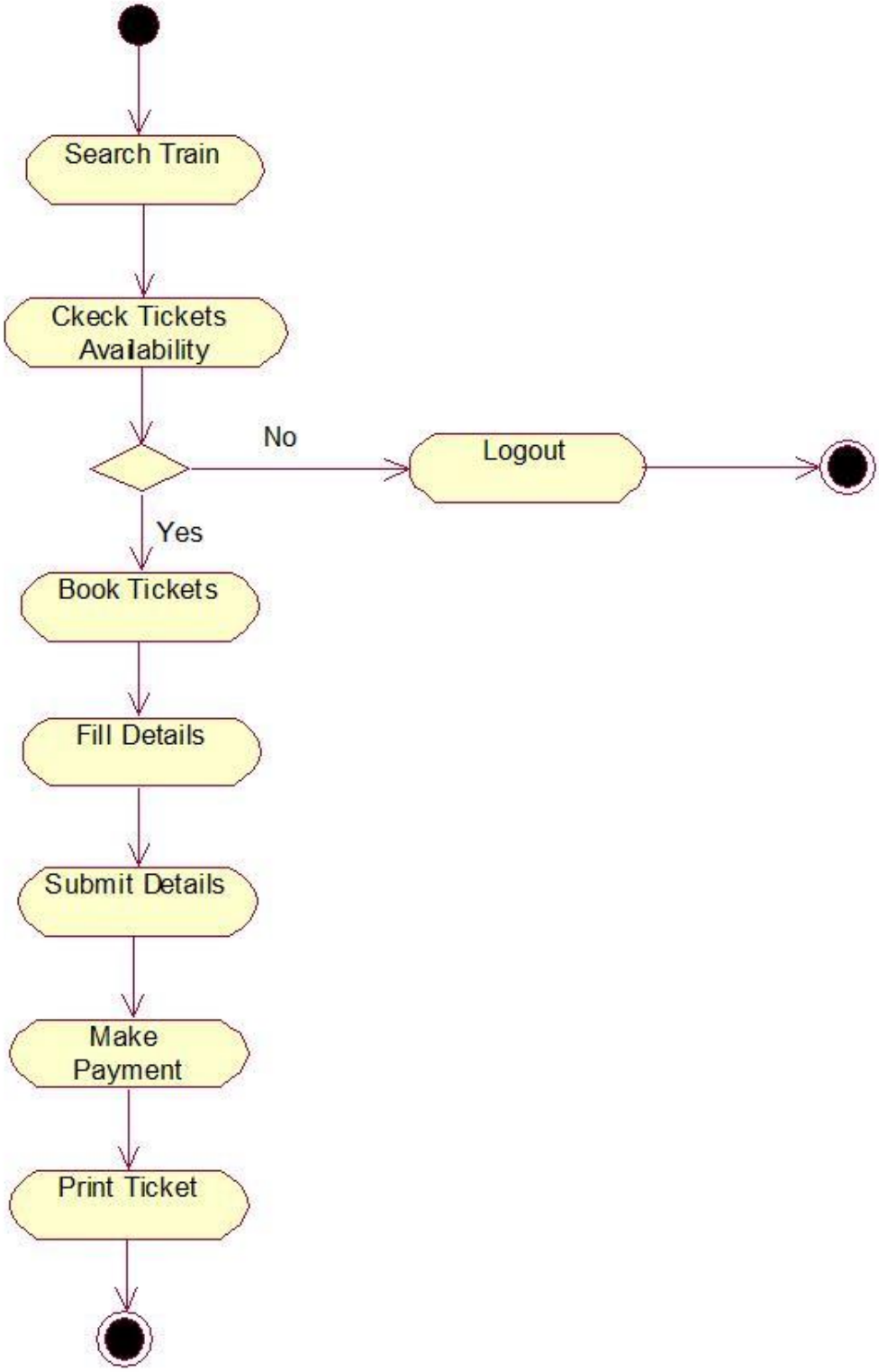
- Login
- Search tickets
- Select seat
- Make payment
- Confirm booking

---

3. CLASS DIAGRAM


<<interface>>
ICustomer
+getDetails(): Collection
+add(CollectionDetails): void
+delete(): void
+Operation4()
<<abstract>>
Customer
+Id: int
+name: string
+lastname: string
+address: string
-dob
+email: string
+getDetails(): Collection
<<abstract>>
User
+checkstatus()
<<entity>>
PayTicket
+saleDate: date
+address: string
+createInvoice()
+makes payment
<<abstract>>
Payment
+id: int
+cardtype: string
+card: CreditCard
+isCredit: boolean
+getDetails()
+add()
+remove()
+getStatus(): string
+getCompany(): Company
Registered User
+paymentid: int
+booking#string
+checkstatus()
makes payment
Credit Card
+number: int
+expDate: date
+secur#: int
+cardType: string
+price: double
+status: string
+getStatus()
<<subject>>
Credit Card Company
+name: string
+notify()
+getDetails()
+Company(name)
<<entity>>
Ticket
+ticket#: string
+event#: string
+price: Double
+TicketHolder: string
+assign(string owner)
+ticket(string ticket#, string event, Double price)
+getprice()
Event
+event#: string
+name: string
+venue: string
+status: Status
+capacity: String
Ticketholder
+id: int
+customer#: string
+eventDate: date
+tickets: vector
+confirm()
+getTotal()
Seat
+seatId: int
+seatstaus: string
+setstatus()
+getstatus()
<<entity>>
Venue
+venue#: string
+name: string
+status: Status
+capacity: string


Login
Ticket Availability
Fill form
+logs to
+enquires
+checks
+Ticket list
Passenger
<<include>>
Book Ticket
<<include>>
<<include>>
Pay Fare amt
Railway Reservation System
Cancle Ticket
<<include>>
Print Ticket
Admin
<<include>>
Refund Money


Search Train
Ckeck Tickets Availability
No
Logout
Yes
Book Tickets
Fill Details
Submit Details
Make Payment
Print Ticket

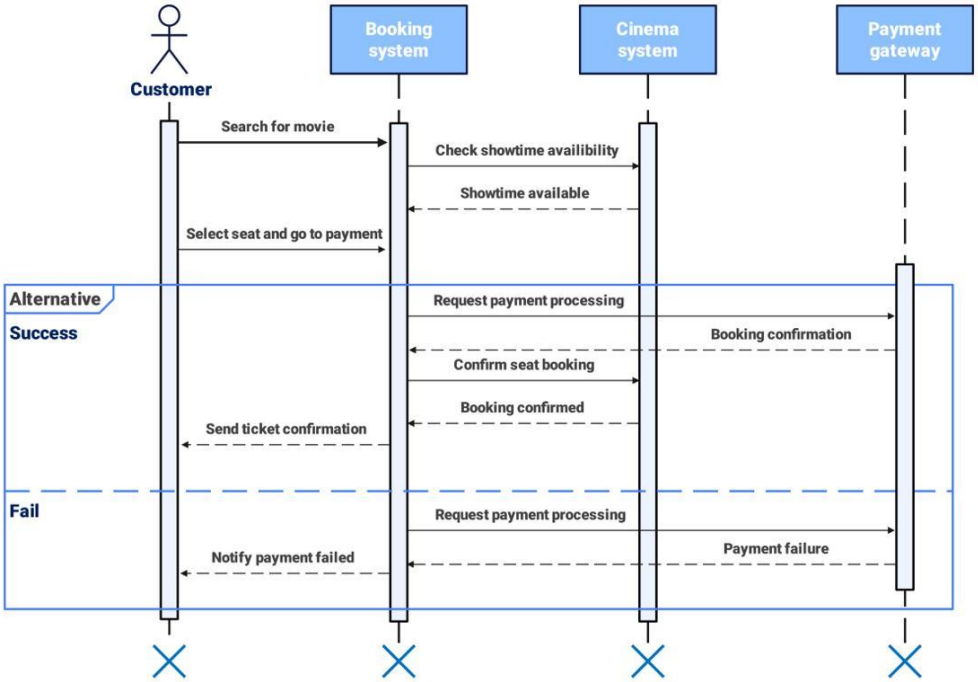
5

Classes

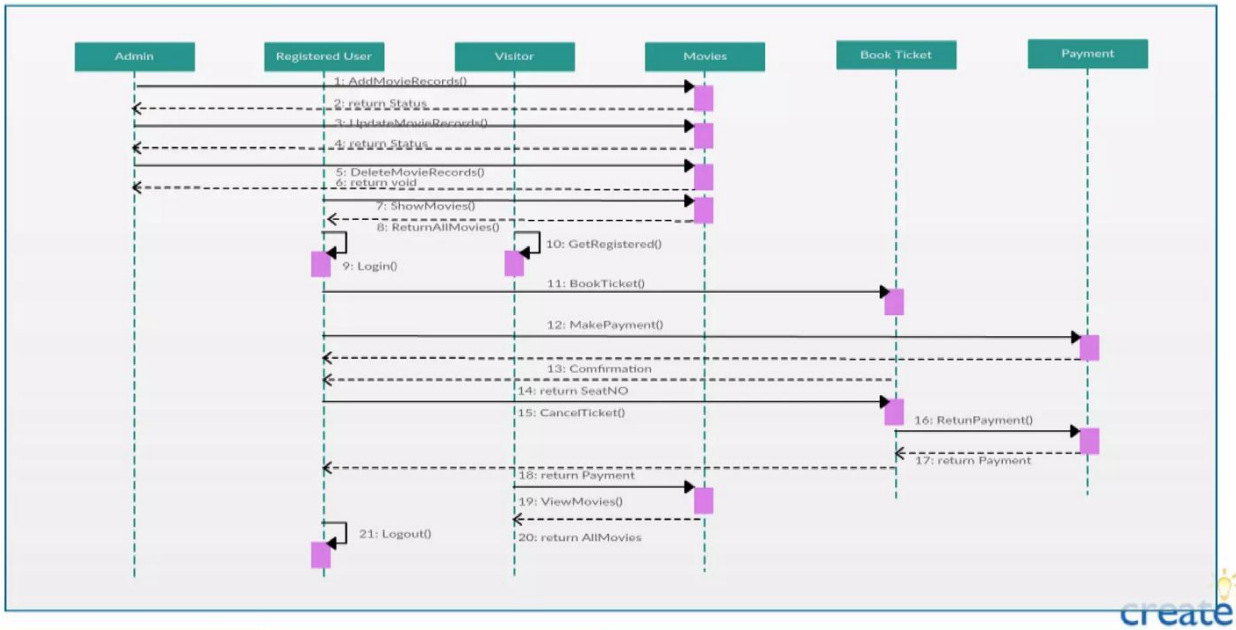
1. User
2. Ticket
3. Booking
4. Payment
5. Admin

---

4. SEQUENCE DIAGRAM

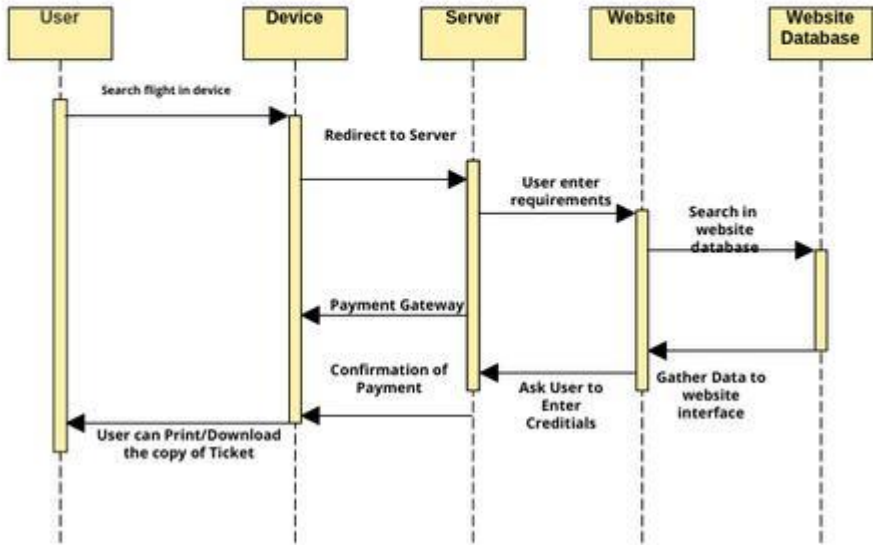


Sequence Diagram Example of an Online Movie Ticket Booking System



aditi saxena
e22cseu0703

Flight Ticket Booking Sequence Diagram



6

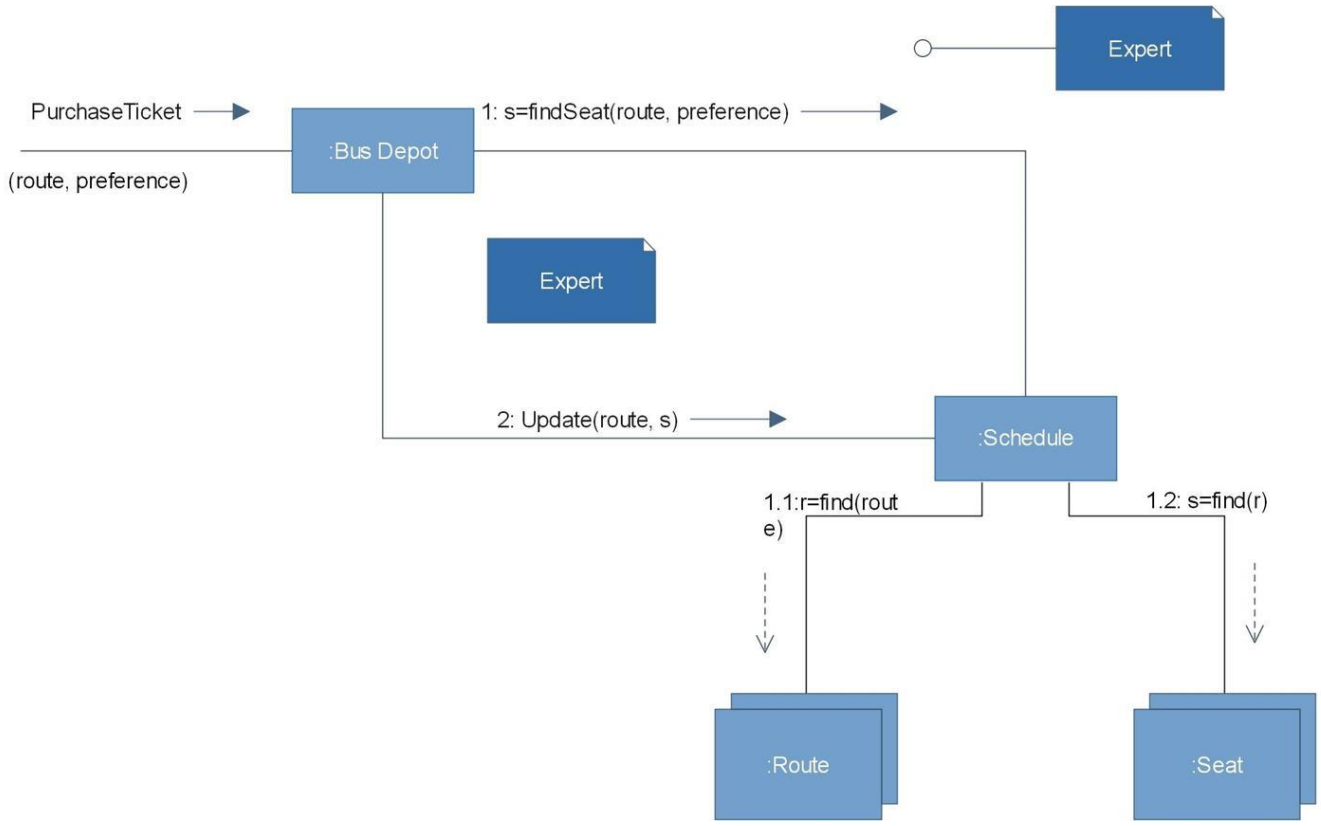
Description

Shows interaction between:

