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 Section2(f) & 12(B)of the UGC act,1956

Department of Computer Science and Engineering(AIML)

# SOFTWARE ENGINEERING LAB

II B.TECH -I SEMESTER

**MLRS-BT25 REGULATION**

**A.Y: 2026-27**

# INDEX

# Department of Computer Science & Engineering (AI&ML)

## CERTIFICATE

This is to certify that this manual is a Bonafide record of practical work carried out **in SOFT WARE ENGINEERING LAB** for the **B.Tech Computer Science Engineering (AI&ML) III Semester** Programme during the academic year **2026–2027.**

This manual has been prepared by **Mr. CH.V KRISHNA MOHAN (Assistant Professor), & Mr.R.MALLIKARJUNA (Assistant Professor)** Department of Computer Science Engineering (AI&ML), with our own efforts and to the best of our knowledge.

**Signature of Lab Faculty** **Signature of HOD**

# PREFACE

This Lab Manual entitled "**Software Engineering Lab**" is intended for the use of II B. Tech I Semester Computer Science and Engineering (AIML) students of Marri Laxman Reddy Institute of Technology and Management, Dundigal, Hyderabad. The main objective of the Software Engineering Lab provides hands-on experience with intelligent computing techniques like neural networks, fuzzy logic, and genetic algorithms. Students learn to solve complex, real-world problems that lack exact mathematical models. It enhances skills in AI and machine learning through practical implementation and experimentation. The lab prepares learners for advanced research and industry applications in adaptive and intelligent systems.

By,

**Mr. CHV KRISHNA MOHAN &**
**Mr. R.MALLIKARJUNA**

# ACKNOWLEDGEMENT

It was really a good experience, working at Software Engineering Lab. First, I would like to thank Mr. CH.V KRISHNA MOHAN, Assistant Professor, & Mr. .MALLIKARJUNA Assistant Professor. Department of Computer Science & Engineering (AI&ML), Marri Laxman Reddy Institute of technology & Management for giving the technical support in preparing the document.

I express my sincere thanks to Dr. B Ravi Prasad, Head of the Department of Computer Science & Engineering (AI&ML), Marri Laxman Reddy Institute of technology & Management, for his concern towards me and gave me opportunity to prepare SOFT WARE ENGINEERING laboratory manual.

I am deeply indebted and gratefully acknowledge the constant support and valuable patronage of Dr. B Ravi Prasad, Dean Academics, Marri Laxman Reddy Institute of technology & Management. I am unboundedly grateful to him for timely corrections and scholarly guidance.

I express my heartful thanks to Dr. P. Sridhar, Director, and Dr. R. Murali Prasad, Principal, Marri Laxman Reddy Institute of technology & Management, for giving me this wonderful opportunity for preparing the SOFT WARE ENGINEERING laboratory manual.

At last, but not the least I would like to thank the entire Computer Science & Engineering Department faculties those who had inspired and helped me to achieve my goal.

By, **Mr. CHV KRISHNA MOHAN &**
**Mr. R.MALLIKARJUNA**
Department of Computer Science & Engineering (AI&ML)

# GENERAL INSTRUCTIONS

1. Students are instructed to come to Software Engineering laboratory on time. Late comers are not entertained in the lab.
2. Students should be punctual to the lab. If not, the conducted experiments will not be repeated.
3. Students are expected to come prepared at home with the experiments which are going to be performed.
4. Students are instructed to display their identity cards before entering into the lab.
5. Students are instructed not to bring mobile phones to the lab.
6. Any damage/loss of system parts like keyboard, mouse during the lab session, it is student's responsibility and penalty or fine will be collected from the student.
7. Students should update the records and lab observation books session wise. Before leaving the lab the student should get his/her lab observation book signed by the faculty.
8. Students should submit the lab records by the next lab to the concerned faculty members in the staff room for their correction and return.
9. Students should not move around the lab during the lab session.
10. If any emergency arises, the student should take the permission from faculty member concerned in written format.
11. The faculty members may suspend any student from the lab session on disciplinary grounds.
12. Never copy the output from other students. Write down your own outputs.

## Department of Computer Science & Engineering (AI&ML)

# SAFETY MEASURES

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

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 provided 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 from equipment.

3. Computer and Equipment Handling

- Handle all computer systems, keyboards, mice, and peripherals with care.
- Do not attempt to open or tamper with any hardware components.
- Use only the assigned computer system; do not switch systems without permission.

