Diploma Engineering

Laboratory Manual

(Fundamentals of Software Development) (4331604)

Information Technology 3rd Semester

Enrolment No	
Name	
Branch	
Academic Term	
Institute	

Directorate Of Technical Education

Gandhinagar - Gujarat

DTE's Vision:

- To provide globally competitive technical education;
- Remove geographical imbalances and inconsistencies;
- Develop student friendly resources with a special focus on girls' education and support to weaker sections;
- Develop programs relevant to industry and create a vibrant pool of technical professionals.

DTE's Mission:

Institute's Vision:

Institute's Mission:

Department's Vision:

Department's Mission:

Certificate

This is to certify that Mr./Ms …….. Enrollment No. ……. of ……. Semester of *Diploma in*……..of ……. (GTU Code) has satisfactorily completed the term work in course **Fundamentals of Software Development (4331604**) for the academic year: …… Term: Odd/Even prescribed in the GTU curriculum.

Place:……..

Date: ……..

Signature of Course Faculty **Head of the Department**

Preface

The primary aim of any laboratory/Practical/field work is enhancement of required skills as well as creative ability amongst students to solve real time problems by developing relevant competencies in psychomotor domain. Keeping in view, GTU has designed competency focused outcome-based curriculum -2021 (COGC-2021) for Diploma engineering programmes. In this more time is allotted to practical work than theory. It shows importance of enhancement of skills amongst students and it pays attention to utilize every second of time allotted for practical amongst Students, Instructors and Lecturers to achieve relevant outcomes by performing rather than writing practice in study type. It is essential for effective implementation of competency focused outcome- based Green curriculum-2021. Every practical has been keenly designed to serve as a tool to develop & enhance relevant industry needed competency in each and every student. These psychomotor skills are very difficult to develop through traditional chalk and board content delivery method in the classroom. Accordingly, this lab manual has been designed to focus on the industry defined relevant outcomes, rather than old practice of conducting practical to prove concept and theory.

By using this lab manual, students can read procedure one day in advance to actual performance day of practical experiment which generates interest and also, they can have idea of judgement of magnitude prior to performance. This in turn enhances predetermined outcomes amongst students. Each and every Experiment /Practical in this manual begins by competency, industry relevant skills, course outcomes as well as practical outcomes which serve as a key role for doing the practical. The students will also have a clear idea of safety and necessary precautions to be taken while performing experiment.

This manual also provides guidelines to lecturers to facilitate student-centered lab activities for each practical/experiment by arranging and managing necessary resources in order that the students follow the procedures with required safety and necessary precautions to achieve outcomes. It also gives an idea that how students will be assessed by providing Rubrics.

The software has changed every aspect of our life and made available everything of our fingertips. It has revolutionized every area of human life like education, health, defence and security, finance and business, travel, social life, politics, entertainment and so on. This course guides the students to analyse, design, implement and test the software product with proper documentation during diploma engineering.

Although we try our level best to design this lab manual, but always there are chances of improvement. We welcome any suggestions for improvement.

Programme Outcomes (POs):

1. **Basic and Discipline specific knowledge:** Apply knowledge of basic mathematics, science and engineering fundamentals and engineering specialization to solve the engineering problems.

2. **Problem analysis**: Identify and analyse well-defined engineering problems using codified standard methods.

3. **Design/ development of solutions:** Design solutions for engineering well-defined technical problems and assist with the design of systems components or processes to meet specified needs.

4. **Engineering Tools, Experimentation and Testing:** Apply modern engineering tools and appropriate technique to conduct standard tests and measurements.

5. **Engineering practices for society, sustainability and environment:** Apply appropriate technology in context of society, sustainability, environment and ethical practices.

6. **Project Management:** Use engineering management principles individually, as a team member or a leader to manage projects and effectively communicate about well-defined engineering activities.

7. **Life-long learning:** Ability to analyze individual needs and engage in updating in the context of technological changes in field of engineering.