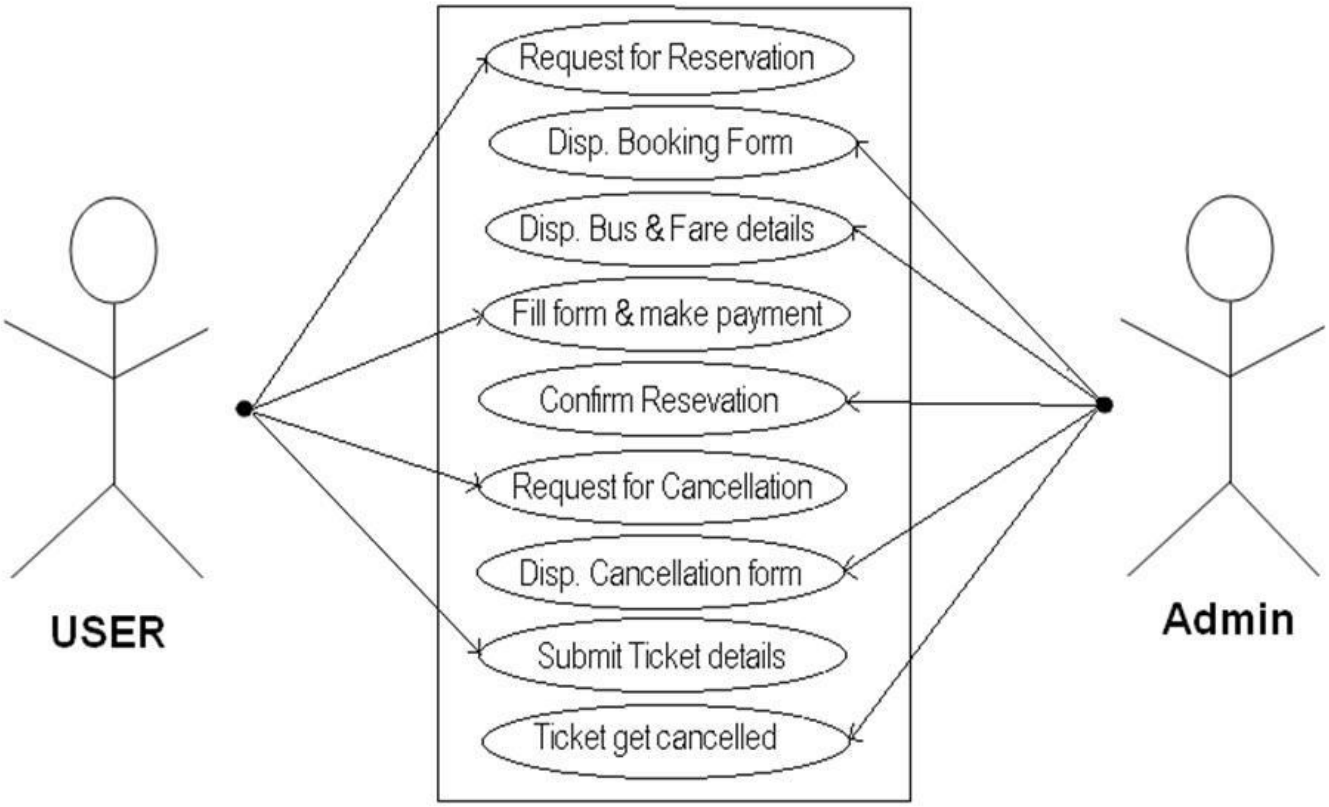
- User
- System
- Database

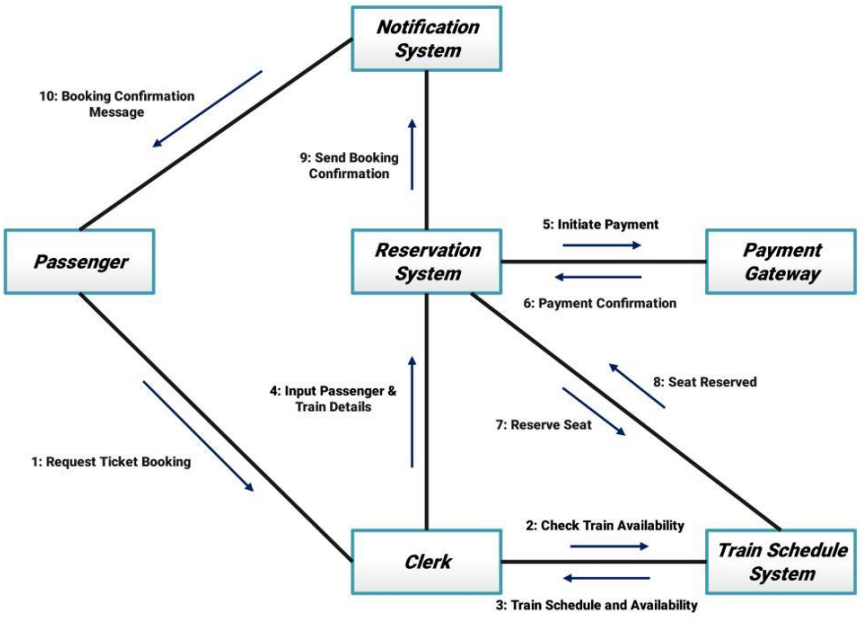
---

5. COLLABORATION DIAGRAM

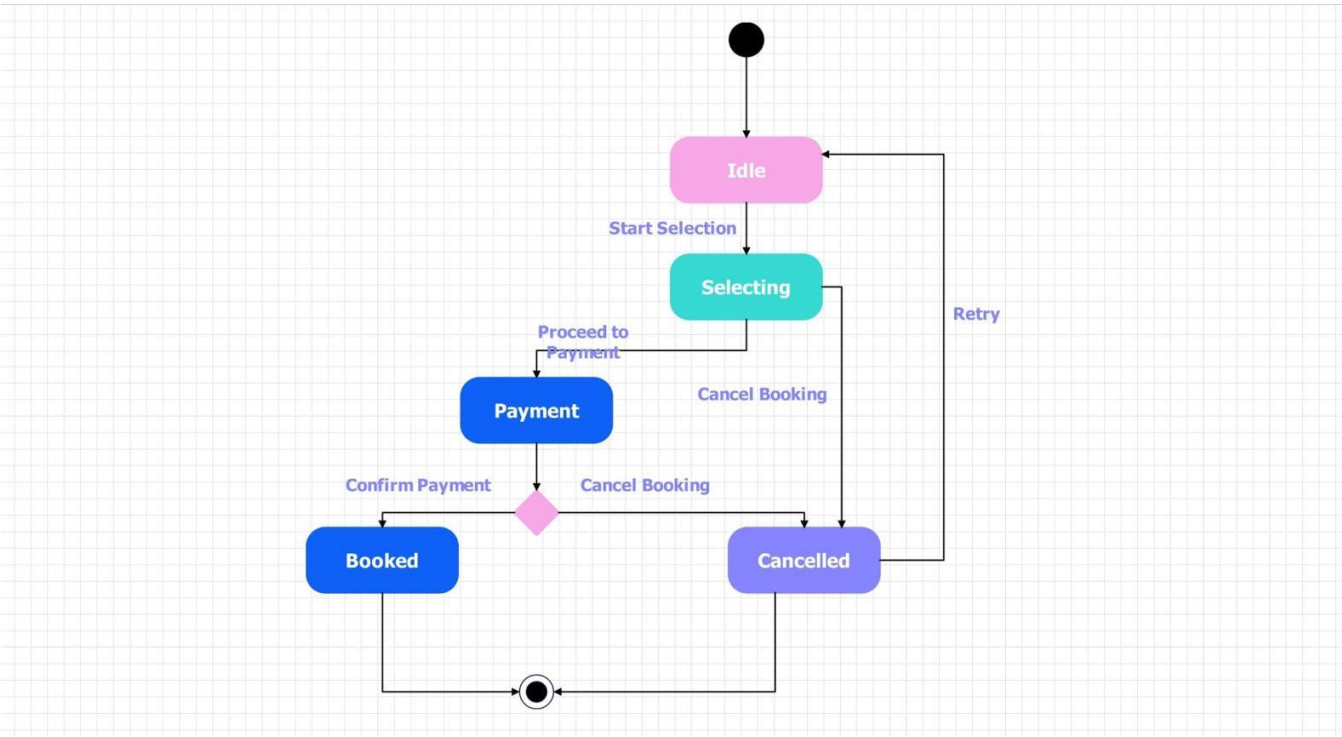


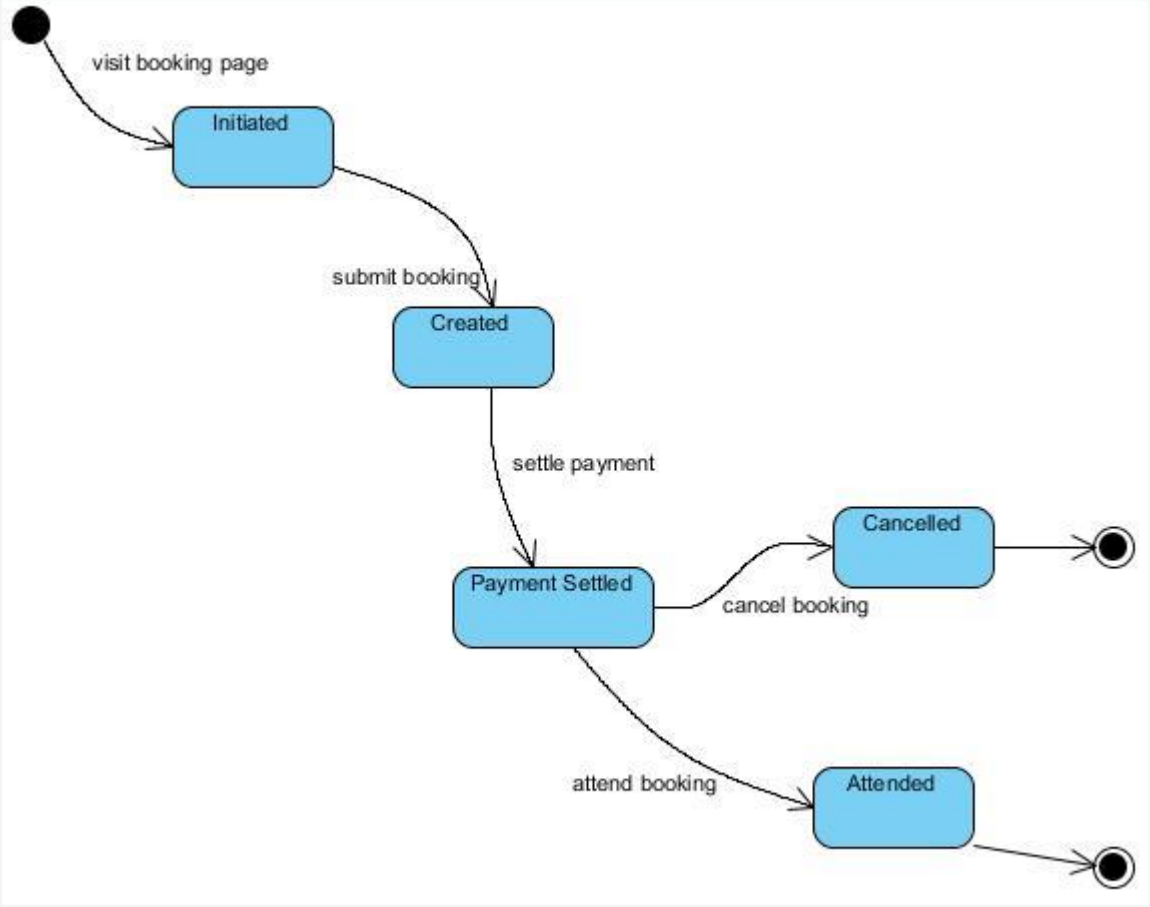
Content Management System - Use Case Diagram For Ticket Booking System

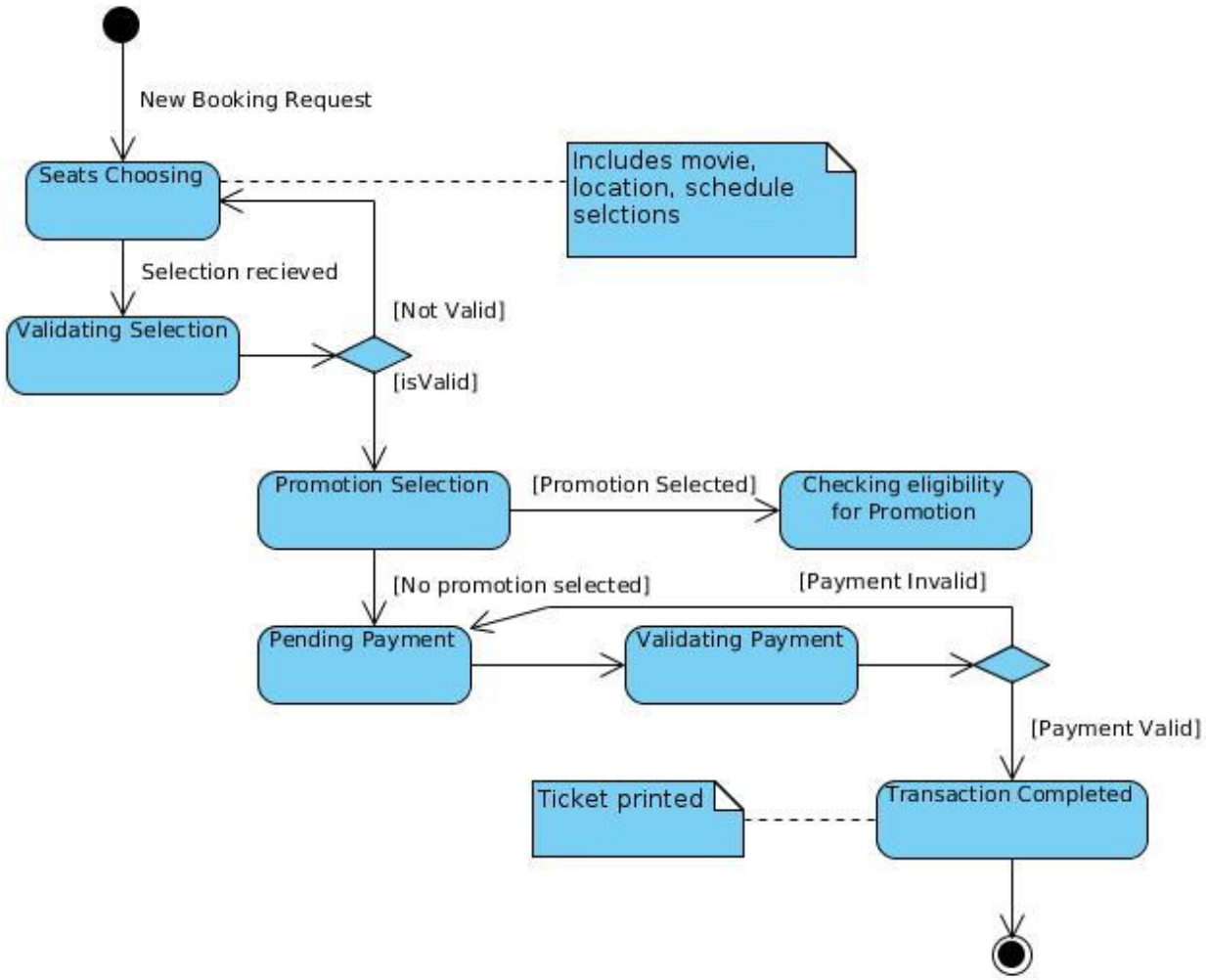




6. STATECHART DIAGRAM




visit booking page
Initiated
submit booking
Created
settle payment
Payment Settled
cancel booking
Cancelled
attend booking
Attended

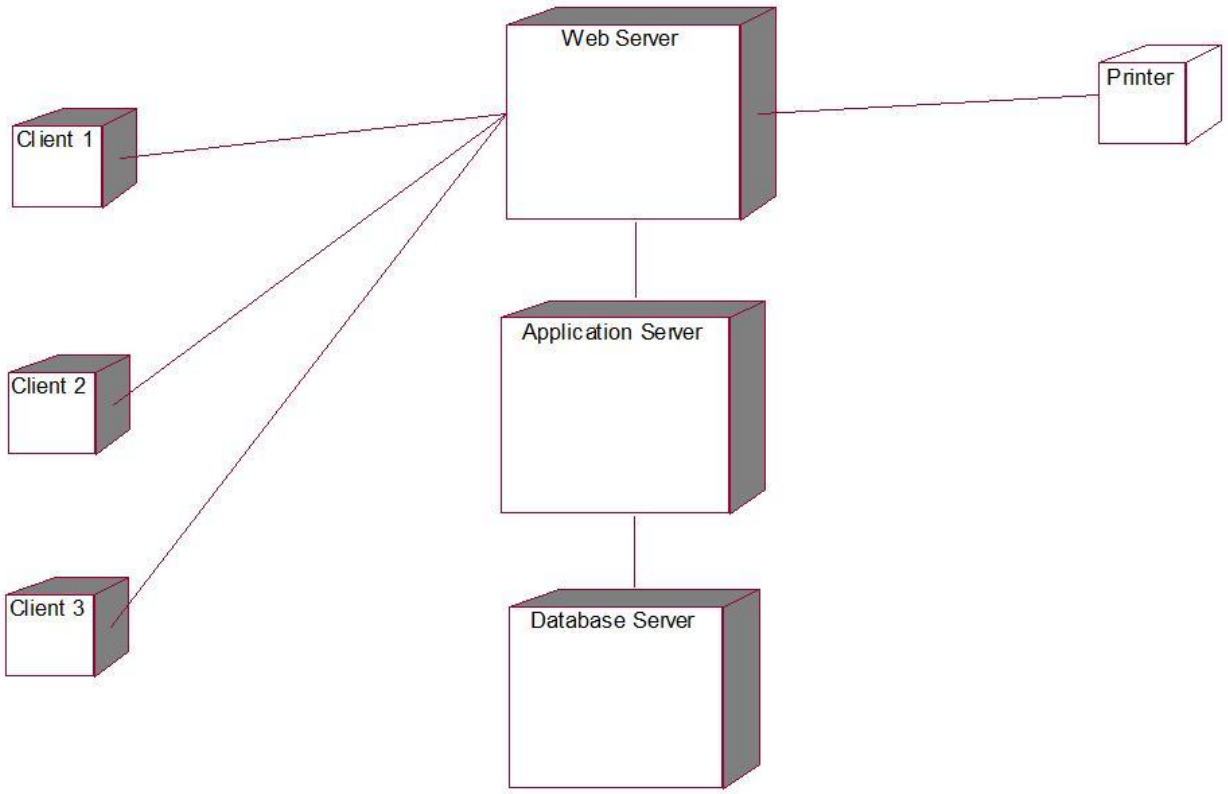


7

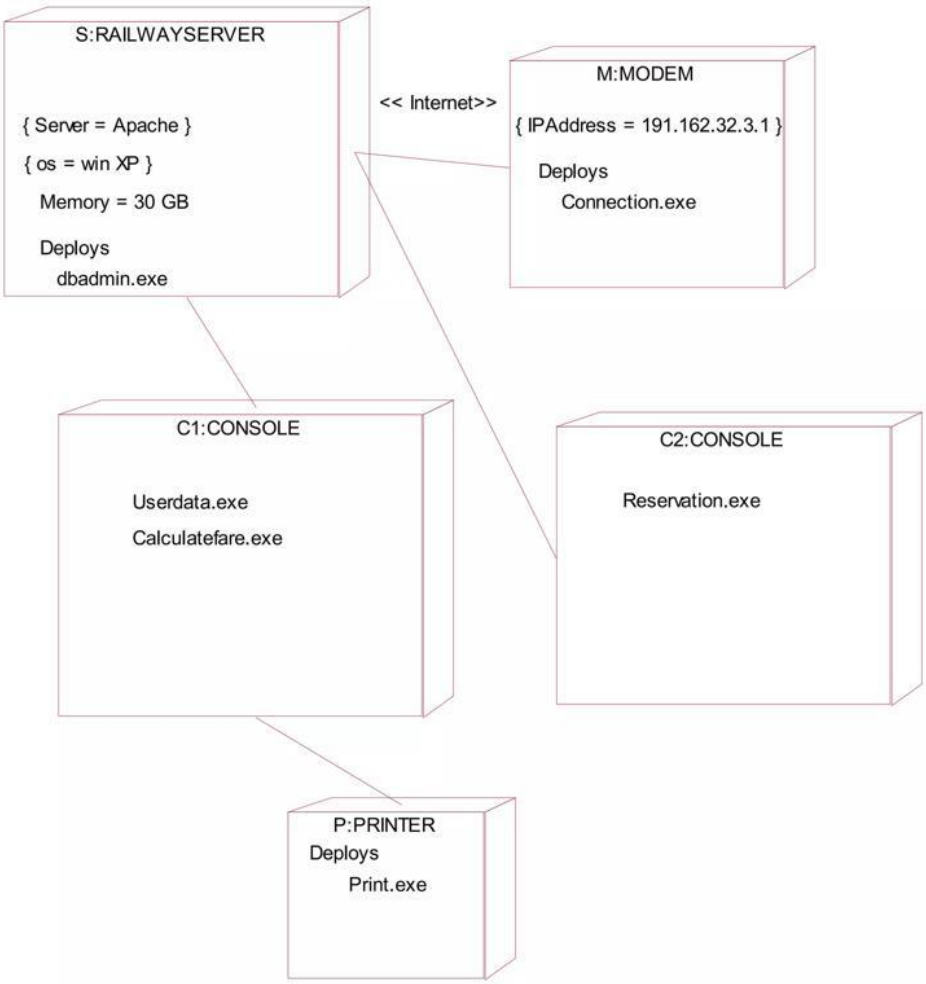
States

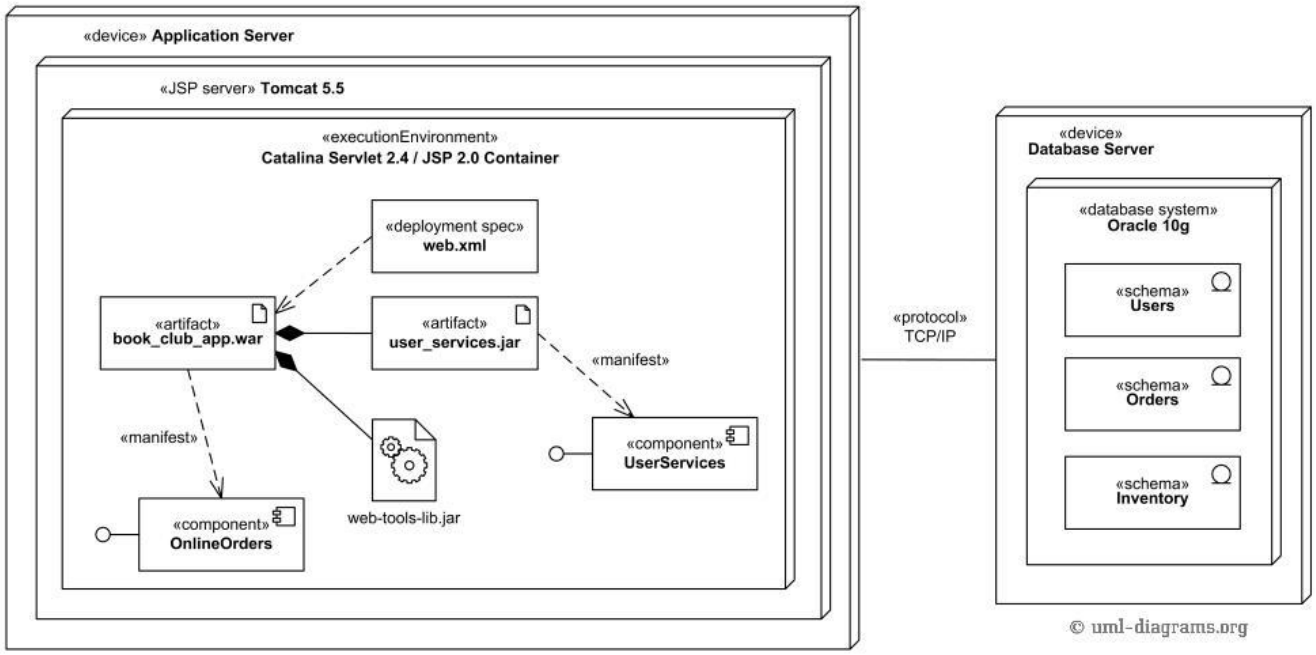
- Available
- Booked
- Confirmed
- Cancelled

7. DEPLOYMENT DIAGRAM



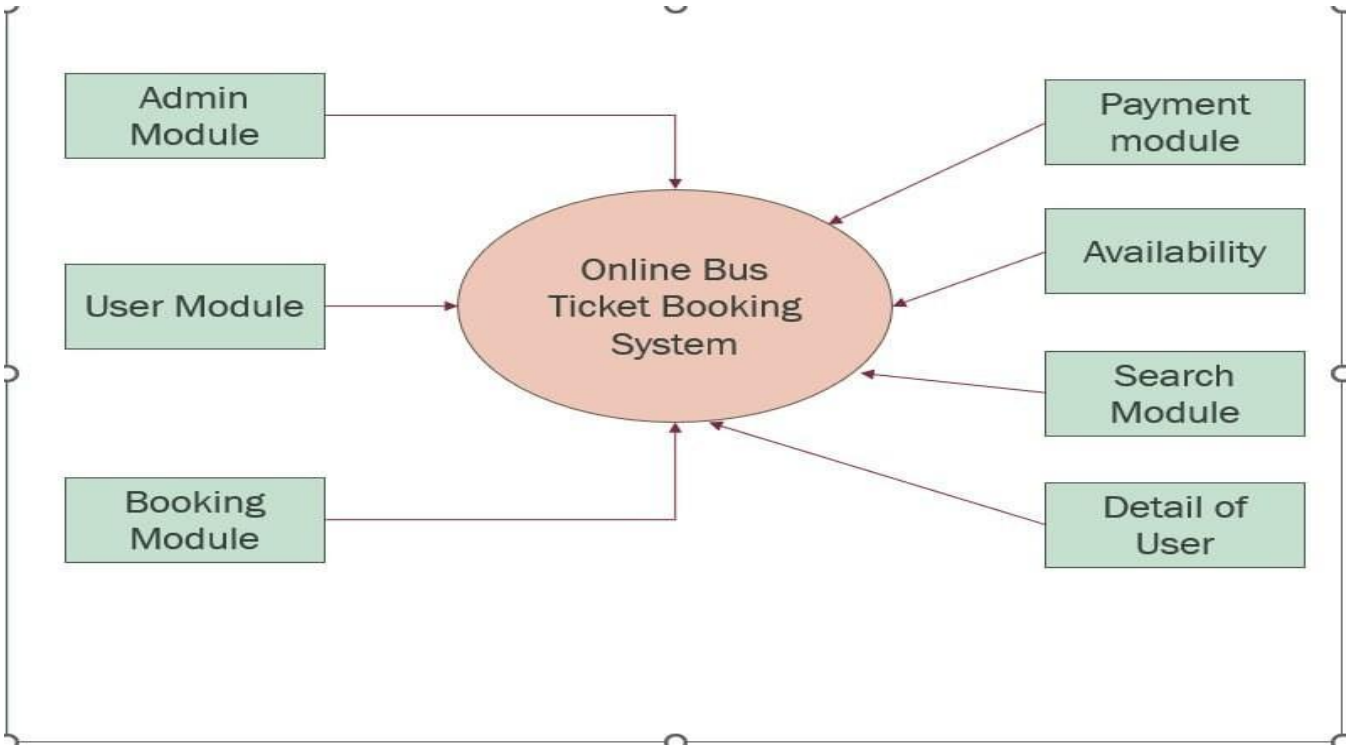
DEPLOYMENT DIAGRAM – ONLINE RAILWAY RESERVATION SYSTEM

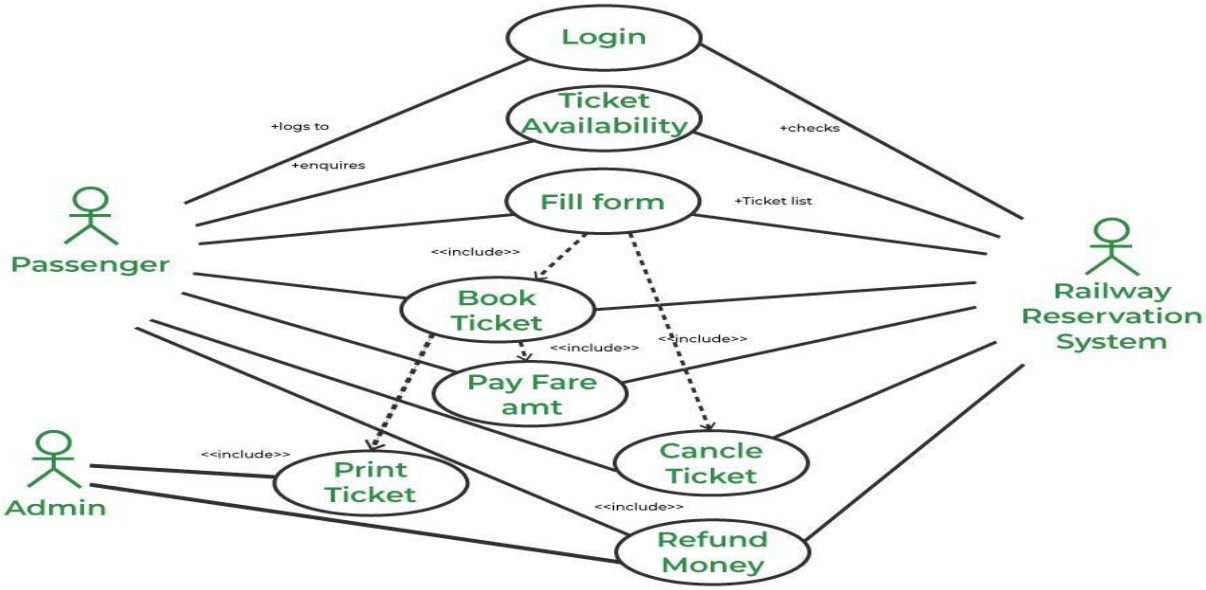


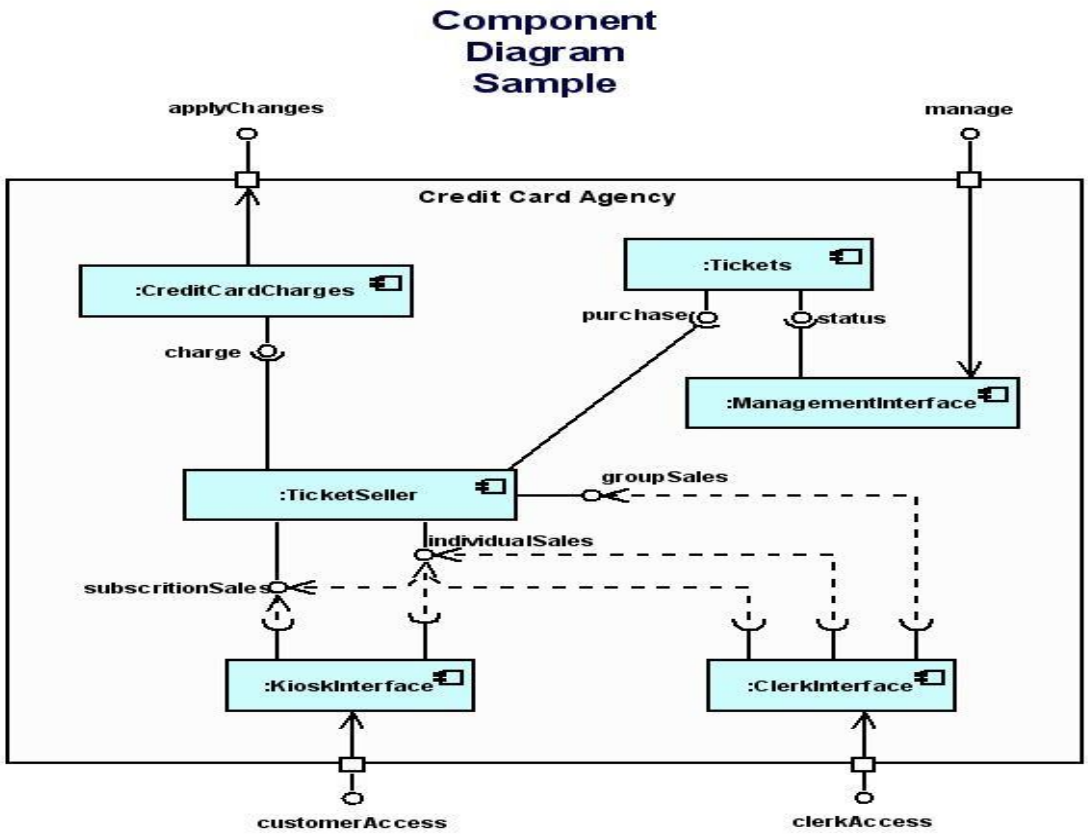


6

8. COMPONENT DIAGRAM




Login
Ticket Availability
Fill form
Book Ticket
Pay Fare amt
Cancle Ticket
Print Ticket
Refund Money
Passenger
Admin
Railway Reservation System
+logs to
+enquires
+checks
+Ticket list
<<include>>
<<include>>
<<include>>
<<include>>
<<include>>


Component Diagram Sample
applyChanges
manage
Credit Card Agency
:CreditCardCharges
:Tickets
charge
purchase
status
:ManagementInterface
:TicketSeller
groupSales
individualSales
subscritionSales
:KioskInterface
:ClerkInterface
customerAccess
clerkAccess

CONCLUSION

The E-Ticketing System:

- Automates ticket booking
- Reduces manual work
- Provides real-time availability
- Improves user convenience

EXPERIMENT–7: SOFTWARE PERSONNEL MANAGEMENT SYSTEM

AIM

PROCEDURE

(I) PROBLEM STATEMENT

The Software Personnel Management System (SPMS) is designed to manage employee details, project assignments, and performance within a software organization.