4. Software and Data Safety

- Use only authorized software installed by the lab administrator.
- Do not attempt to install, uninstall, or modify any software without approval.
- Save your work frequently and ensure backups of important files.

5. Cyber security and Network Usage

- Keep your login credentials confidential.
- Do not attempt to access restricted websites or server
- Avoid activities such as hacking, gaming, or the use of pirated content.

6. Emergency Preparedness

- Be familiar with the location of emergency exits, fire extinguishers, and first aid kits.
- In the event of a fire, electrical hazard, or anyemergency, remain calm and inform the lab instructor immediately.
- Follow the evacuation procedure as instructed.

7. Post-Lab Procedures

- Log out of your session and shut down the system properly after use.
- Leave your workstation clean and organized.
- Return any borrowed materials or equipment to their proper place.

8. Hygiene and Cleanliness

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

# Department of Computer Science & Engineering (AI&ML)

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.

# VISION & MISSION OF THE DEPARTMENT

## Department Vision:

To nurture globally competent professionals in Artificial Intelligence and Machine Learning through excellence in education, research, and innovation, committed to developing sustainable and impactful solutions for the betterment of society.

## Department Mission:

- To provide a transformative learning environment that equips students with in-depth knowledge and practical skills in Artificial Intelligence and Machine Learning, fostering innovation, leadership, and lifelong learning.
- To advance AI and ML through cutting-edge research, strong industry collaboration, and community engagement, preparing students to address real-world challenges on a global scale.
- To produce competent and ethical AI professionals who contribute to technological progress while addressing societal and environmental challenges with sustainable solutions.
- To foster a research-driven culture by partnering with industry and academia, encouraging entrepreneurship, and engaging in community-centered technology development.

## Program Educational Objectives (PEOs)

**PEO:**

**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.

**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.

**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.

**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.

## Course Outcomes (COs)

- 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.

## Program Outcomes (POs)

**PO1.Engineering Knowledge:**

Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

**PO 2. 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.

**PO 3.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.

**PO 4. 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.

**PO 5. 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.

**PO 6. 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.

**PO 7. 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.

**PO 8.Ethics:**

Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

**PO 9. Individual and Team Work:**

Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

**PO 10. 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.

**PO 11. 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.

**PO 12. 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.

# PROGRAM SPECIFIC OUTCOMES (PSOs)

PSO1: Able to identify, analyze and solve the problems related to Artificial Intelligence and Machine Learning by applying the fundamental knowledge of Computer Science and Engineering.

1. Problem Identification and Formulation
2. Application of Computer Science Fundamentals
3. AI/ML Techniques and Tools Proficiency
4. Analytical and Critical Thinking

PSO2: Build innovative tools and techniques to develop project models in the areas related to Deep Learning, Machine learning, Artificial Intelligence.

1. Innovation and Tool Development
2. Implement end-to-end project models using real-world datasets in domains like image processing, NLP, or predictive analytics.
3. Advanced Technical Proficiency
4. Evaluation and Optimization.

PSO 3: Make use of the Artificial Intelligence and Machine Learning knowledge to assess societal, environmental, health, safety issues, Sustainable development goals with professional ethics and can also pursue higher studies, involve in research activities, be employable or entrepreneur.

1. Application of AI/ML for Societal and Environmental Impact
2. Ethical and Responsible AI Practice
3. Lifelong Learning and Research Orientation
4. Employability and Entrepreneurship

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING(AIML)**

**B.TECH VI SEMESTER CSM**
**TIME TABLE**

## ACADEMIC YEAR: 2026-2027

| Period | 1 | 2 | 3 | | 4 | 5 | 6 |
|---|---|---|---|---|---|---|---|
| Time | 09:40 to 10:35 | 10:35 to 11:30 | 11:30 to 12:25 | 12:25 to 1:15 | 1:15 to 02:10 | 02:10 to 03:05 | 03:05 to 04:00 |
| MON | | | | LUNCH BREAK | | | |
| TUE | | | | | | | |
| WED | | | | | | | |
| THU | | | | | | | |
| FRI | | | | | | | |
| SAT | | | | | | | |

## DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

# 25X0579: SOFTWARE ENGINEERING LAB

**B.Tech. II Year I Sem.** **L/T/P/C**

**0 0 2 1**

**Course Objectives:**

- To have hands-on experience in developing a software project by using various software Engineering principles and methods in each of the 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:**

Do the following seven exercises for any two projects given in the list of sample projects or any other Projects:

1. Development of problem statements.
2. Preparation of Software Requirement Specification Document, Design Documents and Testing Phase related documents.
3. Preparation of Software Configuration Management and Risk Management related documents.
4. Study and usage of any Design phase CASE tool
5. Performing the Design by using any Design phase CASE tools.
6. Develop test cases for unit testing and integration testing
7. Develop test cases for various 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 7. Software Personnel Management System 8. Credit Card Processing 9. E-book management System. 10. Recruitment system

**TEXT BOOKS**: 1. Software Engineering, A practitioner's Approach- Roger S. Pressman, 6th edition, McGraw Hill International Edition. 2. Software Engineering- Sommerville, 7th edition, Pearson Education. 3. The unified modeling language user guide Grady Booch, James Rambaugh, Ivar Jacobson, Pearson Education.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

# Virtual lab details

Name of the Virtual Lab: Software engineering

Virtual Lab Host Institute: IIT KHARAGPUR

URL to Lab: https://se-URL -itkgp.vlabs.ac.in

Academic Year: 2026-2027

Semester: III

## List of Experiments Available in Virtual Lab

1. Identifying the Requirements from Problem Statements
2. Estimation of Project Metrics
3. Modelling UML Use Case Diagrams and Capturing Use Case Scenarios
4. E-R Modelling from the Problem Statements

# CSM-A

## Lab Planner

| S.No | Experiment | CO | Virtual Lab Availability | Date planned | Date Conducted |
|---|---|---|---|---|---|
| 1 | Development of problem statements. | CO1 | NA | | |
| 2 | Preparation of Software Requirement Specification Document. | CO1 | NA | | |
| 3 | Design Documents and Testing Phase related documents. | CO2 | NA | | |
| 4 | Preparation of Software Configuration Management and Risk Management related documents. | CO2 | NA | | |
| 5 | Study and usage of any Design phase CASE tool | CO3 | NA | | |
| MID-I | | | | | |
| 6 | Performing the Design by using any Design phase CASE tools | CO3 | NA | | |
| 7 | Develop test cases for unit testing and integration testing | CO4 | NA | | |
| 8 | Develop test cases for various white box and black box testing techniques. | CO4 | NA | | |
| | | | | | |

## DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SOFTWARE ENGINEERING LABORATORY

# LAB PLANNER

| Date planed | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Date conducted | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Roll Number | 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* | Exp No | CO | VL* | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Note:VL*-Virtual Lab Availabilty**

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

## SOFTWARE ENGINEERING LABORATORY

## RUBRICS USED TO ASSESS LEARNINGS IN LABORATORIES

### 1. RUBRICS FOR DAY TO DAY EVALUATION

| Parameter | Max Mark | Level-1 (Very Poor) | Level- 2 (Poor) | Level-3 (Aver age) | Level-4 (Goo d) | Level-5 (Excellent) |
|---|---|---|---|---|---|---|
| **Observation Book** | 05 | No observations or irrelevant data. (0–1) | Incomplete or incorrect data. (2) | Basic values with some errors. (3) | Mostly correct with good format. (4) | Fully correct, clear, and well-formatted. (5) |
| **Record Writing** | 05 | Not submitted. (0-1) | Submitted but mostly incomplete. (2) | Submitted with some missing/wr ongparts. (3) | Submittedwith minor issues. (4) | Fully complete, correct algorithm & flowchart. (5) |
| **Result** | 05 | No result or major errors.(0–1) | Result partially obtained. (2) | Acceptable result with limited error. (3) | Near- correct result and reasonable error. (4) | Accurate result. (5) |
| **Viva-Voce** | 05 | Didnot answer any questions. (1) | Answered veryfew questions. (2) | Answered some questionswith help. (3) | Answered most question s correctly. (4) | Answered all questions accurately. (5) |