Practical Outcome - Course Outcome matrix

Course Outcomes (COs):

a. *CO1- Explain software development activities.*

b. *CO2- Select appropriate software process model for software project development.*

c. *CO3- Prepare software requirement specification (SRS) document for a software project.*

d. *CO4- Organize software project development schedule.*

e. *CO5- Prepare a design of the software with user interface.*

f. *CO6- Apply testing on software product with proper test cases.*

Sr. No.	Practical Outcome/Title of experiment	CO 1	CO2	CO3	CO4	CO5	CO 6
1.	Select a software project and identify the process model with proper justification.		√				
2.	Collect the functional requirements for the project. (Questionnaires/ stakeholders' interview questions)			√			
3.	Analyze functional and non-functional requirement and prepare SRS document for the project.			√			
4.	Prepare GANTT chart for selected system.				√		
5.	Prepare PERT chart for selected system.				√		
6.	Design DFD (context, level-1/2) for function oriented design.					√	
7.	Design data dictionary of the selected system.					√	
8.	Prepare User-case diagrams of the selected system for object oriented design.					√	

9.	Prepare Activity diagrams of the selected system for object oriented design.					**√**	
10.	Design appropriate User Interface based on the type of project.					**√**	
11.	Prepare the test cases to test the functionalities of the modules.						**√**

Industry Relevant Skills

The following industry relevant skills are expected to be developed in the students by performance of experiments of this course.

1. Analytical and logical thinking.
2. Project management, Team-based working, communication skill
3. Creativity & Design ability
4. Effective Analysis and Organizing.

Guidelines to Course Faculty

1. Couse faculty should demonstrate experiment with all necessary implementation strategies described in curriculum.
2. Couse faculty should explain industrial relevance before starting of each experiment.
3. Course faculty should Involve & give opportunity to all students for hands on experience.
4. Course faculty should ensure mentioned skills are developed in the students by asking.
5. Utilise 2 hrs of lab hours effectively and ensure completion of write up with quiz also.
6. Encourage peer to peer learning by doing same experiment through fast learners.

Instructions for Students

1. Organize the work in the group and make record of all observations.
2. Students shall develop maintenance skill as expected by industries.
3. Student shall attempt to develop related hand-on skills and build confidence.
4. Student shall develop the habits of evolving more ideas, innovations, skills etc.
5. Student shall refer technical magazines and data books.
6. Student should develop habit to submit the practical on date and time.
7. Student should well prepare while submitting write-up of exercise.

Continuous Assessment Sheet

Enrolment No: **Name**

Name: **Term:**

Sr.no	Practical Outcome/Title of experiment	Page	Date	Marks (25)	Sign
1	Select a software project and identify the process model with proper justification.				
2	Collect the functional requirements for the project. (Questionnaires/ stakeholders' interview questions)				
3	Analyse functional and non-functional requirement and prepare SRS document for the project.				
4	Prepare GANTT chart for selected system.				
5	Prepare PERT chart for selected system.				
6	Design DFD (context, level-1/2) for function oriented design.				
7	Design data dictionary of the selected system.				
8	Prepare User-case diagrams of the selected system for object oriented design.				
9	Prepare Activity diagrams of the selected system for object oriented design.				
10	Design appropriate User Interface based on the type of project.				
11	Prepare the test cases to test the functionalities of the modules.				

Date: ……

Practical No:1	**Select a software project and identify the process model with proper justification.**

A. Objective: To select and analyse the relevant model for assigned project and to define activities and related tasks set for assigned project. The choice depends on various factors such as project requirements, team size, project complexity, time constraints, and customer involvement.

B. Expected Program Outcomes (POs)

Po1 - Basic Knowledge.
Po2 - Discipline Knowledge.
Po3 - Individual & Teamwork.
Po4 - Communication.
Po5 - Lifelong learning.

C. Expected Skills to be developed based on competency:

This practical is expected to develop the following skills for the industry identified competency: 'Select a software project and identify the process model with proper justification.'

1. Select suitable process model for software development.

D. Expected Course Outcomes (Cos)

CO-2 Select appropriate software process model for software project development.

E. Practical Outcome (PRo)

After completion of this practical student will be able to identify suitable approach to develop software.

F. Expected Affective domain Outcome (ADos)

1. Follow precautionary measures.
2. Demonstrate working as a leader or two members.
3. Follow ethical practices.

G. Prerequisite Theory:

❖ **Software process model:**

A software process model is an abstraction of the software development process. The models specify the stages and order of a process.