- Maintains employee records
- Assigns employees to projects
- Tracks work progress and performance
- Manages payroll and reports

The system reduces manual effort and improves workforce management.

(II) SOFTWARE REQUIREMENT SPECIFICATION (SRS)

1. INTRODUCTION

SPMS acts as an interface between **Employees** and **Management/Admin**, improving efficiency in personnel management.

---

2. PURPOSE

Manual personnel management is inefficient. This system:

- Automates employee record handling
- Improves project tracking
- Ensures accurate data management

---

3. SCOPE

- Employee registration
- Project assignment
- Attendance tracking
- Salary management

- Report generation

---

4. DEFINITIONS & ACRONYMS

- **Employee**: Staff working in organization
- **Admin/Manager**: Manages employees
- **SPMS**: Software Personnel Management System

---

5. TECHNOLOGIES USED

- HTML
- CSS
- JavaScript
- Java / PHP

TOOLS USED

- Eclipse / VS Code
- Rational Rose (UML Tool)

---

6. OVERVIEW

SRS includes:

- Overall Description
- Specific Requirements

---

7. OVERALL DESCRIPTION

Product Perspective

The system provides a centralized platform to manage employees and projects.

Software Interface

- Front End: HTML, CSS
- Back End: Database (MySQL/Oracle)

Hardware Interface

- Client–Server architecture

---

8. SYSTEM FUNCTIONS

- Employee registration
- Project assignment
- Attendance management
- Salary processing
- Report generation

---

9. USER CHARACTERISTICS

- **Employee**: Updates personal/work details
- **Admin/Manager**: Assigns EXPERIMENTs and monitors performance

---

10. CONSTRAINTS

- Requires system access
- Data security must be maintained
- Proper authentication required

---

11. ASSUMPTIONS

- Users have basic computer knowledge
- Organization maintains database

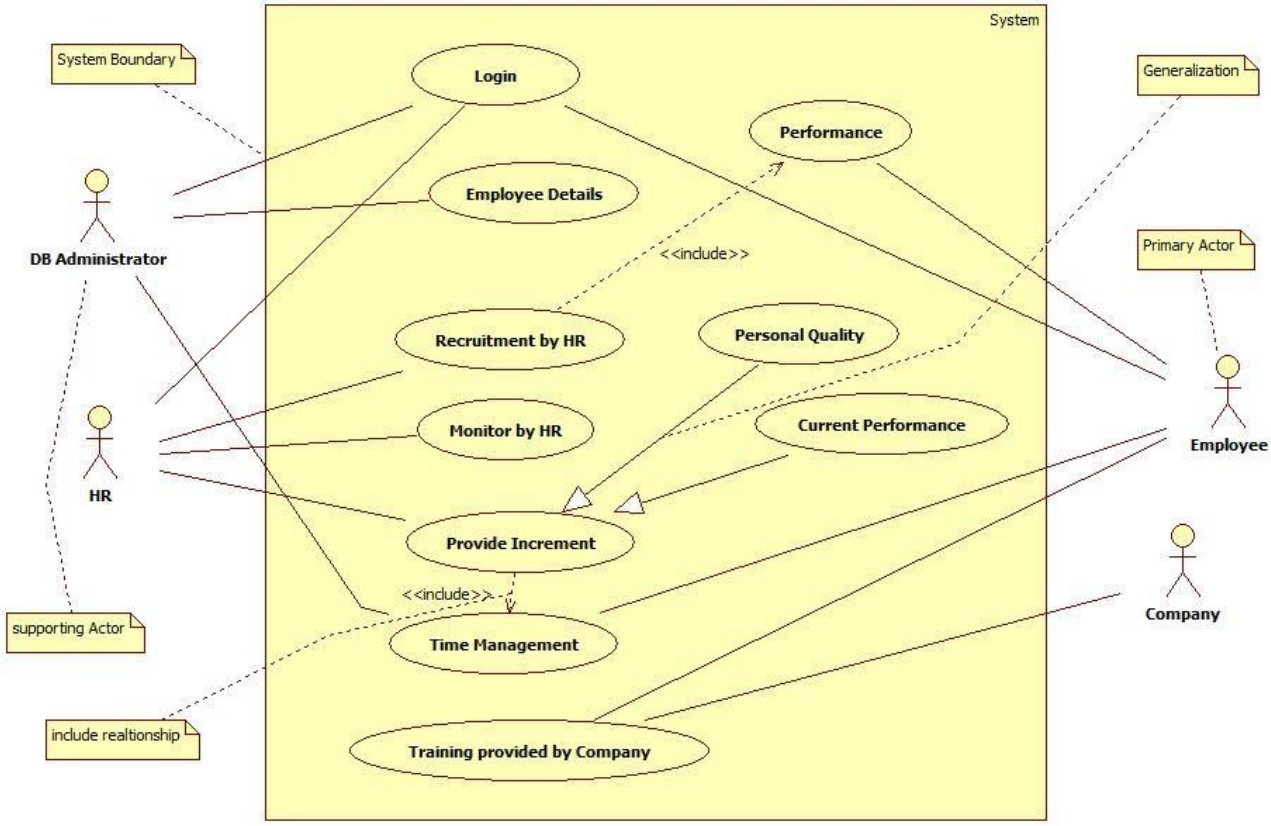
(III) UML DIAGRAMS

---

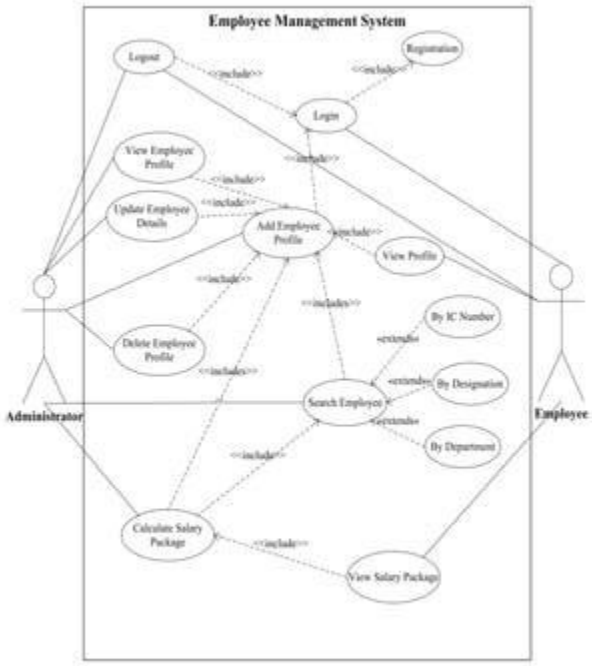
1. USE CASE DIAGRAM

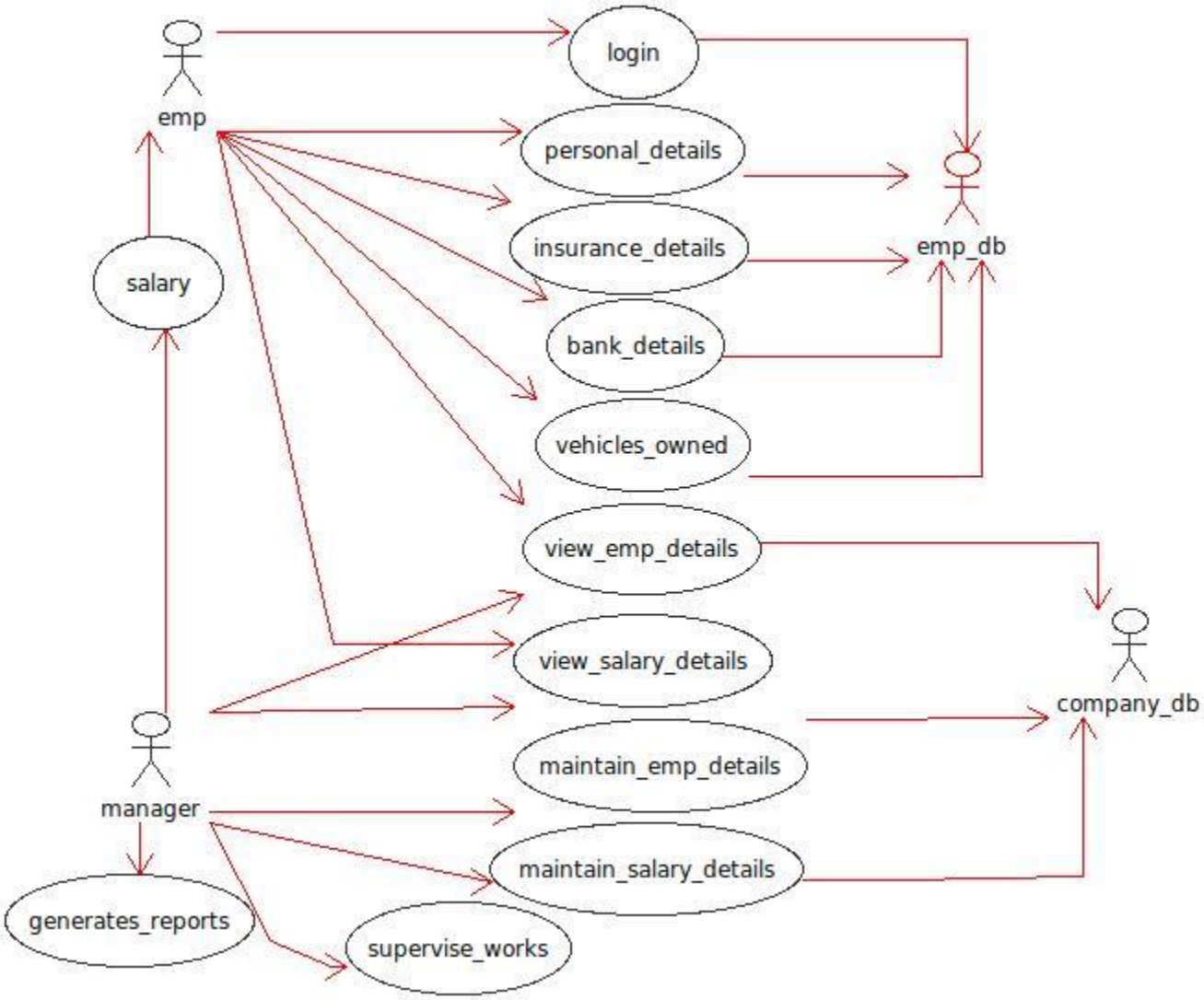
Ex:No:
Date:

Software Personnel Management System



4.2. Use Case Diagram





6

Actors

- Employee
- Admin/Manager

Use Cases

- Login
- Register Employee
- Assign Project
- Update Details
- View Reports

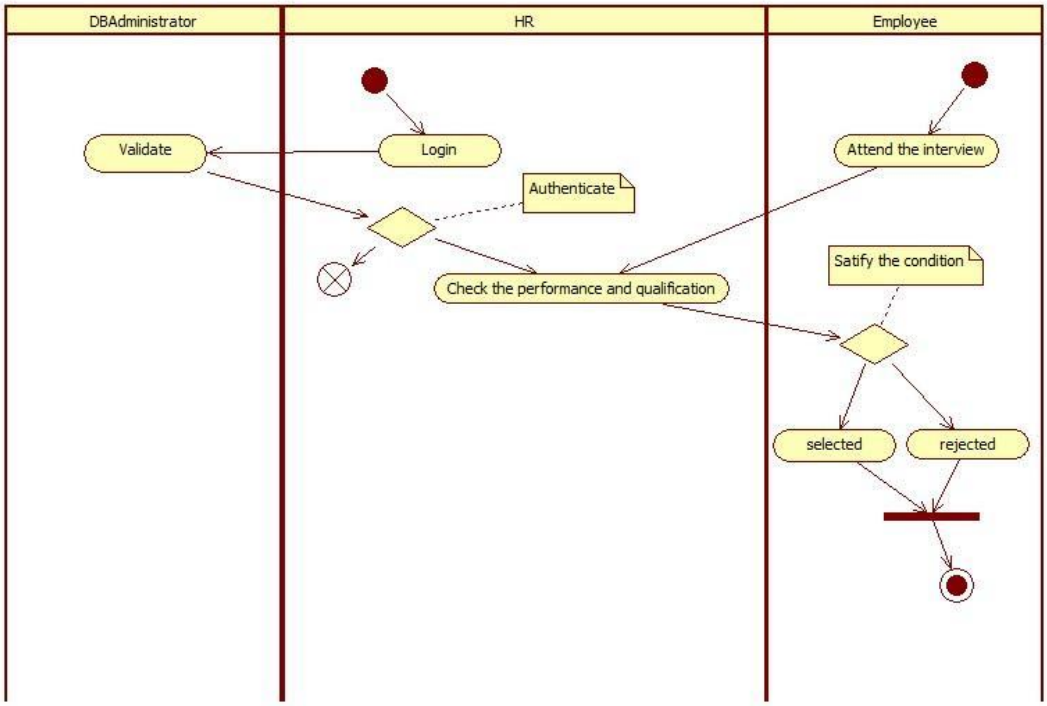
- Salary Processing

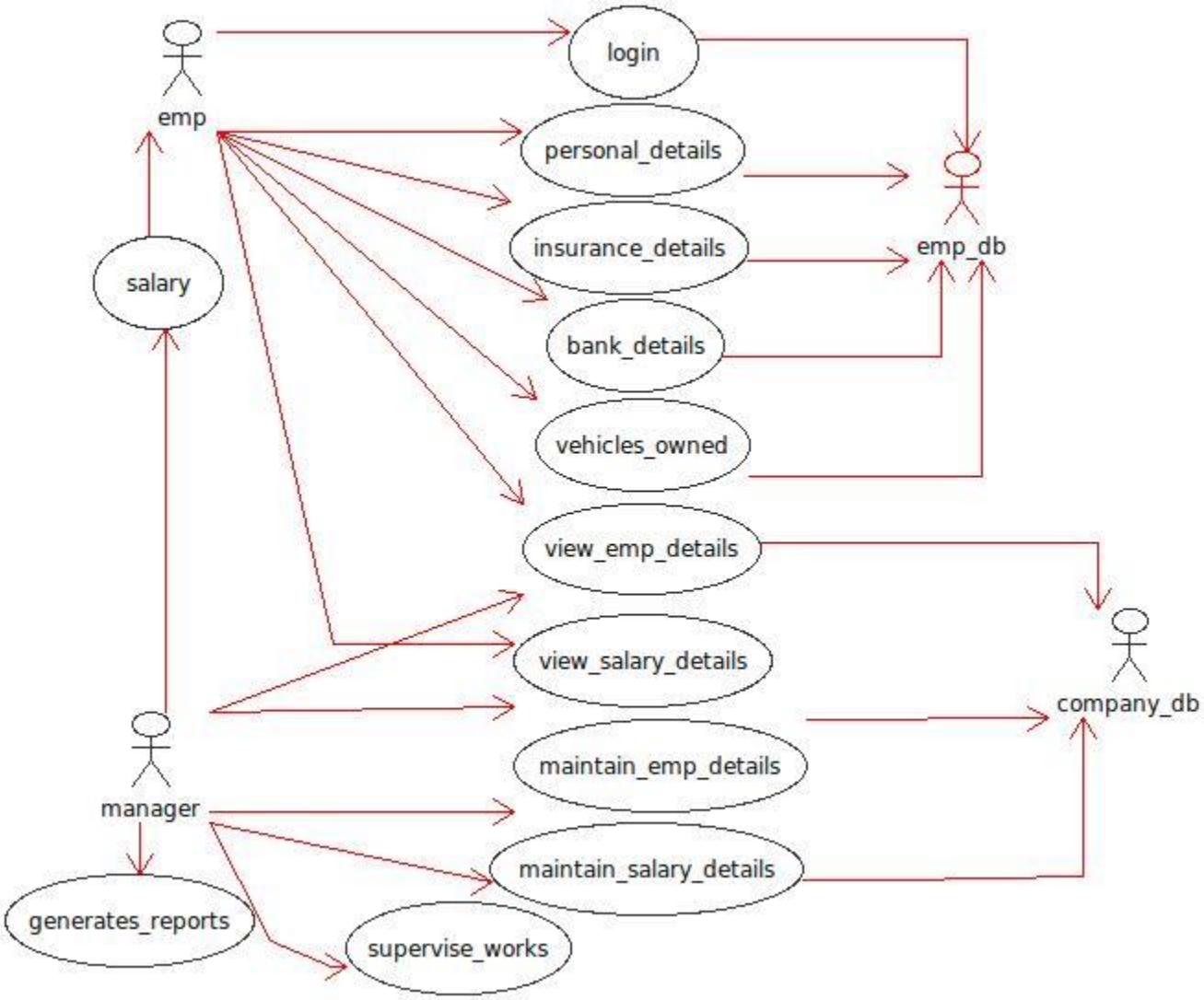
---

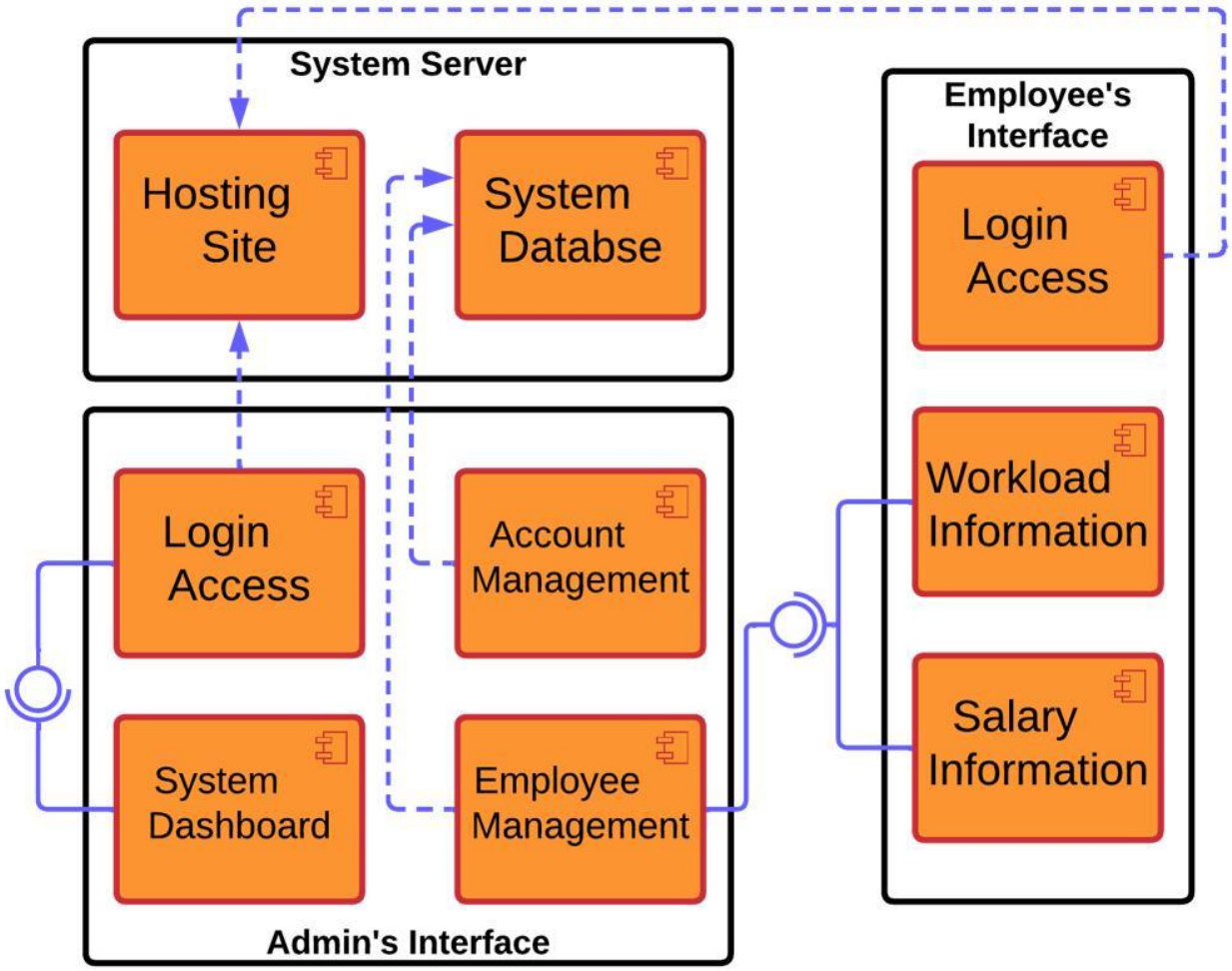
2. ACTIVITY DIAGRAM

Ex:No:
Date

Activity Diagram for Software Personnel Management System




login
emp
personal_details
insurance_details
emp_db
salary
bank_details
vehicles_owned
view_emp_details
view_salary_details
company_db
maintain_emp_details
manager
maintain_salary_details
generates_reports
supervise_works