# 2. RUBRICS FOR INTERNAL EVALUATION

| Criterion | Max Marks | Level-1 *(Very Poor)* | Level-2 *(Poor)* | Level-3 *(Average)* | Level-4 *(Good)* | Level-5 *(Excellent)* |
|---|---|---|---|---|---|---|
| **Design/Tool/Apparatus Selection** | 2Marks | Incorrect tool/design and no reasoning. **(0)** | Tool/design Selection attempted with unclear logic. **(0.5)** | Satisfactory Selection with partial justification. **(1)** | Correct selection and proper analysis with few errors. **(1.5)** | Smart selection with accurate, relevant analysis. **(2)** |
| **Execution (Code/Debug/Run) /Analysis/Method Used** | 4 Marks | Didnot attempt or complete ly failed to execute. **(0)** | Attempted but unable to proceed or with major errors. **(1)** | Partial execution with some logic/syntax errors. **(2)** | Mostly correct execution with minimal help. **(3)** | Fully correct and Independentl y executed program. **(4)** |
| **Results& Documentation** | 2 Marks | Incomplete or poorly presented. **(0)** | Basic structure but lacks clarity or formattin g. **(0.5)** | Complete but generic or with formattin g issues. **(1)** | Well- structure dand mostly clear. **(1.5)** | Well-organized, professional ,and engaging documentatio n. **(2)** |
| **Viva-Voce (Understanding of Concepts)** | 2 Marks | No understandi n g; could notanswer questions. **(0)** | Answere d a few with difficulty . **(0.5)** | Answere dhalf the question swith basic clarity. **(1)** | Good understanding with confident answers. **(1.5)** | Answered all questions with clarity and depth. **(2)** |

# 3. RUBRICS FOR SEMESTER END EXAMINATIONS

| Criterion | Max Marks | Level-1 (Very Poor *(0–2 marks)* | Level-2 (Poor)*(3–4 marks)* | Level-3 (Average)*(5–6 marks)* | Level-4 (Good)*(7–9 marks)* | Level-5 (Excellent)*(10–12 marks)* |
|---|---|---|---|---|---|---|
| **Preparedness for the Experiment** | 12 marks | No clarity or objective or procedure. Unable to explain basics. | Limited idea of the objective/procedure. Needed prompting. | Has basic understanding; minor gaps in concept or preparation. | Well-prepared, with clear understanding of steps and background. | Fully prepared with strong conceptual clarity and confident explanation. |
| Performance inthe Laboratory | 12 marks | Unable to perform experiment. Relied entirely on examiner's help. | 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 calculation errors. Graphs/plots inaccurate or poorly labeled. | Calculations partially correct. Graphs present but with some flaws. | Correct calculations and graphs with minor errors. | Accurate calculations and well-labeled graphs with proper interpretation. |
| **Results & Error Analysis** | 12 marks | No result or invalid result. No error analysis attempted. | Incorrect result with vague or no error discussion. | Acceptable result. Error analysis attempted but limited. | Correct result with sound error discussion. | Accurate result with detailed and relevant error analysis. |
| **Viva-Voce (Subject Knowledge)** | 12 marks | Unable to answer any questions. No conceptual understanding. | Answered few questions with poor logic. | Answered half of the questions with average understanding. | Answered most questions with clarity and confidence. | Answered all questions with depth, clarity,and reasoning. |

# EXPERIMENT-1

**AIM:**

Development of problem statements.

## 1.1 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.

## 1.2 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. The process is time-consuming and involves multiple levels of manual intervention.

## 1.3 Drawbacks of Existing System

- Time-consuming procedures
- Long queues at passport offices
- High chances of human error
- Lack of transparency in application status
- Difficult document management
- Limited accessibility for users

## 1.4 Proposed System

The proposed Passport Automation System is a web-based application that allows users to apply for passports online. It automates the entire workflow, including application submission, document upload, verification, approval, and status tracking.

## 1.5 Features of Proposed System

- Online registration and login
- Digital application form submission
- Document upload facility
- Real-time status tracking
- Admin verification and approval
- Notification system

## 1.6 Objectives

- To reduce manual work and paperwork
- To improve processing speed and efficiency
- To provide transparency in application tracking
- To ensure secure data management
- To enhance user convenience

### 1.7 Scope of the System

The system is intended for:

- Citizens applying for passports
- Government officials for verification
- Administrative authorities for approval

# EXPERIMENT-2

**AIM :**

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

**Introduction**

## Purpose

The purpose of this document is to specify the requirements of the Passport Automation System, which automates passport application, verification, and approval processes.

### Scope

The system provides an online platform for:

- Passport application submission
- Document upload
- Status tracking
- Verification and approval by authorities

**Overall Description**

**User Classes**

- **Applicant (User):** Applies for passport
- **Admin:** Manages system and approvals
- **Verification Officer:** Verifies documents

**Operating Environment**

- Web-based application
- Runs on browsers (Chrome, Edge)
- Backend server with database

**Functional Requirements**

**User Module**

Register and login
Fill passport application form
Upload documents
Track application status

**Admin Module**

View applications
Verify documents
Approve/Reject applications
**Generate reports**
Verification Module
Validate submitted documents
Update verification status

**Non-Functional Requirements**

Performance
Fast response time (< 3 seconds)
Security

Secure login authentication
Data encryption
Reliability
System availability 24/7
Usability
User-friendly interface