There are many different software processes, but all involve:

1. Specification – defining what the system should do.
2. Design and implementation – defining the organization of the system and implementing the system.

3. Validation – checking that it does what the customer wants.
4. Evolution – changing the system in response to changing customer needs.

The goal of a software process model is to provide guidance for controlling and coordinating the tasks to achieve the product and objectives as effectively as possible.

There are many kinds of process models for meeting different requirements. We refer to these as SDLC models (Software Development Life Cycle models). The most popular and important SDLC models are as follows:

- Waterfall model
- Incremental model
- Prototype model
- Spiral model
- RAD model
- Agile model
- Scrum model

Selection of appropriate life cycle model for a project:

Selection of proper lifecycle model to complete a project is the most important task. It can be selected by keeping the advantages and disadvantages of various models in mind. The different issues that are analysed before selecting a suitable life cycle model are given below:

- **Characteristics of the software to be developed:** The choice of the life cycle model largely depends on the type of the software that is being developed. For small services projects, the agile model is favored. User interface part of the project is mainly developed through prototyping model.
- **Characteristics of the development team:** Team member's skill level is an important factor to deciding the life cycle model to use. If the development team is experienced in developing similar software, then even an embedded software can be developed using the Iterative Waterfall model.
- **Risk associated with the project:** If the risks are few and can be anticipated at the start of the project, then prototyping model is useful. If the risks are difficult to determine at the beginning of the project but are likely to increase as the development proceeds, then the spiral model is the best model to use.
- **Characteristics of the customer**: If the customer is not quite familiar with computers, then the requirements are likely to change frequently as it would be difficult to form complete, consistent, and unambiguous requirements. Thus, a prototyping model may be necessary to reduce later change requests from the customers. Initially, the customer's confidence is high on the development team. During the lengthy development process, customer confidence normally drops off as no working software is yet visible. So, the evolutionary model is useful as the customer can experience a partially working software much earlier than whole complete software.

H. Experimental set up/ Program Logic-Flow chart:

N/A

I. Resources/Equipment Required

Sr. No.	Instrument/Equipment /Components/Trainer kit	Specification	Quantity
1	Laptop/Desktop with Internet Connection	Any processor above i3 and any operating system	1

J. Safety and necessary Precautions followed.

1. Keep all computers in the lab area clean and free from dust.
2. Ensure all computers are properly grounded and plugged into surge protectors.
3. Use antivirus software to protect computers from malicious software and viruses.
4. Provide clear instructions and demonstration on how to use the tool.
5. Establish a safe and comfortable environment for students to work in.
6. Refrain from using computers in the lab for activities that could be deemed as inappropriate or illegal.

K. Procedure to be followed/Source code/Lab Task:

a) Write down software project definition and identify the process model. Provide suitable reasons for selecting this model.

L. Observations and Calculations/Input-Output:

N/A

M. Practical Related Quiz/Questions:

1. Define: software process model.
2. Differentiate Agile Model Vs Traditional Models.
3. Which software process model is most suitable for projects with rapidly changing requirements? Justify answer.
4. Explain Scrum Model.

N. References / Suggestions

1. Textbook of software engineering by pressman.
2. https://www.tutorialride.com/software-testing/software-development-process-models.htm
3. https://www.educative.io/blog/software-process-model-types

O. Assessment-Rubrics

	Excellent (5)	Very Good (4)	Good (3)	Satisfactory (2)	Poor (1)
Engagement	Students proactively contributes to practical session by offering thoughts & asking questions more than once per session	Students proactively contributes to practical session by offering thoughts & asking questions only once per session	Students actively contributes to practical session but rarely ask question.	Students rarely contributes to practical session and rarely ask question.	Students never contributes to practical session and rarely ask question.
Understanding & Explanation	Fully understood the practical and able to explain it precisely	Fully understood the practical but not able to explain it precisely	Understood the practical and able to explain it little.	Barely understood the practical and not able to explain it.	Not able to understand correctly and not able to explain it.
Time	Completed work within the deadline	Completed work one week later than the deadline	Completed work two week later than the deadline	Completed work three week later than the deadline	Completed work during submission.
Documentation	Documentation is done precisely.	Documentation is done properly with little bit formatting mistakes.	Documentation is done properly with little bit formatting, grammatical and spelling mistakes.	Documentation is done with too much formatting, grammatical and spelling mistakes.	Documentation is not done properly.
Viva (Q&A)	Answered all the questions with proper justification.	Answered 60% of questions properly.	Answered 40% of questions properly.	Answered 20% of questions properly.	Not answered any of questions properly.