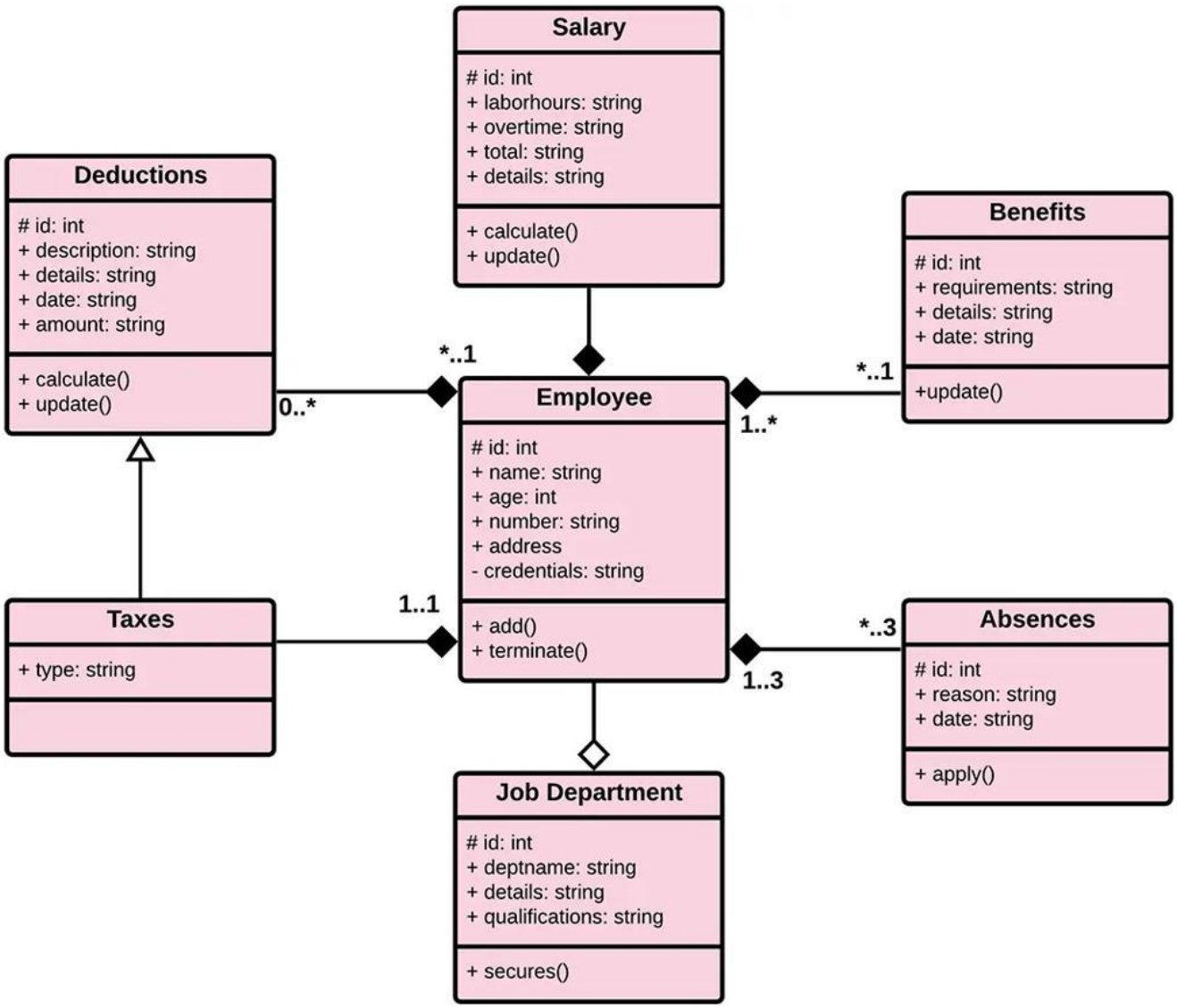
6

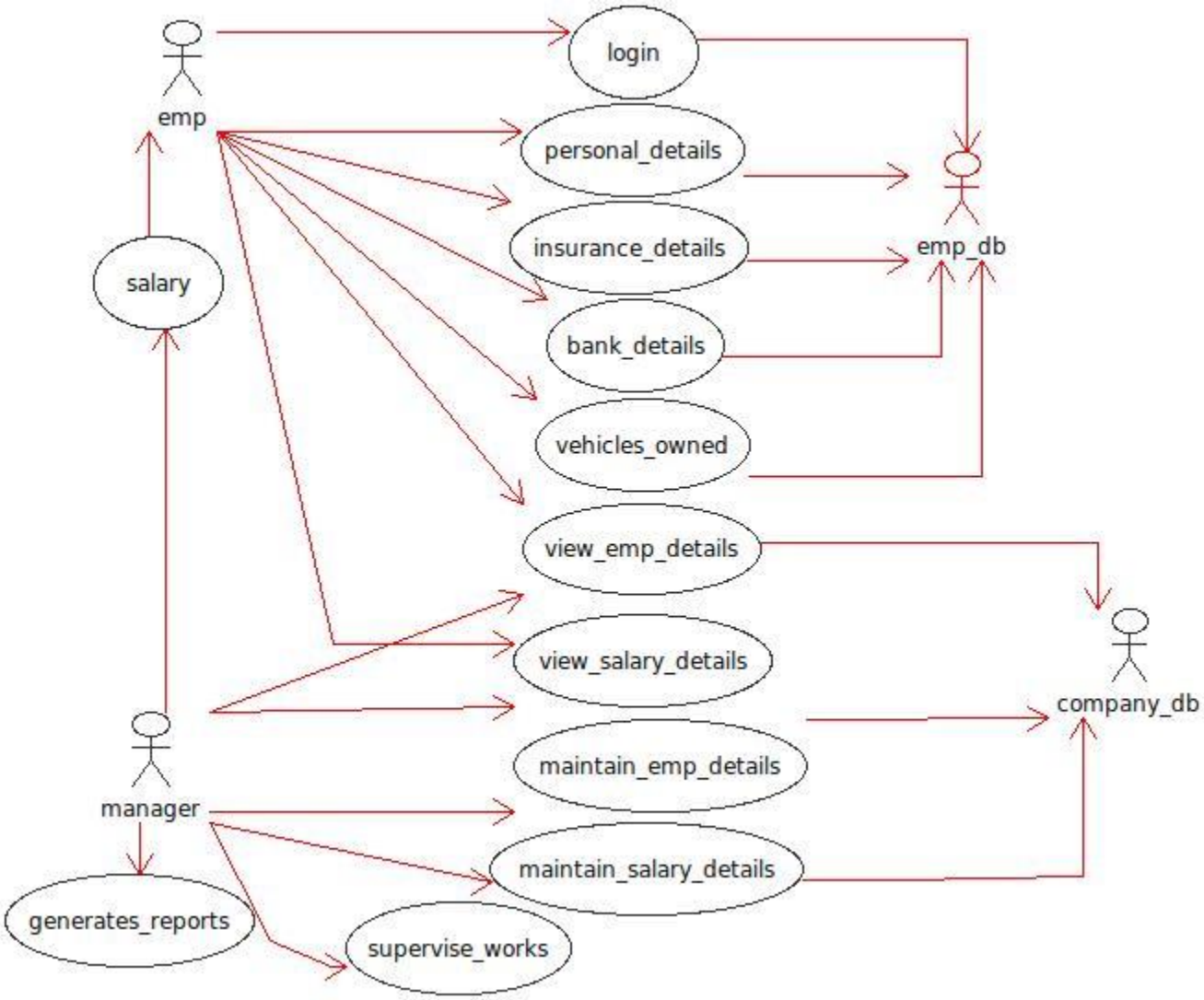
Flow

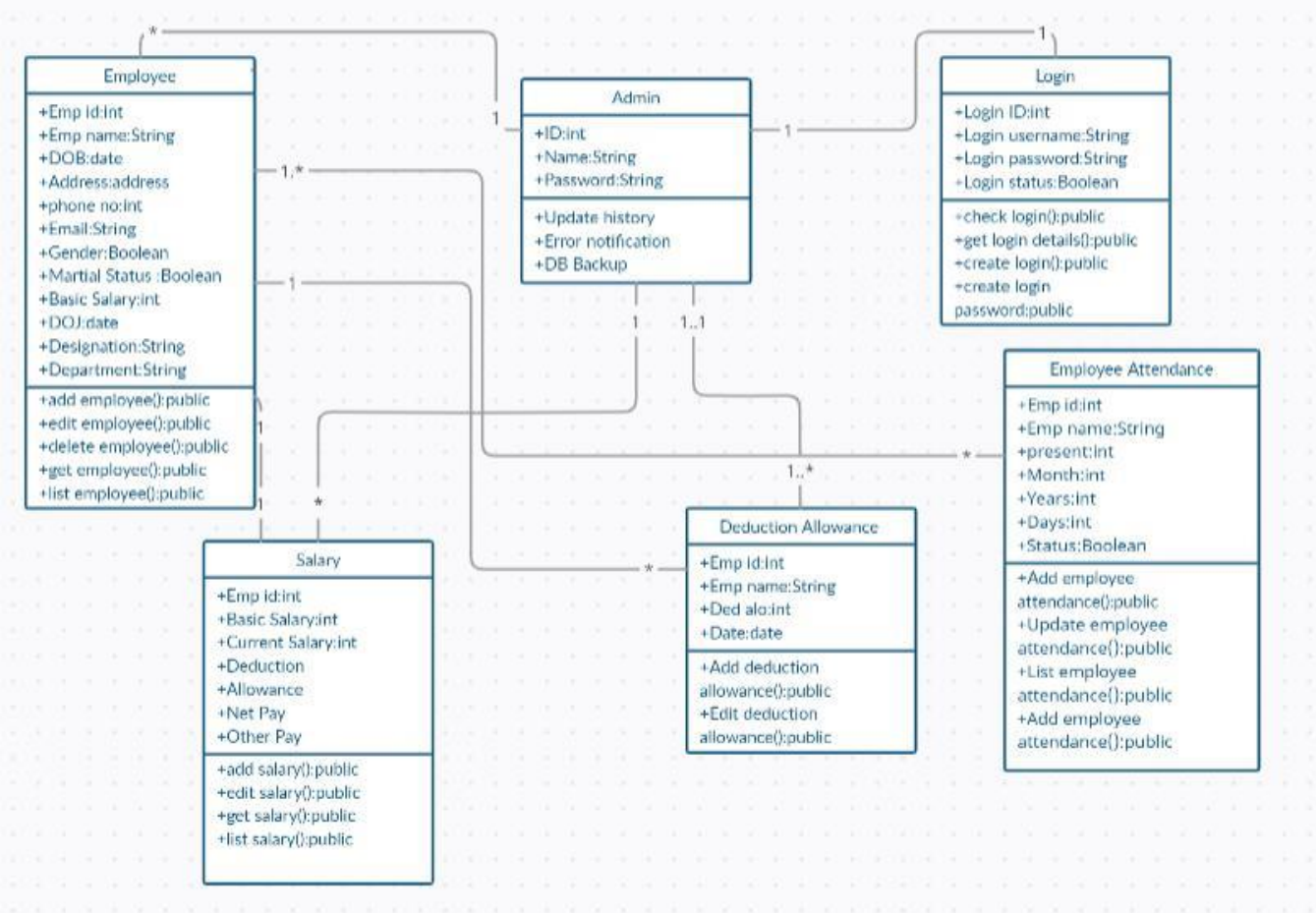
- Login
- Add employee
- Assign project
- Track work
- Generate report

---

3. CLASS DIAGRAM


Salary
id: int
+ laborhours: string
+ overtime: string
+ total: string
+ details: string
+ calculate()
+ update()
Deductions
id: int
+ description: string
+ details: string
+ date: string
+ amount: string
+ calculate()
+ update()
Benefits
id: int
+ requirements: string
+ details: string
+ date: string
+update()
Employee
id: int
+ name: string
+ age: int
+ number: string
+ address
- credentials: string
+ add()
+ terminate()
Taxes
+ type: string
Absences
id: int
+ reason: string
+ date: string
+ apply()
Job Department
id: int
+ deptname: string
+ details: string
+ qualifications: string
+ secures()
*..1
0..*
*..1
1..*
1..1
*..3
1..3


login
emp
personal_details
insurance_details
emp_db
salary
bank_details
vehicles_owned
view_emp_details
view_salary_details
company_db
maintain_emp_details
manager
maintain_salary_details
generates_reports
supervise_works



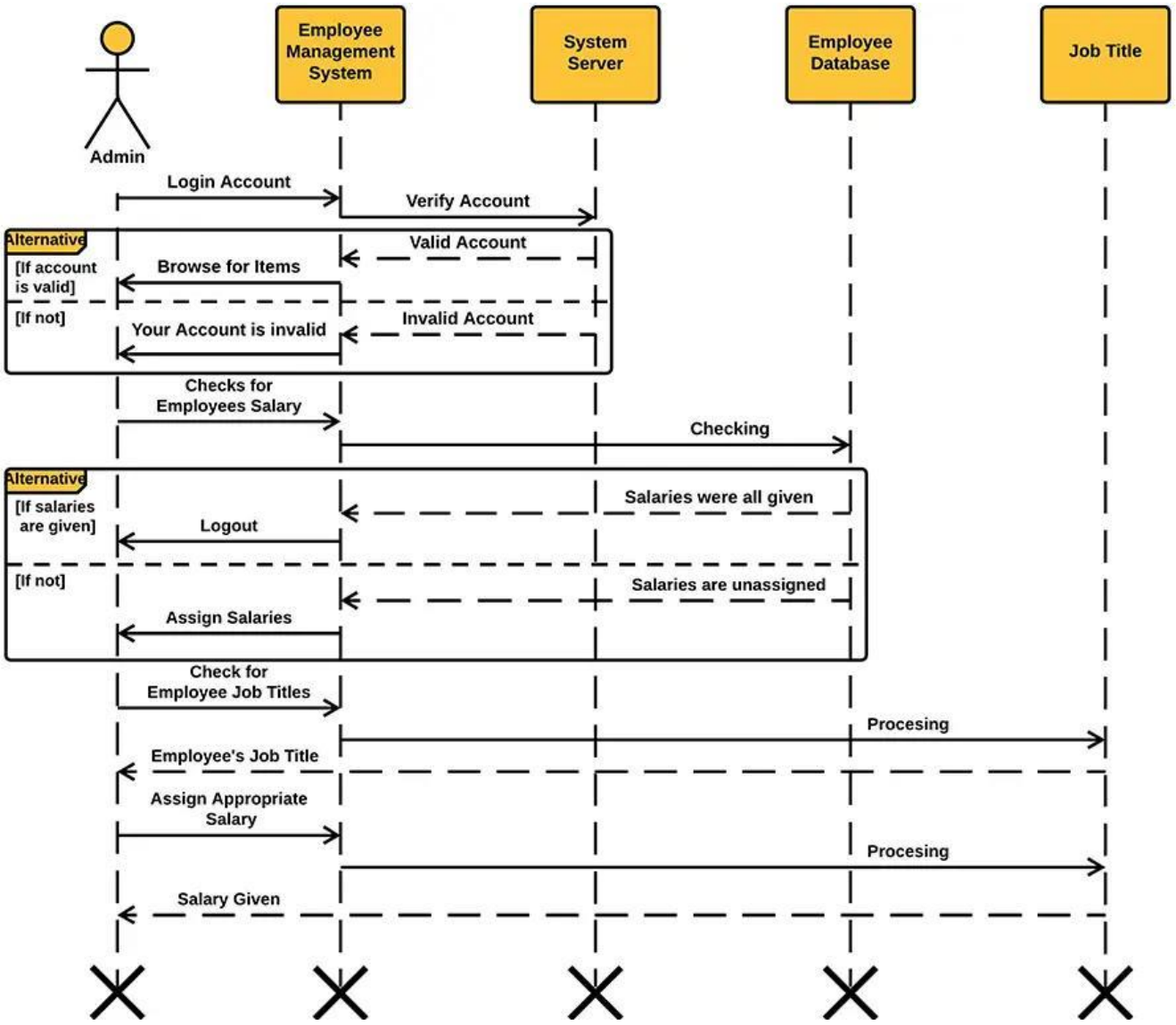
6

Classes

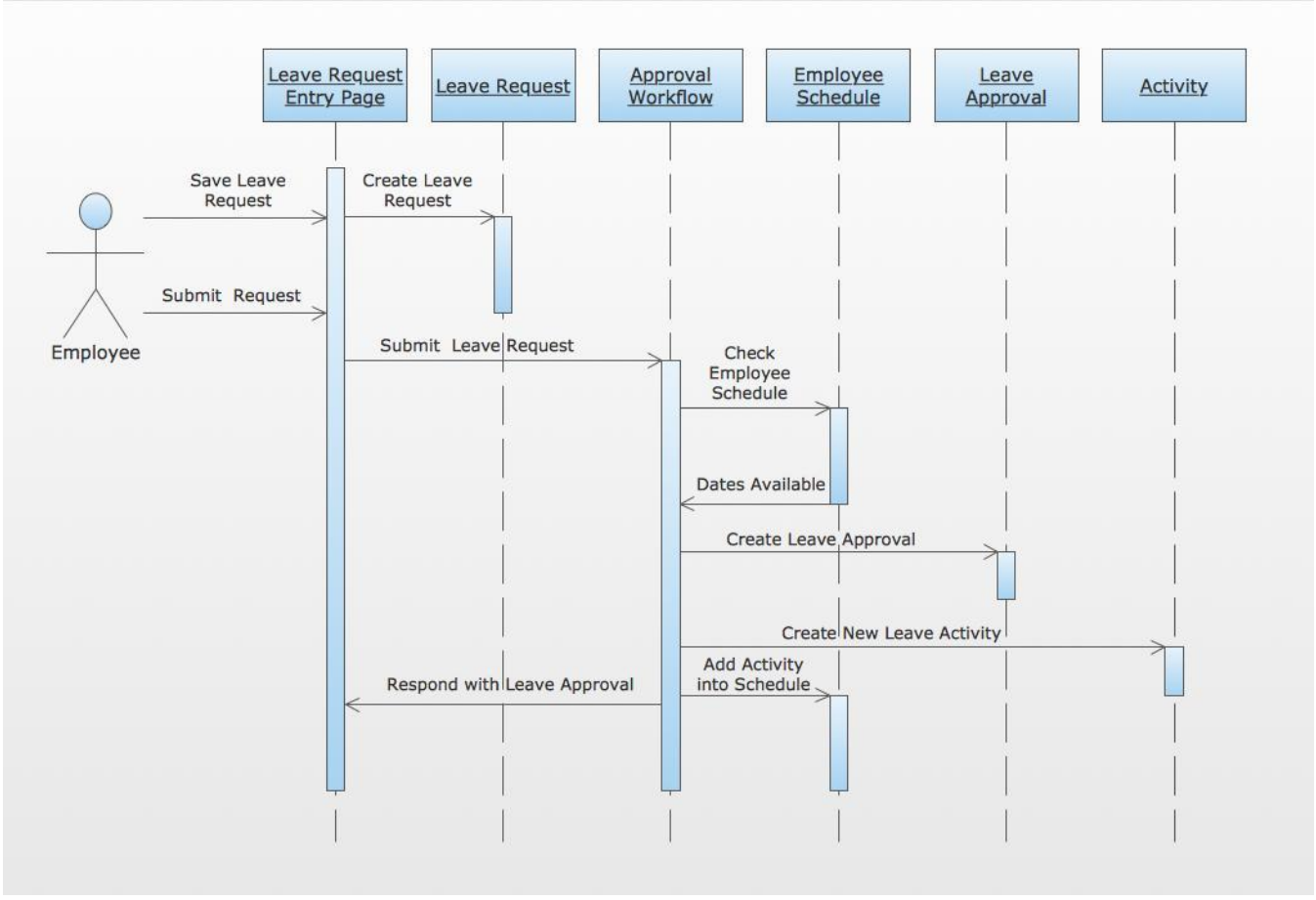
1. Employee
2. Project
3. Attendance
4. Salary
5. Admin

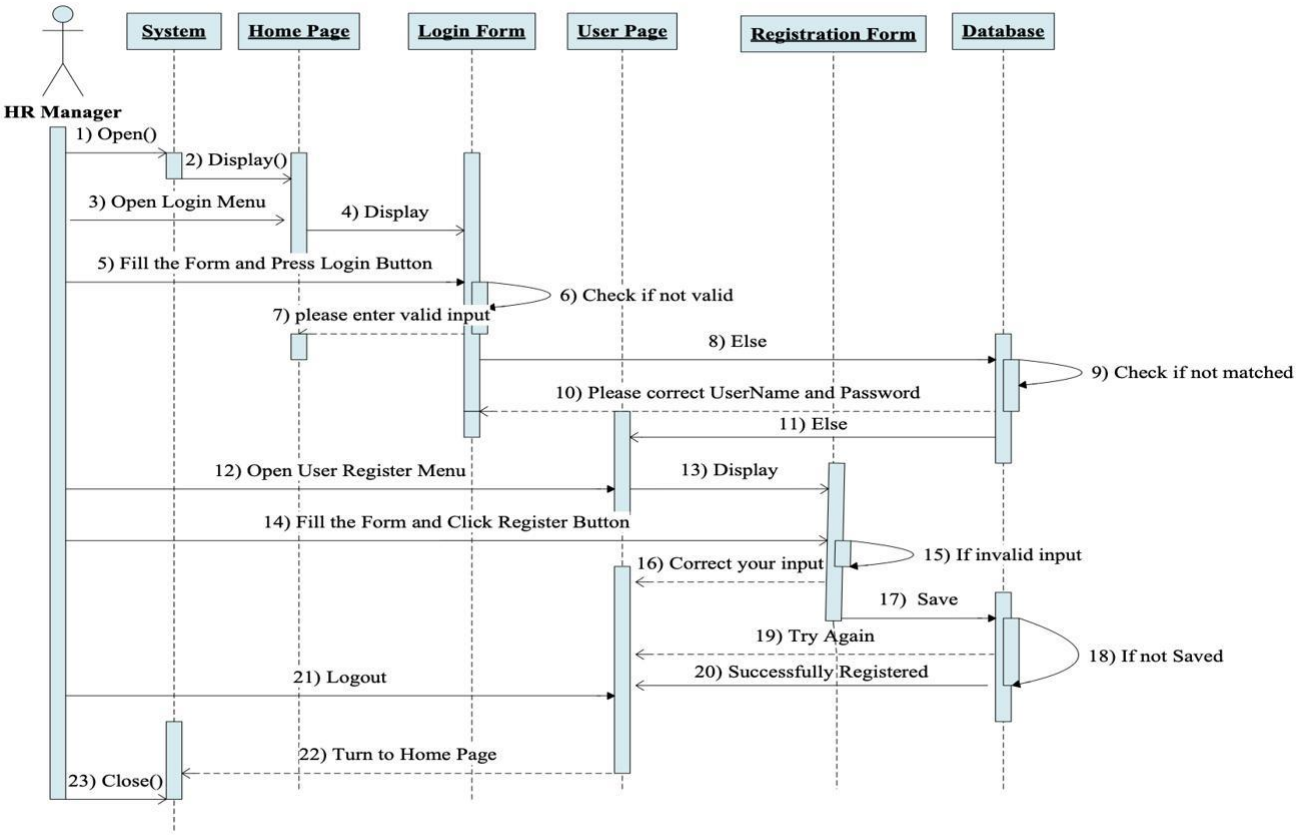
---

4. SEQUENCE DIAGRAM


Admin
Employee Management System
System Server
Employee Database
Job Title
Login Account
Verify Account
Alternative
[If account is valid]
Valid Account
Browse for Items
[If not]
Invalid Account
Your Account is invalid
Checks for Employees Salary
Checking
Alternative
[If salaries are given]
Salaries were all given
Logout
[If not]
Salaries are unassigned
Assign Salaries
Check for Employee Job Titles
Procesing
Employee's Job Title
Assign Appropriate Salary
Procesing
Salary Given