### System Requirements

- OS: Windows/Linux
- Frontend: HTML, CSS, JavaScript
- Backend: Java/Python/PHP
- Database: MySQL

## 2. DESIGN DOCUMENTS

2.1 System Architecture
The system follows 3-Tier Architecture:
Presentation Layer – User Interface
Business Logic Layer – Processing logic
Database Layer – Data storage

2.2 UML DIAGRAMS (Description)
Use Case Diagram
Actors:
User → Apply, Track Status
Admin → Verify, Approve

Class Diagram
Classes:
User (UserID, Name, Email)
Application (AppID, Status)
Passport (PassportID)

Sequence Diagram
Flow:
User → Submit Application → System → Database → Admin → Approval

Activity Diagram
Start → Login → Fill Form → Upload Docs → Verification → Approval → End

2.3 Database Design
Tables
Users: UserID, Name, Email
Applications: AppID, UserID, Status

## 3. TESTING PHASE DOCUMENTS

3.1 Testing Objectives
Ensure system works correctly
Identify and fix errors
Validate requirements

3.2 Types of Testing
Unit Testing
Tests individual modules

| Test Case Input | Expected Output |
|---|---|

| Test Case | Input | Expected Output |
|---|---|---|
| Login | Valid credentials | Success |
| Login | Invalid | Error |

Integration Testing
Tests interaction between modules

| Modules | Expected Result |
|---|---|
| Login + Dashboard | Works correctly |
| Application + Database | Data stored |

System Testing
Tests complete system functionality
End-to-end application process
Document upload validation

Acceptance Testing
Conducted by users/admin
Confirms system meets requirements

3.3 White Box Testing
Basis Path Testing
Loop Testing
Condition Testing

3.4 Black Box Testing
Equivalence Partitioning
Valid: Age ≥ 18
Invalid: Age < 18
Boundary Value Analysis
17 → Invalid
18 → Valid
60 → Valid
61 → Invalid
Decision Table

| Condition | Output |
|---|---|
| Valid Data | Accepted |
| Invalid Data | Rejected |

3.5 Test Case Template

| Test Case ID | Description | Input | Expected Output | Result |
|---|---|---|---|---|
| TC01 | Login | Valid | Success | Pass |

# EXPERIMENT-3

## 3. AIM :

Preparation of Software Configuration Management and Risk Management related documents.

1.1 Introduction
Software Configuration Management (SCM) is the process of managing and controlling changes in software development. It ensures that all components of the Passport Automation System are systematically maintained and updated.

1.2 Objectives of SCM
To manage changes in the system effectively
To maintain version control of software artifacts
To ensure consistency and integrity of the system
To track modifications and updates

1.3 SCM Plan
1.3.1 Configuration Identification
The following items are identified as configuration items:
Source Code (Frontend & Backend)
Database Scripts
SRS Document
Design Documents (UML, DFD)
Test Cases
User Manuals

1.3.2 Version Control
Tool Used: Git
Repository: GitHub
Version naming example:
v1.0 – Initial version
v1.1 – Bug fixes
v2.0 – Feature updates

1.3.3 Change Management Process
Steps involved:
Change Request Submission
Impact Analysis
Approval by Project Manager
Implementation
Testing
Release

1.3.4 Configuration Status Accounting
Maintain records of:
Version history
Changes made
Current system status

1.3.5 Configuration Audits
Ensure all changes are properly implemented
Verify consistency between documents and code
Conduct periodic reviews

2. **RISK MANAGEMENT**
2.1 Introduction
Risk Management involves identifying, analyzing, and mitigating risks that may affect the Passport Automation System.

2.2 Objectives
Identify potential risks early
Reduce impact of risks
Ensure smooth project execution

2.3 Risk Identification

| Risk ID | Risk Description |
|---|---|
| R1 | Data security breach |
| R2 | System failure |
| R3 | Requirement changes |
| R4 | Network issues |
| R5 | Data loss |

2.4 Risk Analysis

| Risk | Probability | Impact |
|---|---|---|
| Data breach | Medium | High |
| System failure | Low | High |
| Requirement change | High | Medium |

2.5 Risk Mitigation Strategies

| Risk | Mitigation |
|---|---|
| Data breach | Use encryption, secure login |
| System failure | Backup systems |
| Requirement changes | Proper requirement analysis |
| Network issues | Use reliable servers |
| Data loss | Regular backups |

2.6 Risk Monitoring
Regular system checks
Continuous testing
Performance monitoring

2.7 Risk Management Plan
Identify risks
Analyze risks
Implement mitigation
Monitor continuously

# EXPERIMENT-4

**AIM** - Study and usage of any Design phase CASE tool

2. Introduction
CASE tools are software applications that support software development activities such as analysis, design, implementation, and testing. In the design phase, CASE tools are mainly used to create **UML diagrams**, which help visualize system structure and behavior.