UML Sequence Diagram





7

Description

Shows interaction between:

- Employee
- Admin
- System

---

5. COLLABORATION DIAGRAM

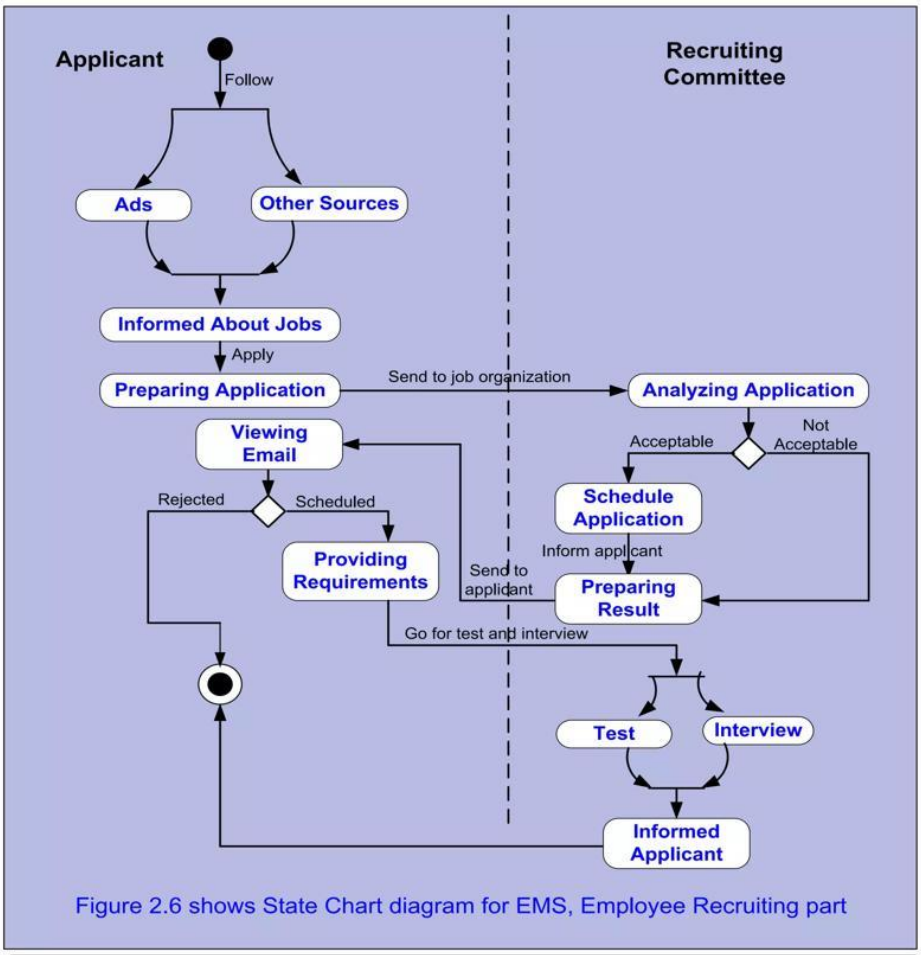


Figure 2.6 shows State Chart diagram for EMS, Employee Recruiting part

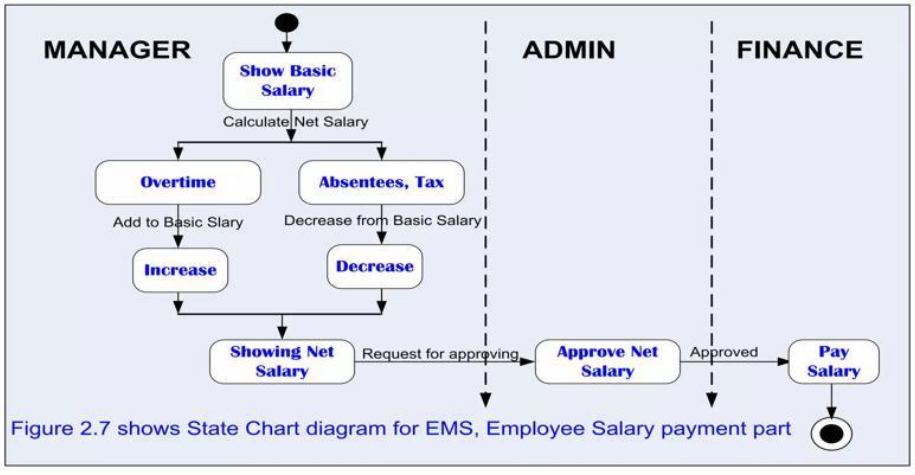
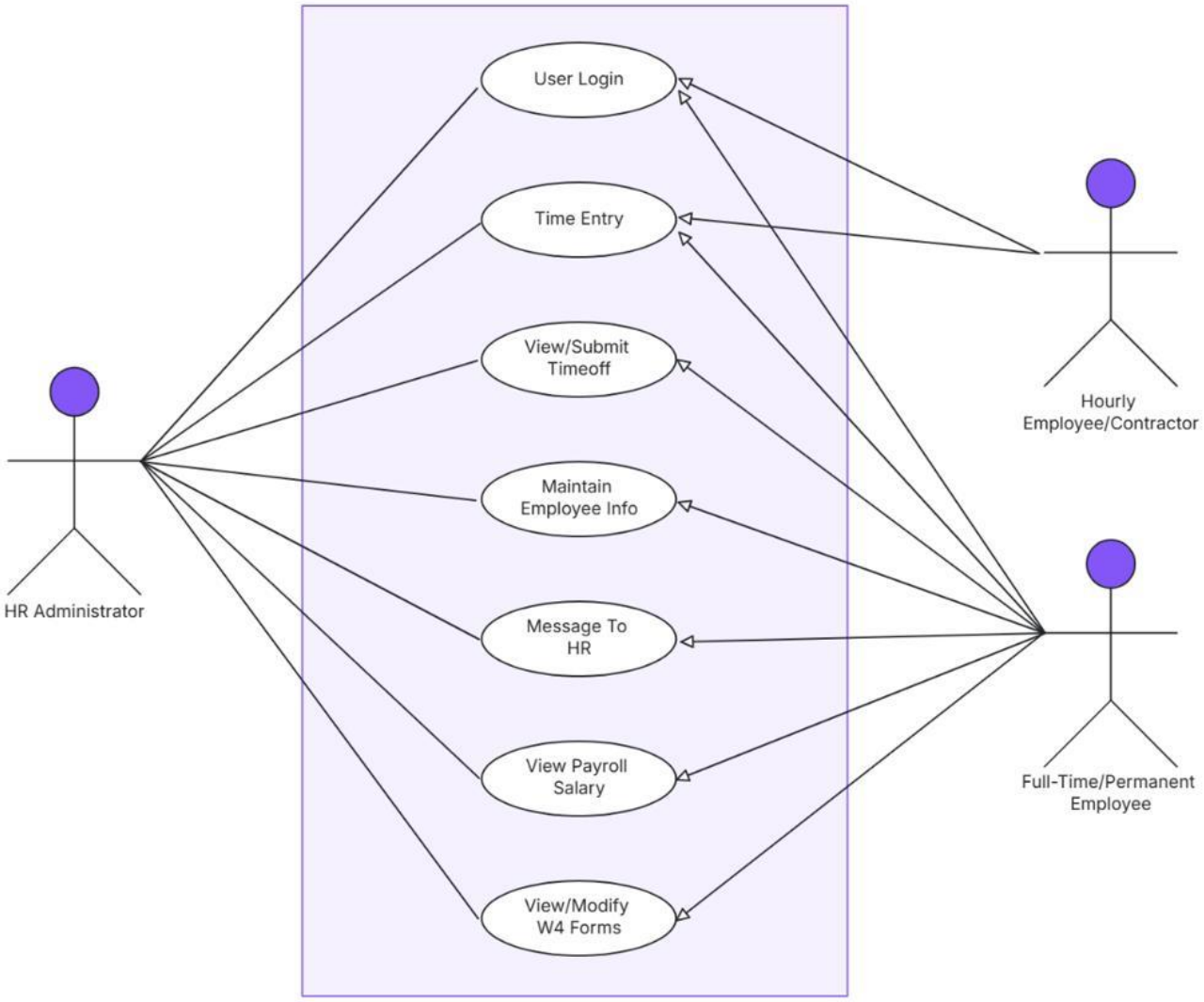
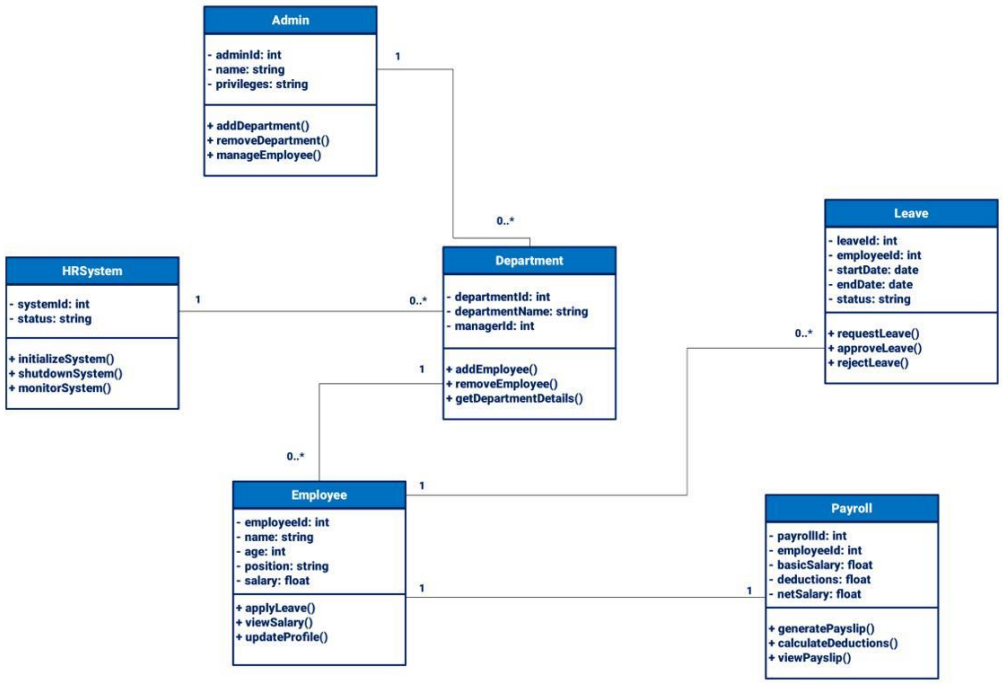


Figure 2.7 shows State Chart diagram for EMS, Employee Salary payment part

Human Resource Management

UML Diagram Template





---

6. STATECHART DIAGRAM

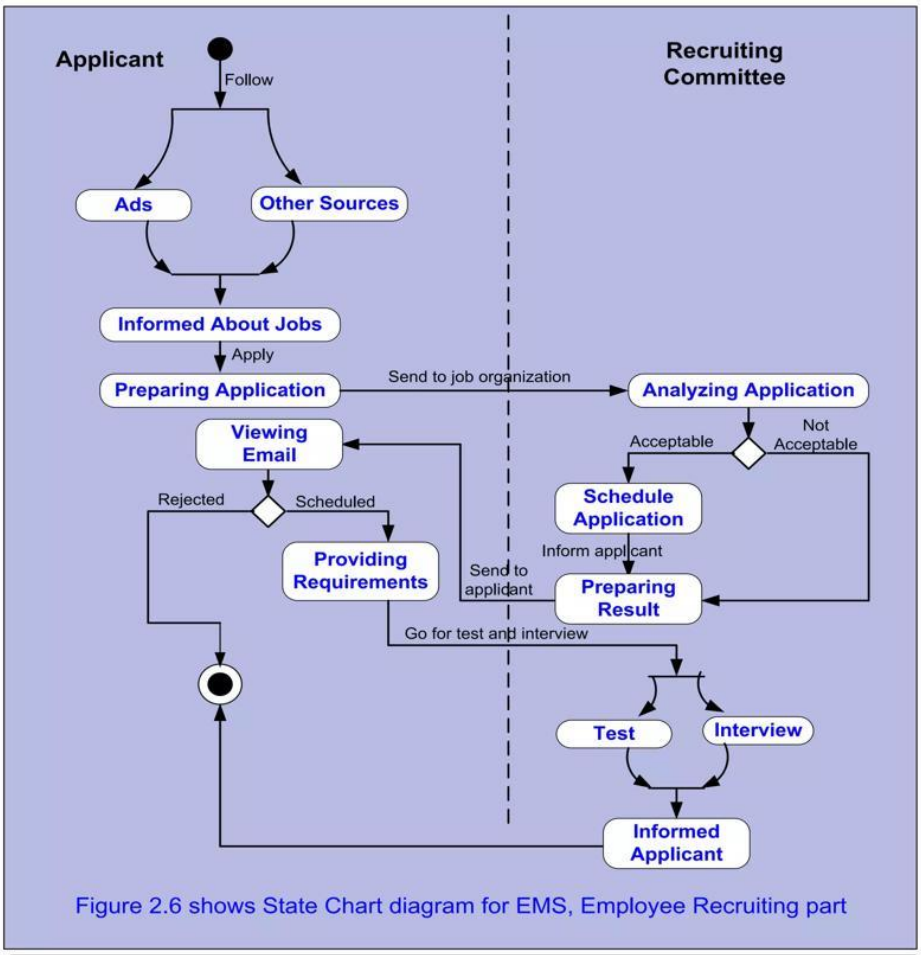


Figure 2.6 shows State Chart diagram for EMS, Employee Recruiting part

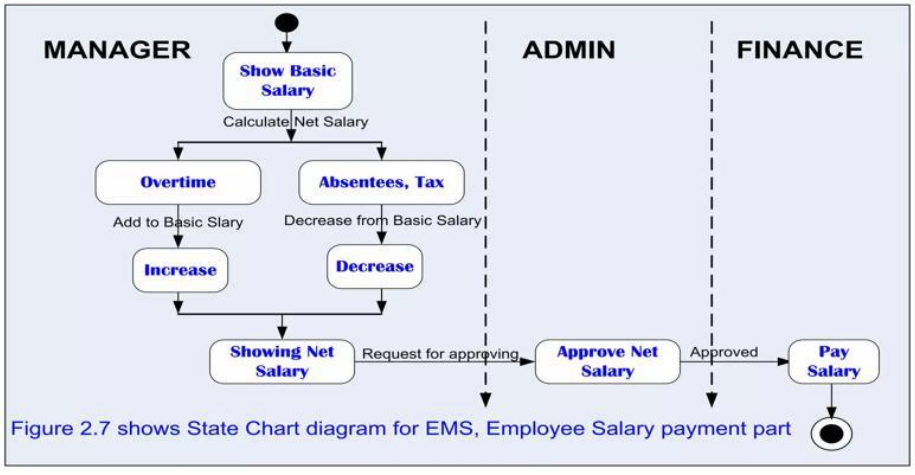
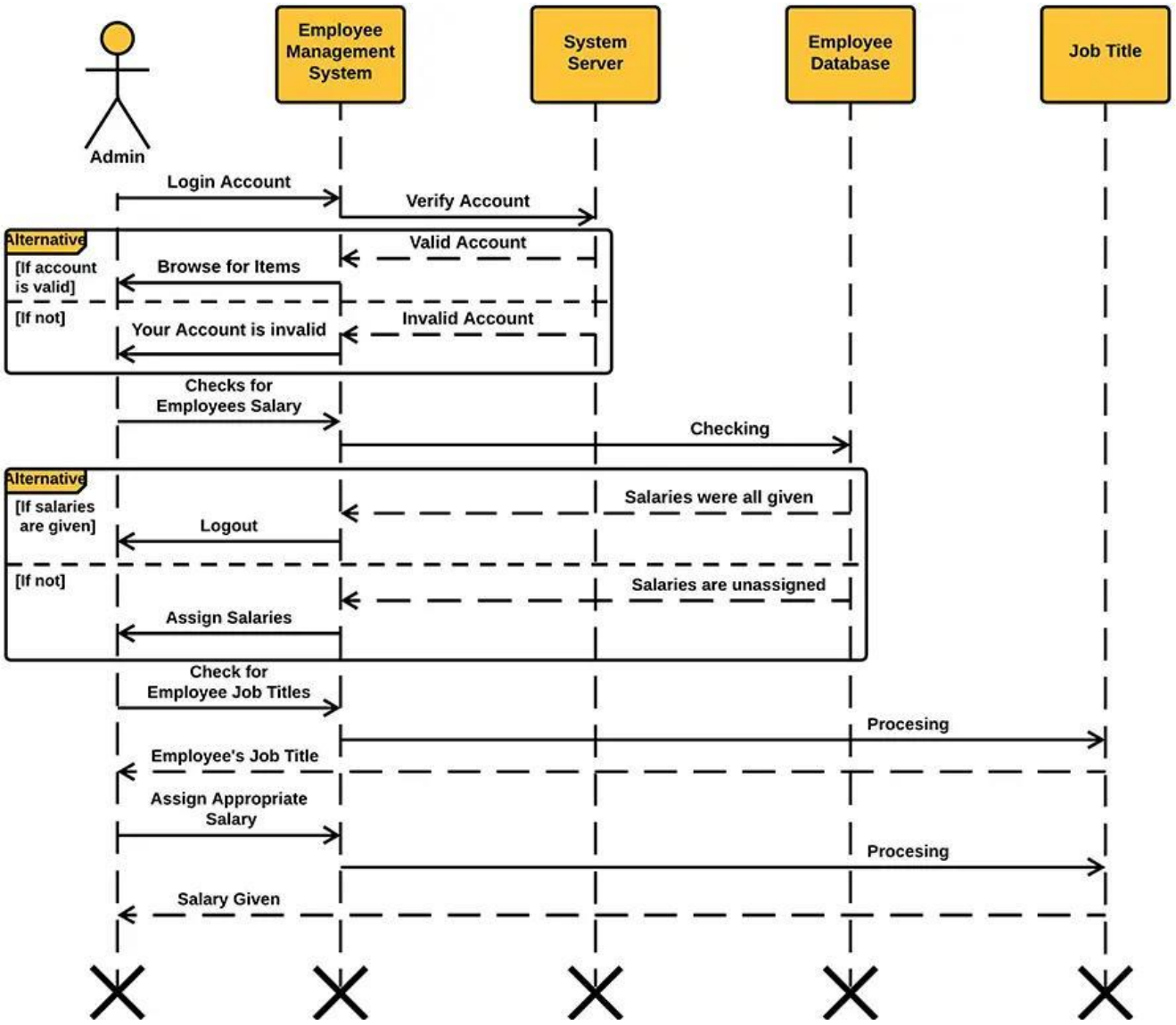
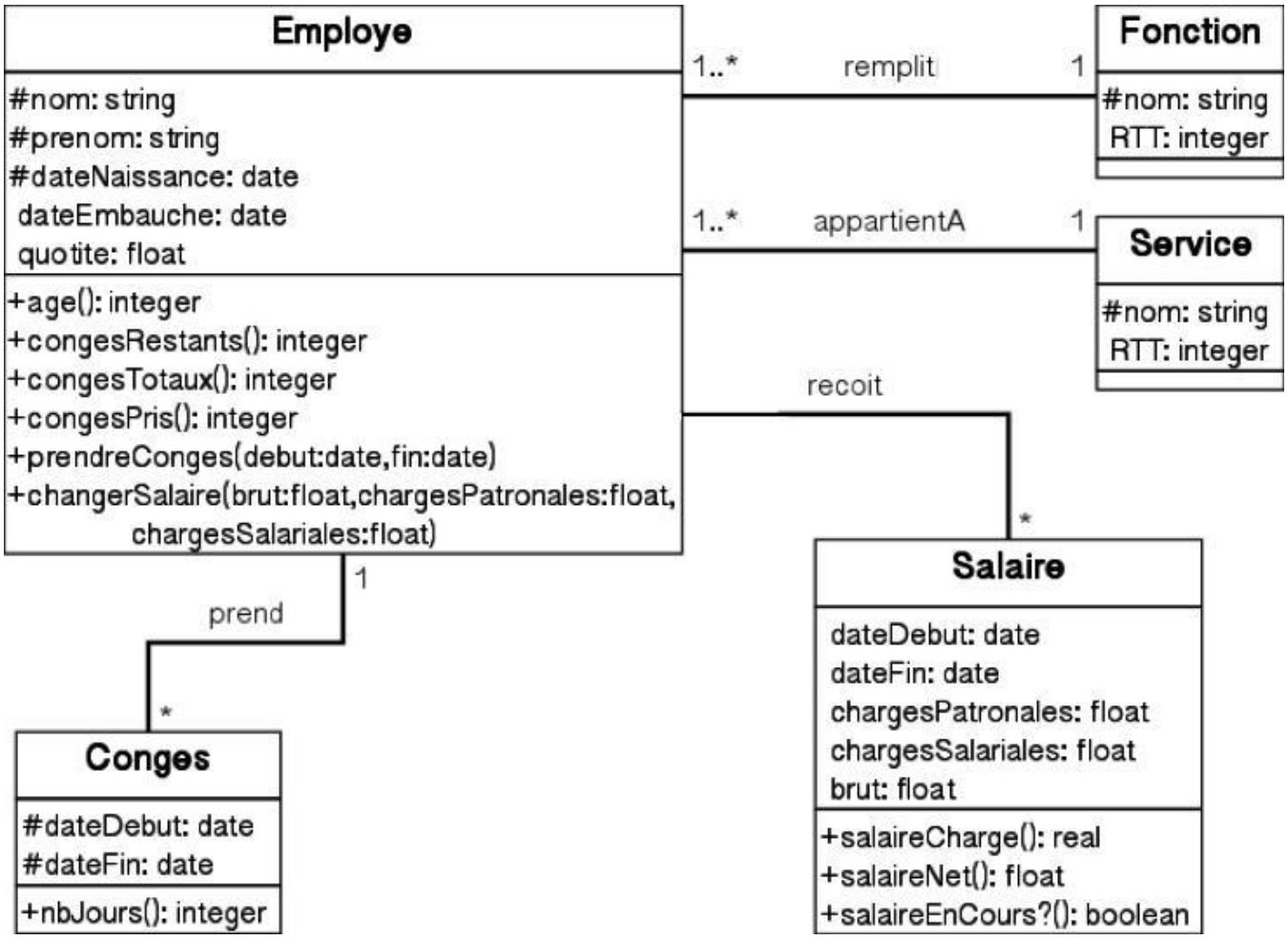


Figure 2.7 shows State Chart diagram for EMS, Employee Salary payment part


Admin
Employee Management System
System Server
Employee Database
Job Title
Login Account
Verify Account
Alternative
[If account is valid]
Valid Account
Browse for Items
[If not]
Invalid Account
Your Account is invalid
Checks for Employees Salary
Checking
Alternative
[If salaries are given]
Salaries were all given
Logout
[If not]
Salaries are unassigned
Assign Salaries
Check for Employee Job Titles
Procesing
Employee's Job Title
Assign Appropriate Salary
Procesing
Salary Given

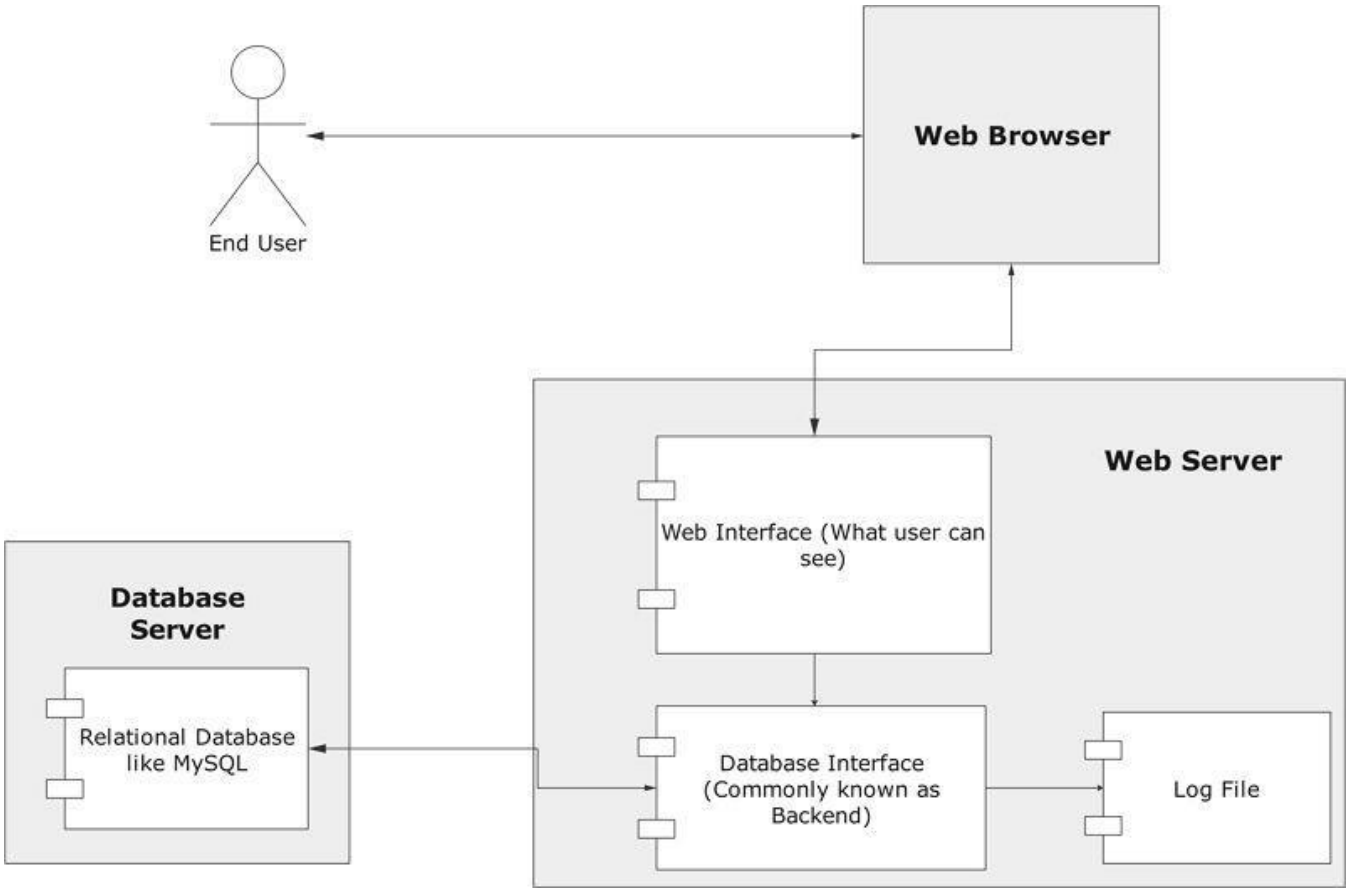


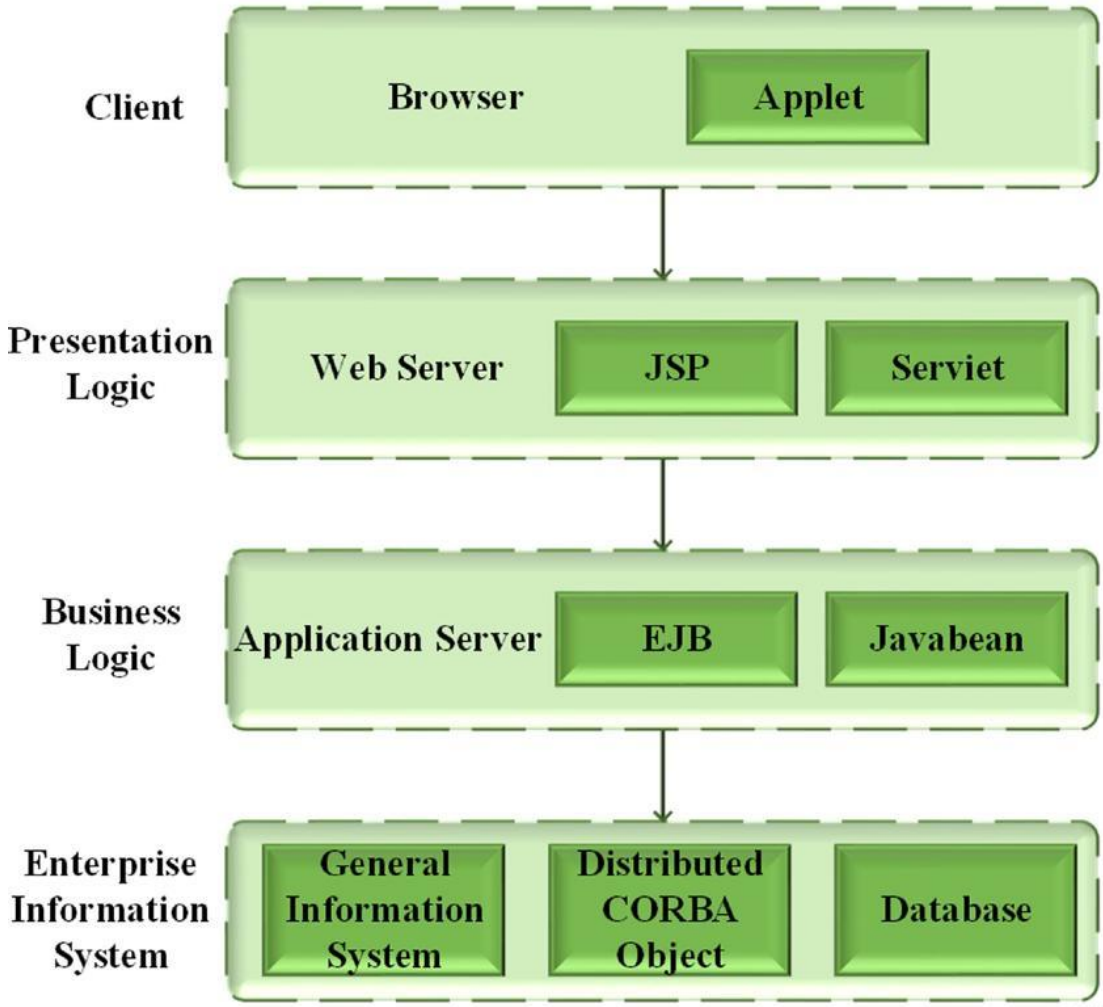
States

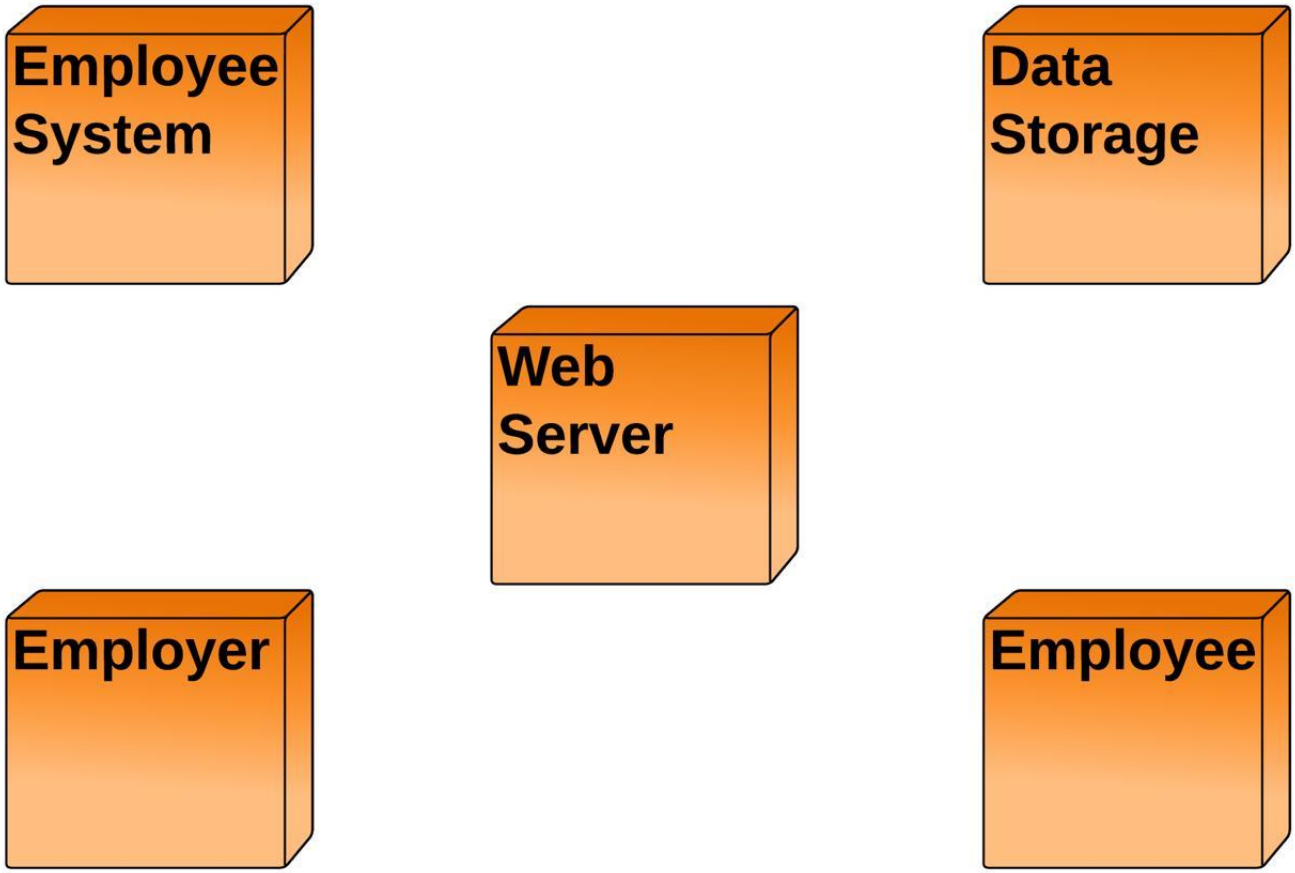
- Registered
- Assigned
- Working
- Completed
- Resigned

---

7. DEPLOYMENT DIAGRAM

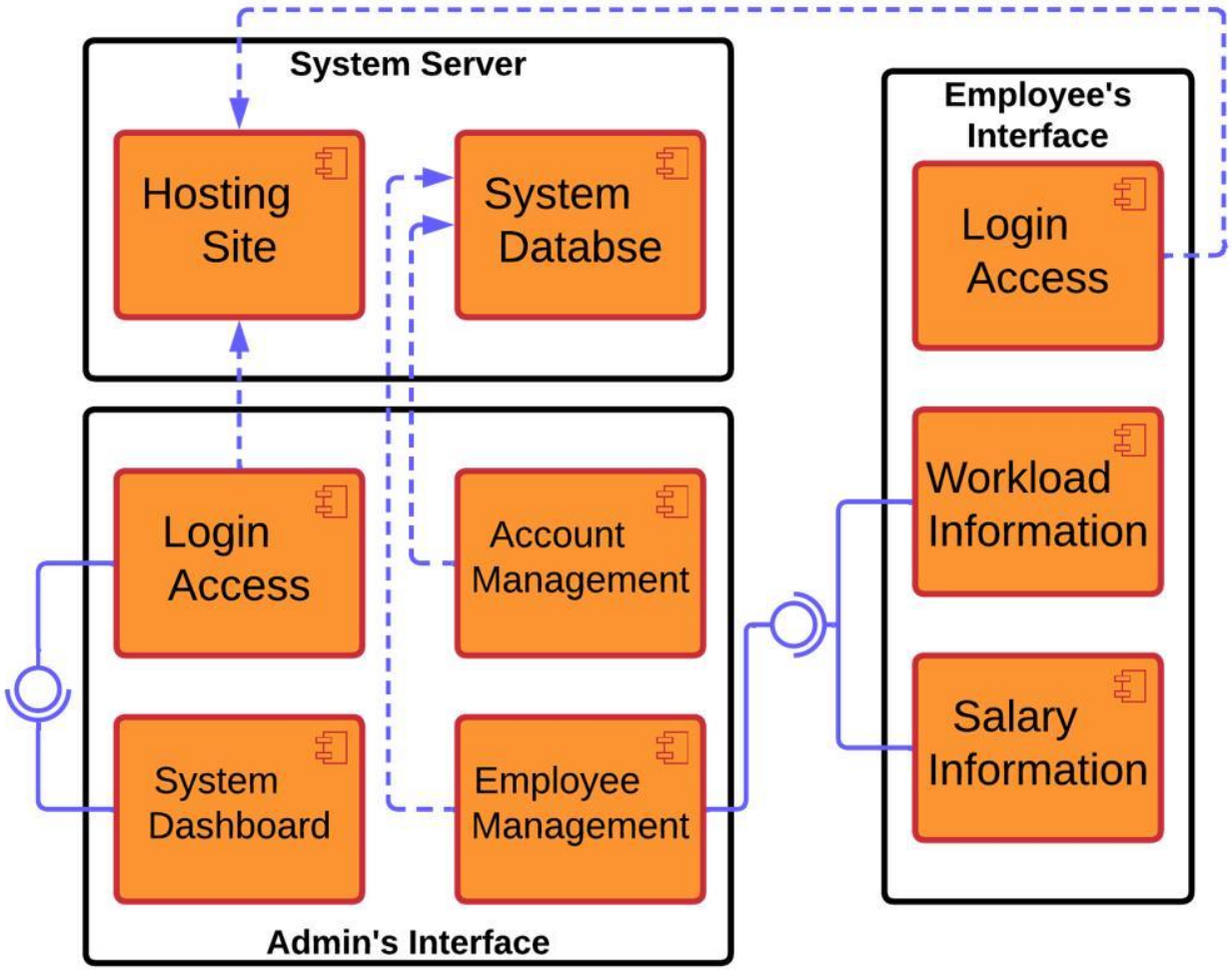

End User
Web Browser
Web Server
Web Interface (What user can see)
Database Server
Relational Database like MySQL
Database Interface (Commonly known as Backend)
Log File





---

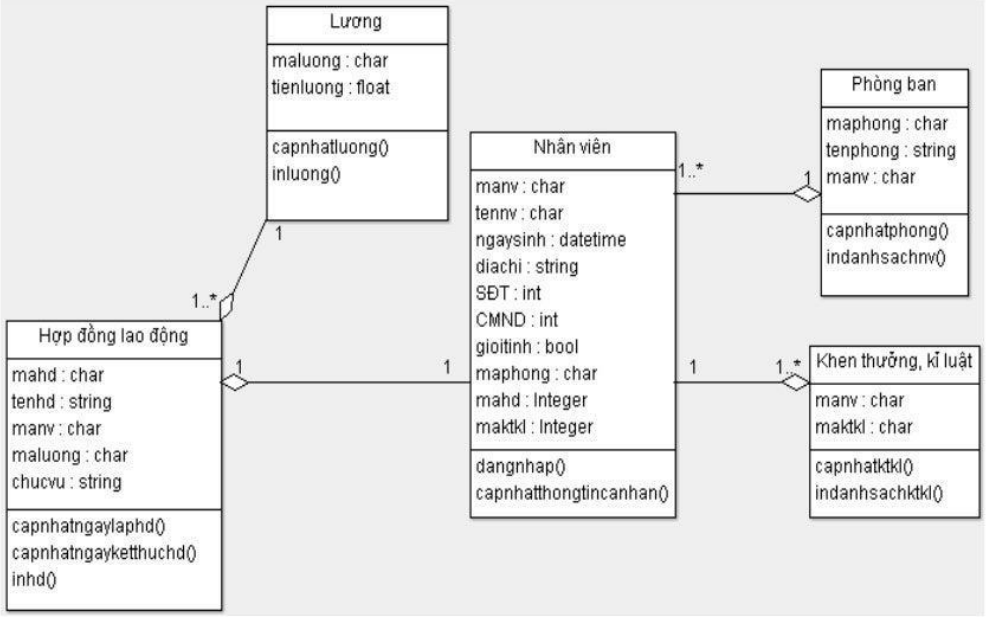
8. COMPONENT DIAGRAM


System Server
Hosting Site
System Databse
Employee's Interface
Login Access
Workload Information
Salary Information
Login Access
Account Management
System Dashboard
Employee Management
Admin's Interface

III. *Class Diagram*

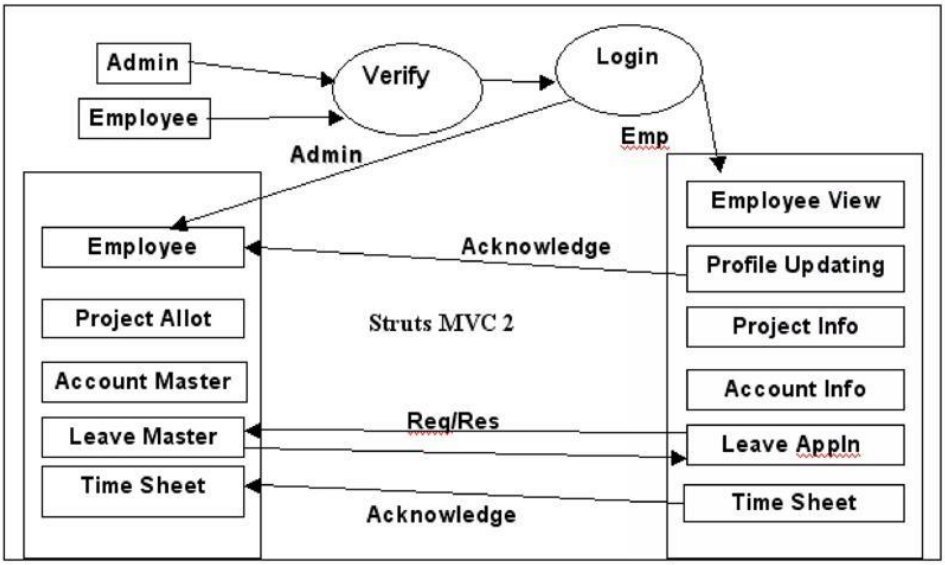
- Khái niệm:
- Class diagram là sơ đồ dùng để mô tả một hệ thống bằng các khái niệm lớp (bao gồm các thuộc tính và phương thức) và mối quan hệ giữa các lớp với nhau.

1. Class Diagram của hệ thống quản lý nhân sự và tiền lương



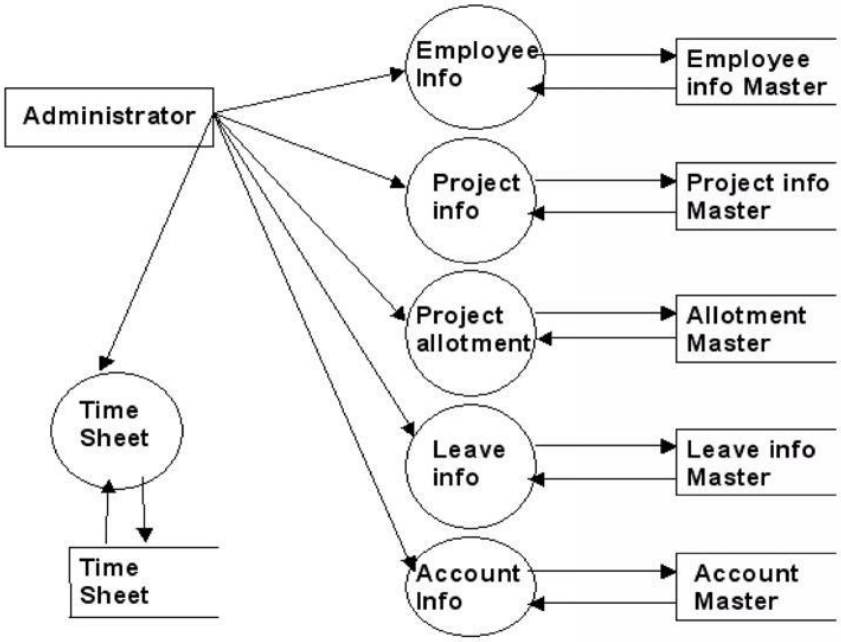
IV. *State diagram:*

- Khái niệm:
- State diagram (còn có tên gọi là state machine) là sơ đồ dùng để mô tả các trạng thái có thể có của một đối tượng và sự chuyển đổi giữa các trạng thái (transition) khi có các sự kiện (event) tác động.
- Đối với các đối tượng có nhiều trạng thái thì state diagram là sự lựa chọn tốt nhất giúp hiểu rõ hơn về hệ thống.



Architectural Design

ADMINISTRATOR MODULE



CONCLUSION

The Software Personnel Management System:

- Improves employee management
- Reduces manual workload
- Enhances productivity
- Provides accurate reporting

EXPERIMENT–8: CREDIT CARD PROCESSING SYSTEM

AIM

To design and develop a system to process credit card transactions securely.

(I) PROBLEM STATEMENT

The Credit Card Processing System is used to handle online and offline card transactions.

- Validates card details
- Processes payments
- Communicates with bank server
- Generates transaction status

The system ensures **secure and fast financial transactions**.

(II) SOFTWARE REQUIREMENT SPECIFICATION

INTRODUCTION

Acts as an interface between **Customer**, **Merchant**, and **Bank**.