3. Selected CASE Tool
Tool Name: Draw.io (diagrams.net) / StarUML

4. Features of CASE Tool
Easy-to-use graphical interface
Drag-and-drop design elements
Supports UML diagrams (Use Case, Class, Sequence, Activity)
Export diagrams as images or PDF
Cloud storage support

5. Purpose of Using CASE Tool
To design system architecture visually
To improve understanding of system flow
To reduce design complexity
To ensure clear communication among developers

6. Steps to Use CASE Tool (Draw.io Example)
Select Create New Diagram
Choose Blank Diagram or UML Template
Drag required shapes (actors, classes, processes)
Connect elements using arrows
Label all components clearly
Save or export the diagram

7. UML Diagrams Created for Passport Automation System
7.1 Use Case Diagram
Actors:
User (Applicant)
Admin
Functions:
Apply for Passport
Upload Documents
Track Status
Verify & Approve

7.2 Class Diagram
Classes:
User (UserID, Name, Email)
Application (AppID, Status)
Passport (PassportID)
Admin

7.3 Sequence Diagram
Flow:
User → Login → Submit Application → System → Database → Admin → Approval
7.4 Activity Diagram

Start → Login → Fill Form → Upload Documents → Verification → Approval → End

8. Advantages of CASE Tools
Improves design accuracy
Saves time and effort
Enhances documentation quality
Provides standard design formats

9. Limitations
Requires basic training, Some tools are paid, Limited customization in free versions

# EXPERIMENT-5

**AIM** : Performing the Design by using any Design phase CASE tools.

To design the Passport Automation System using a CASE tool by creating UML diagrams that represent system structure and behavior.

2. Objective
To understand system design using CASE tools
To create UML diagrams for the system
To visualize system workflow and architecture

3. Tool Used
**CASE Tool:** Draw.io (diagrams.net) / StarUML

4. Procedure
Open the CASE tool (Draw.io / StarUML)
Create a new project/diagram
Select UML diagram type
Add components (actors, classes, processes)
Connect components with arrows
Label all elements clearly
Save and export diagrams

5. System Design Using CASE Tool
5.1 Use Case Diagram
Actors:
Applicant (User)
Admin
Use Cases:
Register/Login
Apply for Passport
Upload Documents
Track Application Status
Verify Application (Admin)
Approve/Reject Application
*Explanation:*
The use case diagram shows interactions between users and the system.

5.2 Class Diagram
Classes & Attributes:
User
UserID
Name
Email
Application
ApplicationID
Status
Passport
PassportID
IssueDate
Admin
AdminID
Role
*Explanation:*
Represents structure of system and relationships between classes.

5.3 Sequence Diagram
Flow:
User logs in
User submits application
System validates data
Data stored in database
Admin verifies
System updates status
*Explanation:*
Shows step-by-step interaction between system components.

---

5.4 Activity Diagram
Flow:
Start → Login → Fill Application → Upload Documents → Verification → Approval → End
*Explanation:*
Represents workflow of passport application process.

5.5 System Architecture
3-Tier Architecture:
Presentation Layer**:** User Interface
Business Logic Layer**:** Processing
Database Layer**:** Data storage

6. Output
UML diagrams (Use Case, Class, Sequence, Activity) are successfully created using CASE tool.

7. Advantages
Easy visualization of system design
Improves communication
Reduces design errors

8. Result
The Passport Automation System is successfully designed using a CASE tool with appropriate UML diagrams representing system functionality and structure.

9. Conclusion
Using CASE tools for system design enhances clarity and efficiency. The Passport Automation System design helps in understanding system workflow and structure effectively.

# EXPERIMENT-6

**AIM :** Develop test cases for unit testing and integration testing

Unit testing focuses on individual modules like login, registration, application form, etc.