PURPOSE

- Replace manual payment processing
- Ensure fast and secure transactions

SCOPE

- Card validation
- Payment authorization
- Transaction processing
- Receipt generation

ACTORS

- Customer
- Merchant
- Bank

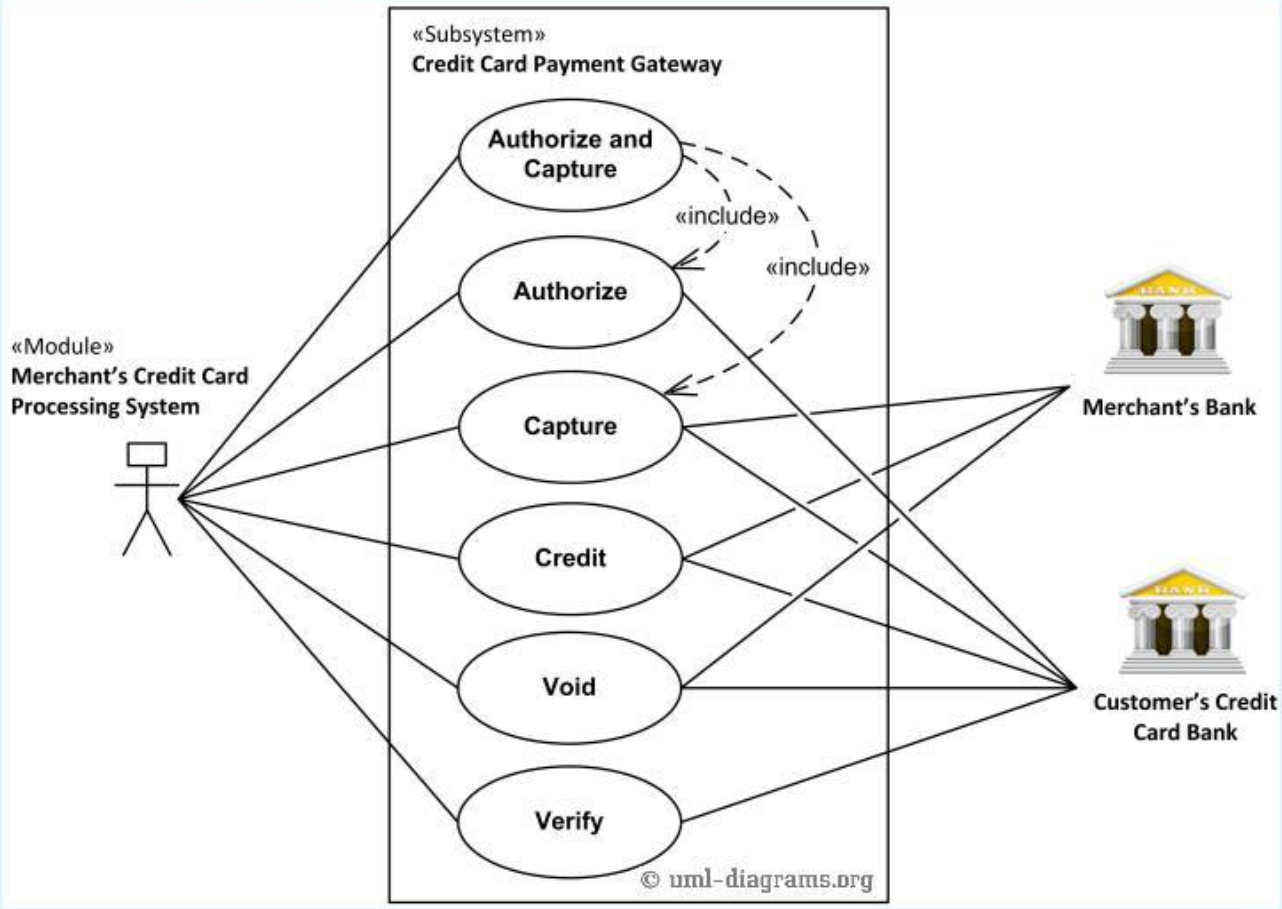
UML DIAGRAMS

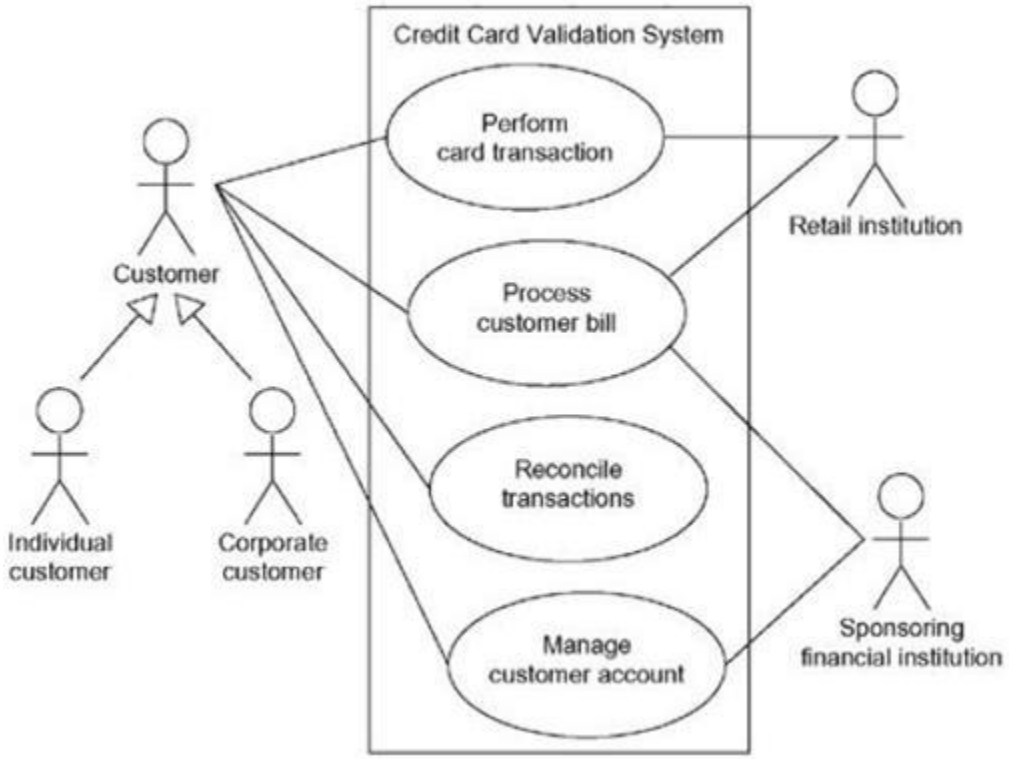
1. USE CASE DIAGRAM

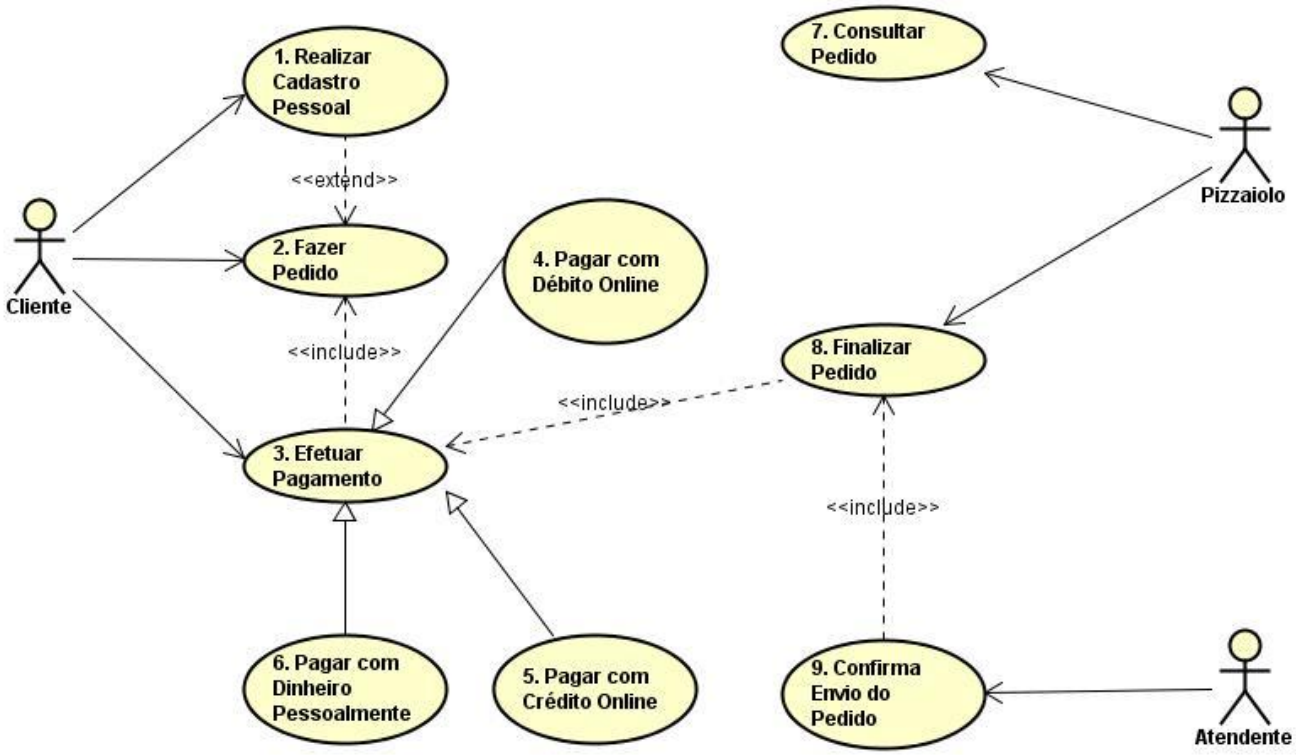
based on your understanding of the use case diagram below, which is for a credit cards processing system:

1- Draw the miss-use case diagram for the same system.

2-Draw DFD level one for the credit card processing system.




Credit Card Validation System
Perform card transaction
Process customer bill
Reconcile transactions
Manage customer account
Customer
Individual customer
Corporate customer
Retail institution
Sponsoring financial institution

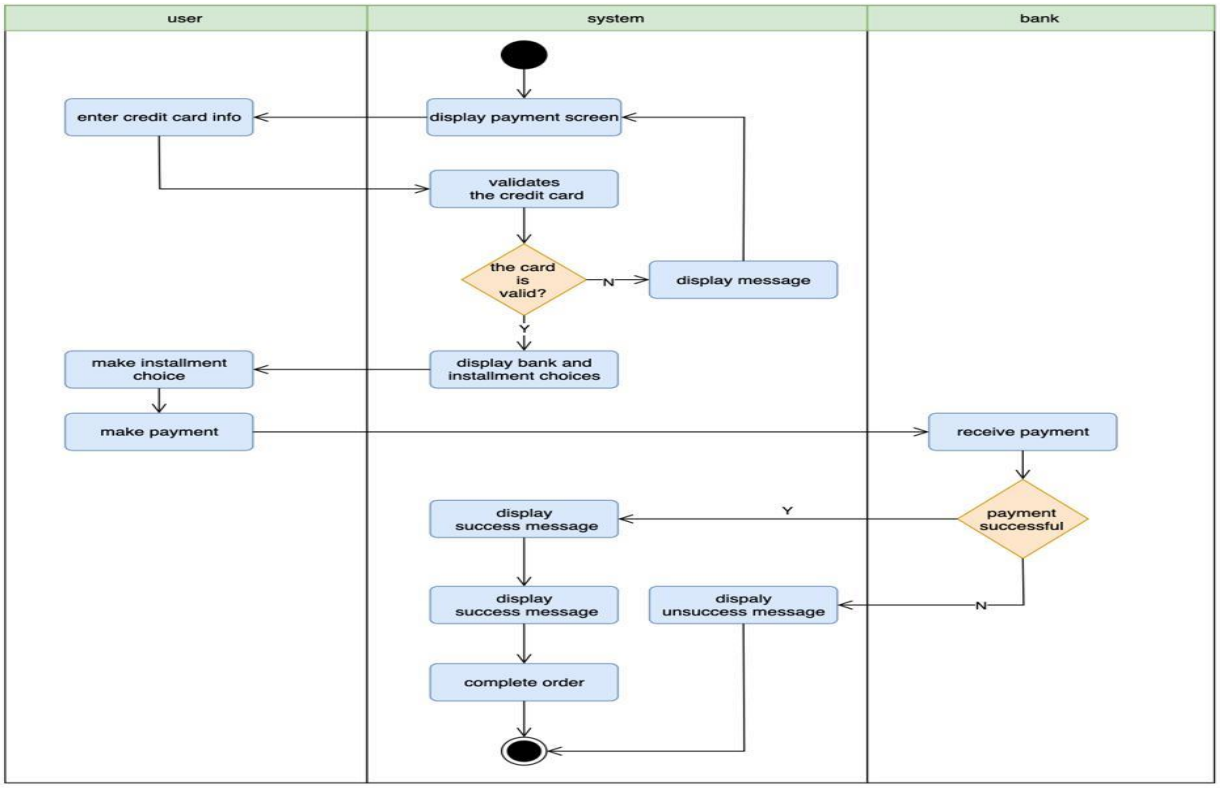

1. Realizar Cadastro Pessoal
<<extend>>
2. Fazer Pedido
Cliente
4. Pagar com Débito Online
<<include>>
3. Efetuar Pagamento
6. Pagar com Dinheiro Pessoalmente
5. Pagar com Crédito Online
7. Consultar Pedido
Pizzaiolo
8. Finalizar Pedido
<<include>>
<<include>>
9. Confirma Envio do Pedido
Atendente

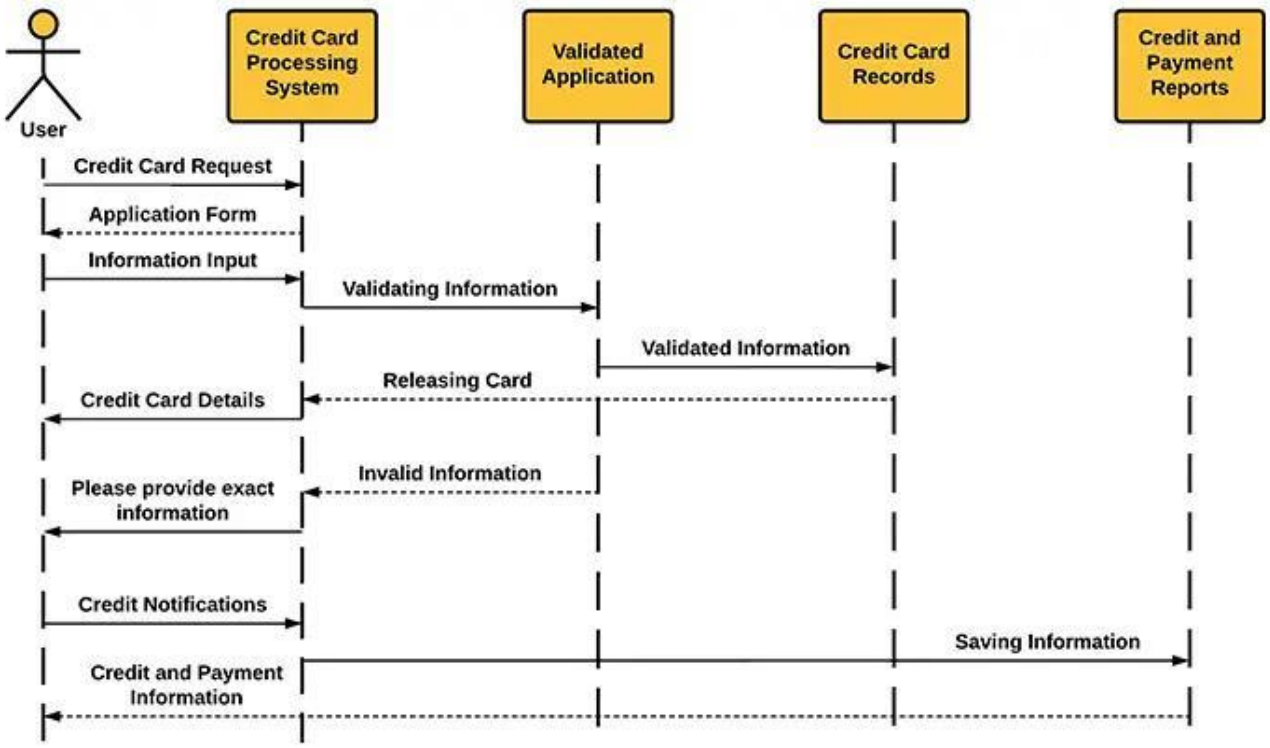
Use Cases

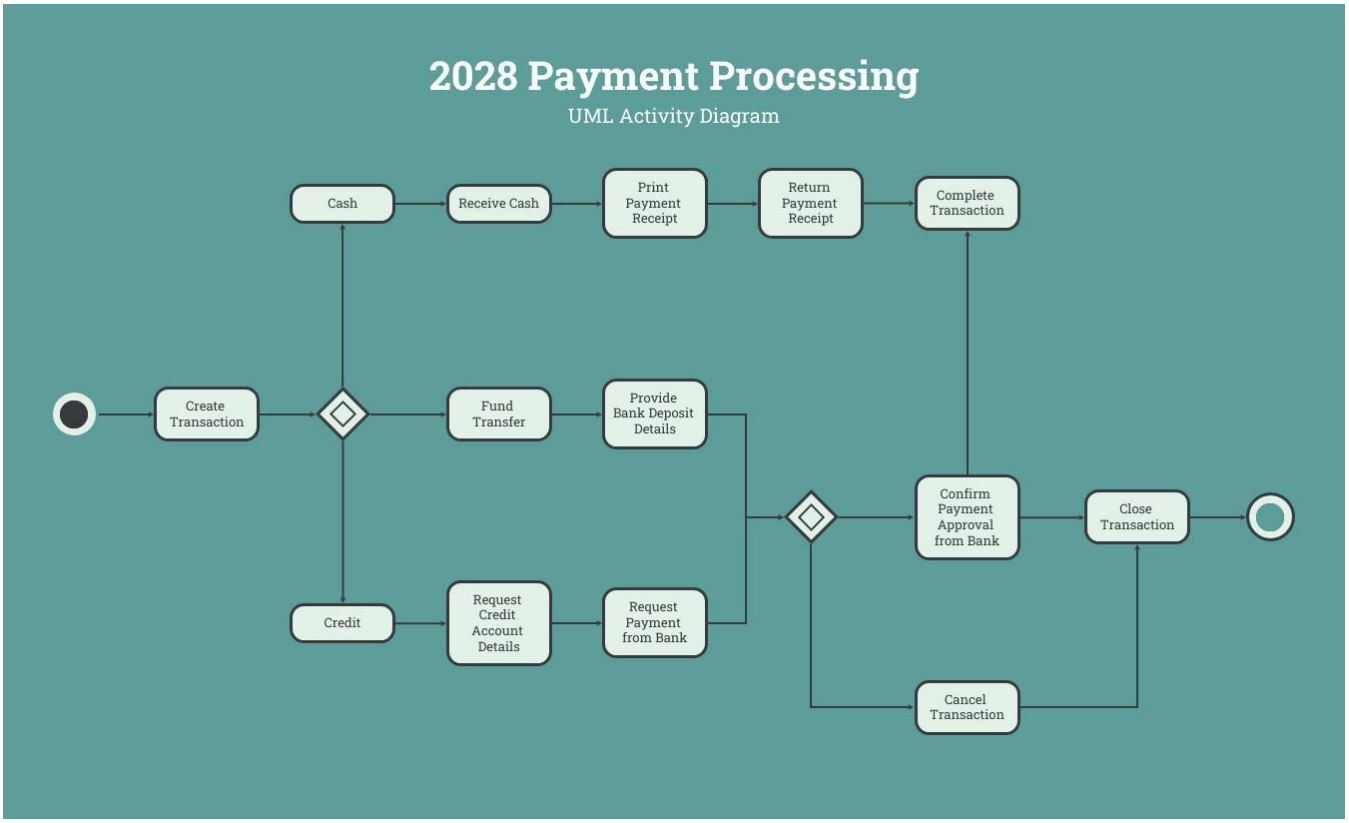
- Enter Card Details
- Validate Card
- Process Payment
- Generate Receipt