1.1 Login Module

| Test Case ID | Description | Input | Expected Output | Result |
|---|---|---|---|---|
| UT01 | Valid login | Correct username & password | Login success | Pass |
| UT02 | Invalid login | Wrong password | Error message | Pass |
| UT03 | Empty fields | No input | Validation error | Pass |

1.2 Registration Module

| Test Case ID | Description | Input | Expected Output |
|---|---|---|---|
| UT04 | Valid registration | All valid details | Registration success |
| UT05 | Duplicate email | Existing email | Error message |
| UT06 | Missing fields | Incomplete form | Validation error |

1.3 Application Form Module

| Test Case ID | Description | Input | Expected Output |
|---|---|---|---|
| UT07 | Valid form | Correct details | Form submitted |
| UT08 | Invalid age | Age < 18 | Error message |
| UT09 | Empty fields | Missing data | Validation error |

1.4 Document Upload Module

| Test Case ID | Description | Input | Expected Output |
|---|---|---|---|
| UT10 | Valid file upload | PDF/JPG file | Upload success |
| UT11 | Invalid file type | .exe file | Error message |
| UT12 | File size exceeded | Large file | Upload failed |

1.5 Admin Module

| Test Case ID | Description | Input | Expected Output |
|---|---|---|---|
| UT13 | Approve application | Valid data | Status = Approved |
| UT14 | Reject application | Invalid data | Status = Rejected |

2. INTEGRATION TESTING

Integration testing checks interaction between modules

2.1 Login + Dashboard

| Test Case ID | Modules | Description | Expected Output |
|---|---|---|---|
| IT01 | Login + Dashboard | Valid login | Dashboard displayed |
| IT02 | Login + Dashboard | Invalid login | Access denied |

2.2 Application Form + Database

| Test Case ID | Modules | Description | Expected Output |
|---|---|---|---|
| IT03 | Form + DB | Submit form | Data stored in DB |
| IT04 | Form + DB | Invalid data | Not stored |

2.3 Upload + Application

| Test Case ID | Modules | Description | Expected Output |
|---|---|---|---|
| IT05 | Upload + Form | Upload documents | Linked to application |
| IT06 | Upload + Form | Upload fails | Error message |

2.4 Admin + Application

| Test Case ID | Modules | Description | Expected Output |
|---|---|---|---|
| IT07 | Admin + Application | Approve request | Status updated |
| IT08 | Admin + Application | Reject request | Status updated |

2.5 End-to-End Flow

| Test Case ID | Modules | Description | Expected Output |
|---|---|---|---|
| IT09 | All modules | Complete process | Passport issued |
| IT10 | All modules | Invalid data | Process stopped |

# EXPERIMENT-7

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

White Box & Black Box Testing Techniques

1. WHITE BOX TESTING
Focus: Internal logic, code structure, paths, conditions

1.1 Basis Path Testing
Example: Login Module
Logic:
IF (username == valid AND password == valid)
  Login Success
ELSE
  Error
Test Cases

| Test Case ID | Path | Input | Expected Output |
|---|---|---|---|
| WB01 | Valid path | Correct credentials | Login success |
| WB02 | Invalid path | Wrong password | Error |
| WB03 | Invalid path | Wrong username | Error |

1.2 Condition Testing
Example: Age Validation (Passport Eligibility)
Condition:
IF (age >= 18 AND age <= 60)
  Valid
ELSE
  Invalid
Test Cases

| Test Case ID | Condition | Input | Expected Output |
|---|---|---|---|
| WB04 | True | Age = 25 | Accepted |
| WB05 | False | Age = 17 | Rejected |
| WB06 | False | Age = 61 | Rejected |

1.3 Loop Testing
Example: Document Upload Loop
Loop:
FOR each document upload
  Validate document
Test Cases

| Test Case ID | Scenario | Input | Expected Output |
|---|---|---|---|
| WB07 | Zero iteration | No documents | Error |
| WB08 | One iteration | 1 document | Success |
| WB09 | Multiple iteration | Multiple docs | All validated |

1.4 Data Flow Testing
Example: Application Form

| Test Case ID | Variable | Input | Expected Output |
|---|---|---|---|
| WB10 | Name field | Valid name | Stored correctly |
| WB11 | Name field | Empty | Error |

| Test Case ID | Variable | Input | Expected Output |
|---|---|---|---|
| WB12 | Email field | Invalid email | Validation error |

2. BLACK BOX TESTING
Focus: Input/output behavior (no internal logic)

2.1 Equivalence Partitioning
Example: Age Field

| Class | Input Range | Test Case | Expected Output |
|---|---|---|---|
| Valid | 18–60 | Age = 30 | Accepted |
| Invalid | <18 | Age = 16 | Rejected |
| Invalid | >60 | Age = 65 | Rejected |