---

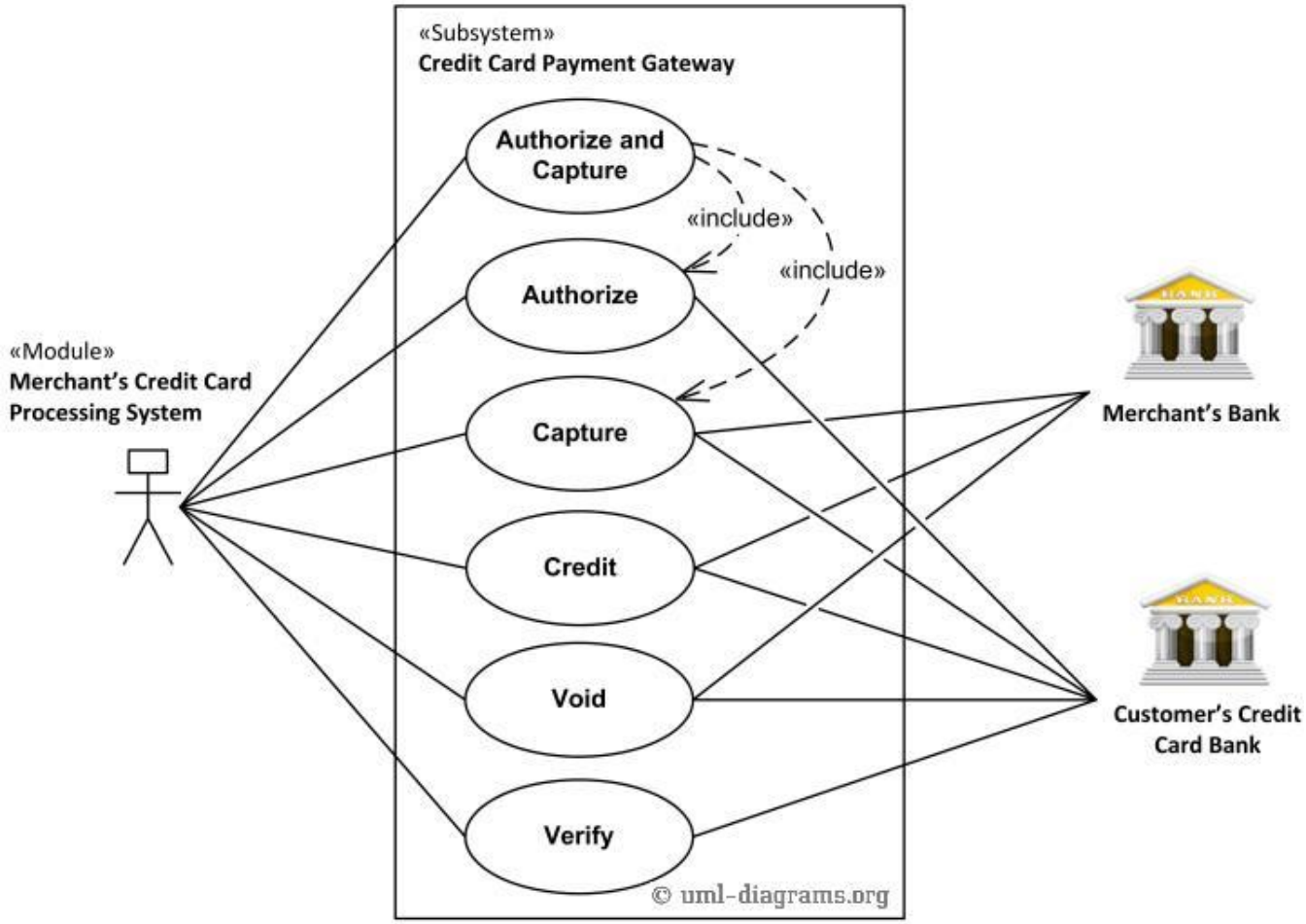
2. ACTIVITY DIAGRAM

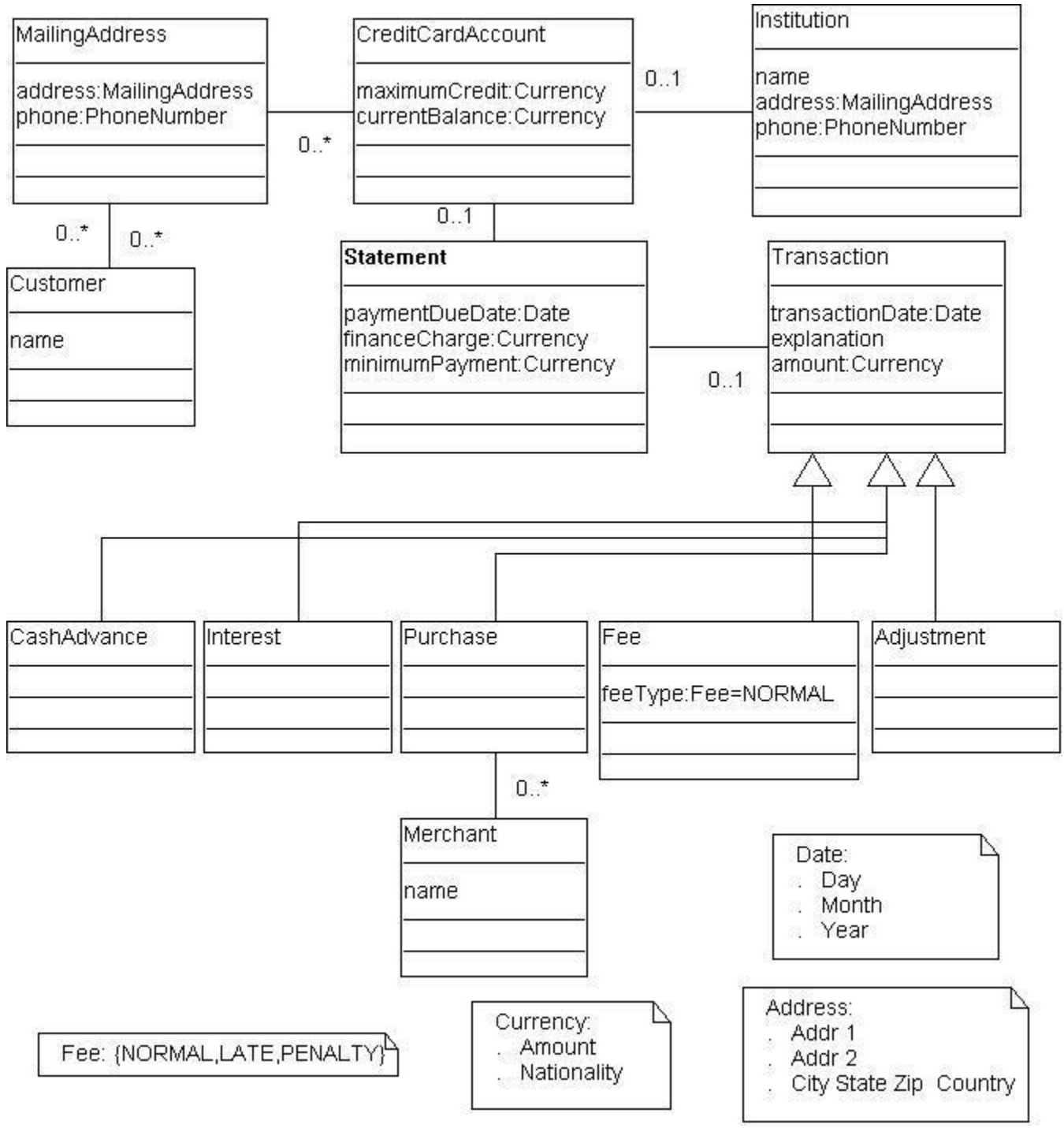


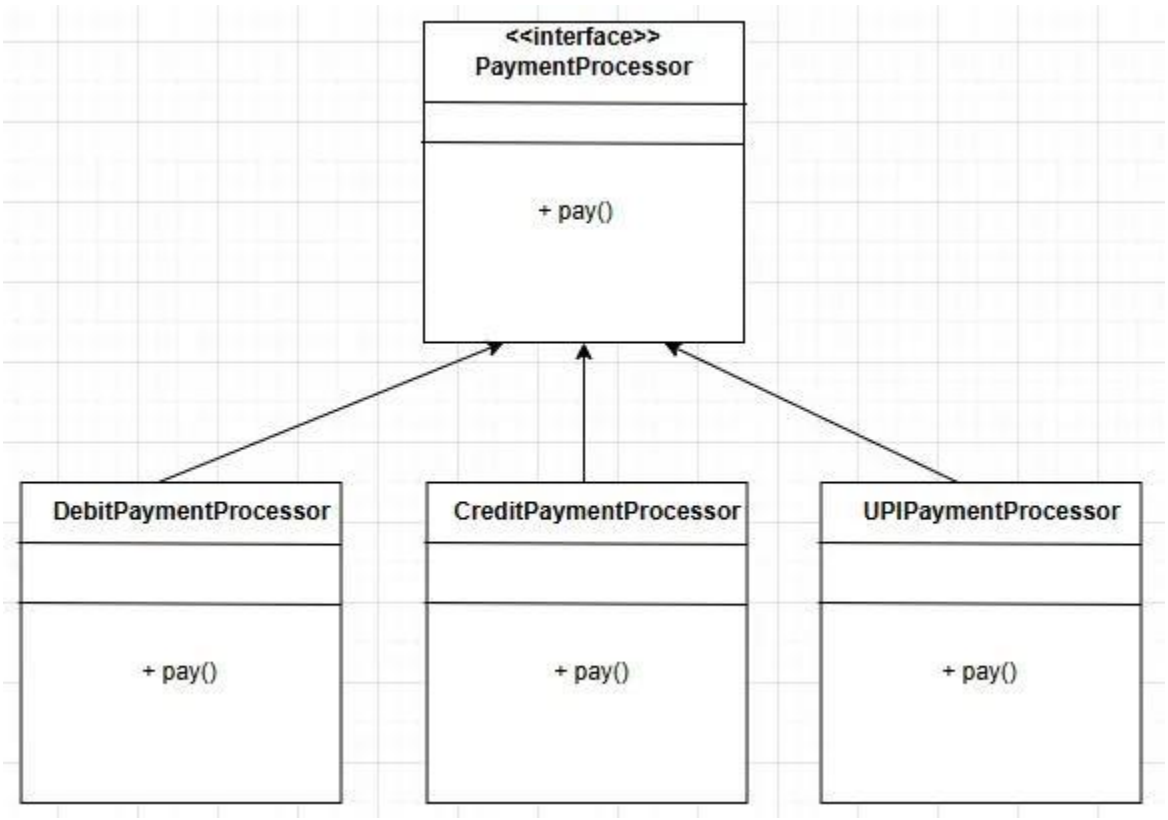

User
Credit Card Processing System
Validated Application
Credit Card Records
Credit and Payment Reports
Credit Card Request
Application Form
Information Input
Validating Information
Validated Information
Releasing Card
Credit Card Details
Invalid Information
Please provide exact information
Credit Notifications
Saving Information
Credit and Payment Information


2028 Payment Processing
UML Activity Diagram
Cash
Receive Cash
Print Payment Receipt
Return Payment Receipt
Complete Transaction
Create Transaction
Fund Transfer
Provide Bank Deposit Details
Confirm Payment Approval from Bank
Close Transaction
Credit
Request Credit Account Details
Request Payment from Bank
Cancel Transaction

3. CLASS DIAGRAM

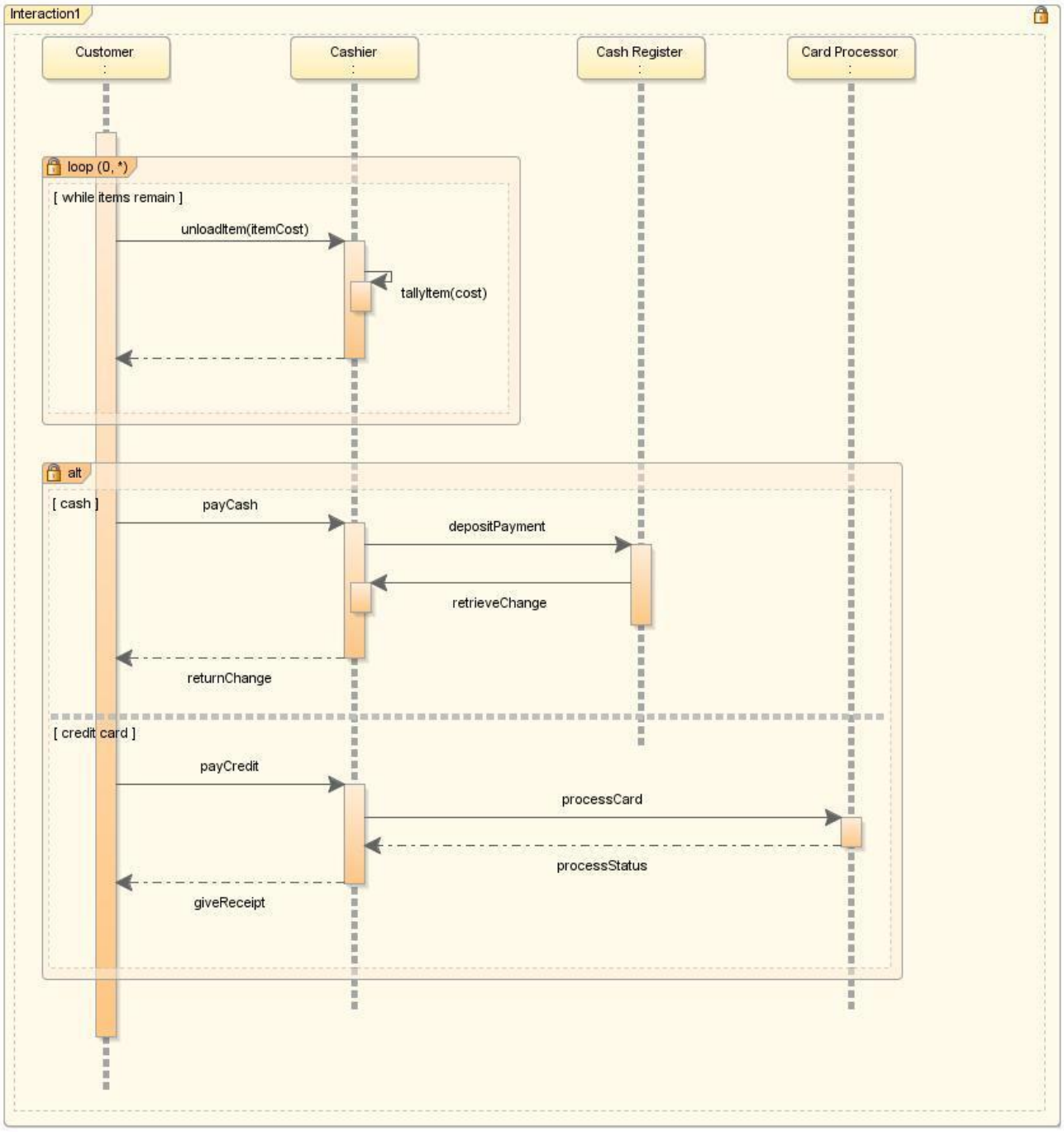


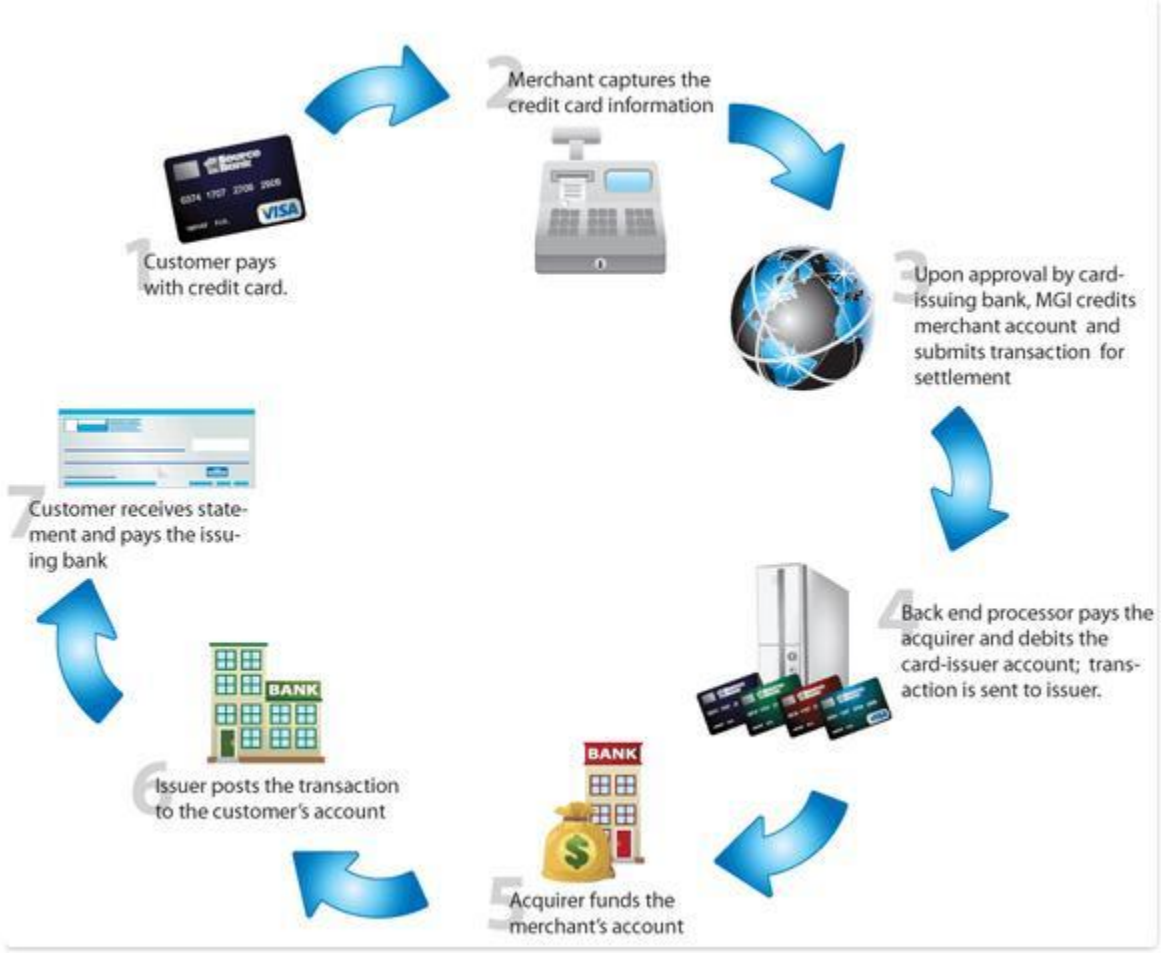

MailingAddress
address:MailingAddress
phone:PhoneNumber
CreditCardAccount
maximumCredit:Currency
currentBalance:Currency
Institution
name
address:MailingAddress
phone:PhoneNumber
0..1
0..*
0..*
0..*
0..1
Customer
name
Statement
paymentDueDate:Date
financeCharge:Currency
minimumPayment:Currency
0..1
Transaction
transactionDate:Date
explanation
amount:Currency
CashAdvance
Interest
Purchase
Fee
feeType:Fee=NORMAL
Adjustment
0..*
Merchant
name
Date:
. Day
. Month
. Year
Fee: {NORMAL,LATE,PENALTY}
Currency:
. Amount
. Nationality
Address:
. Addr 1
. Addr 2
. City State Zip Country

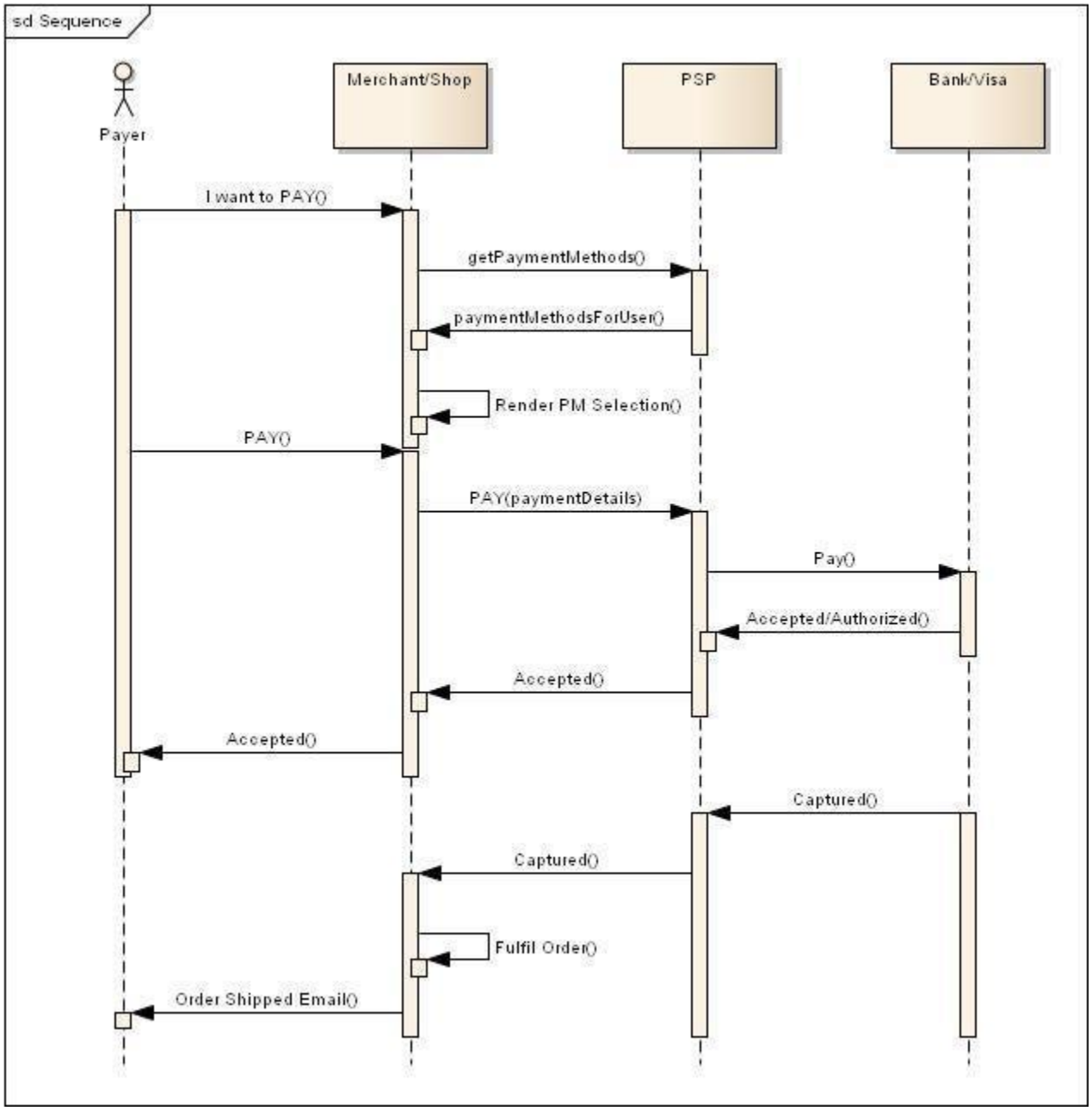


---

4. SEQUENCE DIAGRAM


Interaction1
Customer
Cashier
Cash Register
Card Processor
loop (0, *)
[while items remain]
unloadItem(itemCost)
tallyItem(cost)
alt
[cash]
payCash
depositPayment
retrieveChange
returnChange
[credit card]
payCredit
processCard
processStatus
giveReceipt


1 Customer pays with credit card.
2 Merchant captures the credit card information
3 Upon approval by card-issuing bank, MGI credits merchant account and submits transaction for settlement
4 Back end processor pays the acquirer and debits the card-issuer account; transaction is sent to issuer.
5 Acquirer funds the merchant's account
6 Issuer posts the transaction to the customer's account
7 Customer receives statement and pays the issuing bank
BANK
BANK



CONCLUSION

Ensures secure, fast, and reliable payment processing.

EXPERIMENT–9: E-BOOK MANAGEMENT SYSTEM

AIM

Develop test cases for unit testing and integration testing

(I) PROBLEM STATEMENT

The E-Book Management System allows users to:

- Store and manage digital books
- Search and download e-books
- Maintain user access records

PROJECT 1: E-BOOK MANAGEMENT SYSTEM

◈ 1. UNIT TEST CASES

Test Case ID	Module	Input	Expected Output
TC_U1	Login	Valid username/password	Login success
TC_U2	Login	Invalid credentials	Error message
TC_U3	Search Book	Valid title	Book displayed
TC_U4	Search Book	Invalid title	No results
TC_U5	Upload Book	Valid file	Upload success
TC_U6	Upload Book	Invalid format	Upload failed
TC_U7	Download Book	Available book	Download starts
TC_U8	Delete Book	Valid ID	Book removed

◈ 2. INTEGRATION TEST CASES

Test Case ID	Modules	Scenario	Expected Output
TC_I1	Login + Search	Login → search book	Results shown

Test Case ID	Modules	Scenario	Expected Output
TC_I2	Search + Download	Select book → download	File downloaded
TC_I3	Upload + Database	Upload book	Stored successfully
TC_I4	Admin + Delete	Delete book	Removed from system
TC_I5	User + Notification	Download complete	Notification sent

---

FINAL CONCLUSION

- **Unit Testing** verifies individual modules like validation, login, search
- **Integration Testing** ensures modules work together correctly
- Both are essential for building a reliable system

EXPERIMENT–10: RECRUITMENT SYSTEM

AIM

Develop test cases for various white box and black box testing techniques.

A. WHITE BOX TESTING

1. Statement Coverage

- **Ensure every line of code is executed at least once**
  Test Case:
- **Login with valid credentials → executes all statements**

---

2. Branch Coverage

Condition	Input	Expected Output
Valid login	Correct username/password	Success
Invalid login	Wrong password	Error

---

3. Path Testing

Paths:

1. **Login → Success → Dashboard**
2. **Login → Failure → Retry**
3. **Login → Failure → Exit**

---

4. Loop Testing

- **Attendance calculation loop**
  Test Case:
- **30 days attendance → correct calculation**

◈ B. BLACK BOX TESTING

1. Equivalence Partitioning

Input Type	Test Data	Expected Result
Valid Employee ID	EMP001	Accepted
Invalid Employee ID	NULL	Rejected

---

2. Boundary Value Analysis

Field	Min	Max	Expected
Salary	0	Max limit	Valid
Experience	0	40 years	Valid

---

3. Decision Table Testing

Condition	Login	Output
Correct credentials	Yes	Access
Incorrect credentials	No	Denied

---

4. Use Case Testing

- **Add Employee**
- **Assign Project**
- **Generate Report**

☑ PROJECT 2: RECRUITMENT SYSTEM

◈ A. WHITE BOX TESTING

1. Statement Coverage

- **Execute all statements in job application module**

2. Branch Coverage

Condition	Input	Expected Output
Eligible candidate	Valid resume	Selected
Not eligible	Missing skills	Rejected

3. Path Testing

Paths:

1. **Apply Job → Screening → Selected**
2. **Apply Job → Screening → Rejected**
3. **Apply Job → Interview → Final Selection**

4. Loop Testing

- **Loop through multiple applicants**
  Test Case:
- **Process 100 applicants → all evaluated**

◈ B. BLACK BOX TESTING

1. Equivalence Partitioning

Input	Type	Expected
Valid resume	PDF/DOC	Accepted
Invalid file	TXT	Rejected

2. Boundary Value Analysis

Field	Min	Max	Expected
Experience	0	20+ years	Valid
Age	18	60	Valid

---

3. Decision Table Testing

Condition	Eligible	Output
Yes	Yes	Shortlisted
No	No	Rejected

---

4. Use Case Testing

- **Apply for job**
- **Upload resume**
- **Attend interview**
- **Get selection status**