2.2 Boundary Value Analysis (BVA)
Example: Age Limits

| Test Case ID | Input | Expected Output |
|---|---|---|
| BB01 | 17 | Invalid |
| BB02 | 18 | Valid |
| BB03 | 60 | Valid |
| BB04 | 61 | Invalid |

2.3 Decision Table Testing
Example: Login Validation

| Username | Password | Output |
|---|---|---|
| Valid | Valid | Success |
| Valid | Invalid | Error |
| Invalid | Valid | Error |
| Invalid | Invalid | Error |

2.4 Error Guessing
Possible Errors

| Test Case ID | Scenario | Expected Output |
|---|---|---|
| BB05 | Empty form submission | Error message |
| BB06 | Invalid file upload | Rejected |
| BB07 | Network failure | Retry message |

2.5 Use Case Testing

Example: Passport Application Flow

| Test Case ID | Scenario | Expected Output |
|---|---|---|
| BB08 | Complete valid process | Application success |
| BB09 | Missing documents | Error<br>Rejected |
| BB10 | Invalid data | |

**VIVA QUESTIONS:**

1. What is the aim of your Software Engineering lab?
2. What project did you develop in the lab?
3. What is the importance of Software Engineering in projects?
4. What are the phases involved in your project?
5. What is the difference between a project and a product?
6. What is a DFD?
7. What are the components of DFD?
8. What is Level 0 DFD?
9. What is Level 1 and Level 2 DFD?
10. Difference between DFD and flowchart?
11. What is a data store?
12. Name different UML diagrams used in your project.
13. What is a Use Case Diagram?
14. What is a Class Diagram?
15. What is a Sequence Diagram?
16. What is an Activity Diagram?
17. Difference between sequence and activity diagram?
18. What is a CASE tool?
19. Which CASE tool did you use?
20. What are the advantages of CASE tools?
21. Give examples of CASE tools.
22. Why are CASE tools used in design phase?
23. What is software design?
24. What is system architecture?
25. What is modularity?
26. What is cohesion and coupling?
27. What is 3-tier architecture?
28. What is software testing?
29. What are the types of testing?
30. What is unit testing?
31. What is integration testing?
32. What is system testing?

33. What is white box testing?
34. What is black box testing?
35. What is boundary value analysis?
36. What is equivalence partitioning?
37. What is basis path testing?
38. Explain your project (Passport Automation System).
39. What are the modules in your project?
40. What is the input and output of your system?
41. What database did you use?
42. What testing techniques did you apply?
43. What are the advantages of your system?
44. What are the limitations of your system?

`

# EXPERIMENT (Additional)

**AIM:** Study of SDLC Models

To study different Software Development Life Cycle (SDLC) models.
Models Covered:
Waterfall Model
Spiral Model
Agile Model
Incremental Model
Outcome:
Understanding of different development approaches and when to use them.

2. Requirement Elicitation Techniques
Aim:
To study techniques for gathering requirements.
Techniques:
Interviews
Questionnaires
Observation
Brainstorming
Outcome:
Ability to collect accurate system requirements.
3. Preparation of SRS Document
Aim:
To prepare a Software Requirement Specification.
Contents:
Introduction
Functional Requirements
Non-Functional Requirements
4. Data Flow Diagram (DFD)
Aim:
To draw DFD for the system.
Levels:
Level 0 (Context Diagram)
Level 1
Level 2

5. UML Diagram Design
Aim:
To design UML diagrams.
Diagrams:
Use Case Diagram
Class Diagram
Sequence Diagram
Activity Diagram

6. Study of CASE Tools
Aim:
To study CASE tools used in software design.

Examples:
Draw.io
StarUML
Rational Rose

7. Software Project Estimation
Aim:
To estimate software cost and effort.
Methods:
COCOMO Model
Function Point Analysis
Aim:
To identify and manage risks.
Activities:
Risk identification
Risk analysis
Risk mitigation

9. Test Case Design
Aim:
To design test cases for software.
Types:
Unit Testing
Integration Testing

10. White Box Testing Techniques
Aim:
To study internal testing methods.
Techniques:
Basis Path Testing
Loop Testing
Condition Testing

11. Black Box Testing Techniques
Aim:
To test system functionality.
Techniques:
Equivalence Partitioning
Boundary Value Analysis
Decision Table

12. Software Configuration Management
Aim:
To study version control and change management.
Tools:
Git
GitHub

13. Debugging Techniques
Aim:
To identify and fix errors.
Methods:
Logging
Breakpoints
Step execution