Sign with Date

Date: ……

Practical No:2	**Collect the functional requirements for the project. (Questionnaires/ stakeholders' interview questions)**

A. Objective: identify project functional requirement and understanding different aspects for user requirements.

B. Expected Program Outcomes (POs)

Po1 - Basic Knowledge.
Po2 - Discipline Knowledge.
Po3 - Individual & Teamwork.
Po4 - Communication.
Po5 - Lifelong learning.

C. Expected Skills to be developed based on competency:

This practical is expected to develop the following skills for the industry identified competency: 'Collect the functional requirements for the project. (Questionnaires/ stakeholders' interview questions)'.

1. Gather application specific requirement for your project.
2. You can use Microsoft World, Notepad etc. for writing problem statement.

D. Expected Course Outcomes (Cos)

CO-3: Prepare software requirement specification (SRS) document for a software project.

E. Practical Outcome (PRo)

After completion of this practical student will be able to prepare proper SRS document.

F. Expected Affective domain Outcome (ADos)

1. Work as a leader/ a team member
2. Follow the project title.
3. Collect functional requirement of the selected project.

G. Prerequisite Theory:

- **Requirement Gathering: -**
  - Requirement engineering produces a specification of what a system should do.
  - The intention of requirement engineering is to provide a clear definition of requirement of the systems.

- This phase is a very important phase because, if the customer requirements are not clearly understood, the ambiguity can get into the other phase of the development.
- To avoid such issues, requirement has to be elicited using the right elicitation techniques, to be analysed effectively, specified clearly and verified thoroughly. All activities are collectively termed as requirement development activities.
- Requirement Analysis, also known as Requirement Engineering, is the process of defining user expectations for a new software being built or modified.
- Requirements analysis encompasses those tasks that go into determining the needs or conditions to meet for a new or altered product or project, taking account of the possibly conflicting requirements of the various stakeholders, analyzing, documenting, validating and managing software or system requirements. Here are the objectives for performing requirement analysis in the early stage of a software project.
- Requirements elicitation (also known as Requirements Gathering or Capture) is the process of generating a list of requirements (functional, system, technical, etc.) from the various stakeholders (customers, users, vendors, IT staff, etc.) that will be used as the basis for the formal Requirements Definition.

❖ **Techniques Used for Requirement Gathering: -**

- **Interviews -** These are an invaluable tool at the beginning of the process for getting background information on the business problems and understanding a current-world perspective of what the system being proposed needs to do. You need to make sure that your interviews cover a diverse cross-section of different stakeholders, so that the requirements are not skewed towards one function or area.
- **Questionnaires -** One of the challenges with interviews is that you will only get the information that the person is consciously aware of. Sometimes there are latent requirements and features that are better obtained through questionnaires. By using carefully chosen, probing questions (based on the information captured in prior interviews), you can drill-down on specific areas that the stakeholders don't know are important, but can be critical to the eventual design of the system.
- **User Observation -** One of the best ways to determine the features of a system, that does not result in "paving the cowpath" (i.e. building a slightly improved version of the current state) is to observe users actually performing their daily tasks, and ideally recording the actions and activities that take place. By understanding the holistic context of how they perform the tasks, you can write requirements that will reinvent the processes rather than just automating them, and will ensure that usability is paramount.
- **Brainstorming -** This is a powerful activity, which can be performed either in the context of a work shop or on its own. By considering different parts of the system and considering 'what-if' scenarios, or 'blue-sky' ideas, you can break

out of the context of the current-state and consider visionary ideas for the future. Tools such as whiteboards or mind-mapping software can be very helpful in this phase.

- **Surveys-** Organization may conduct surveys among various stakeholders by querying about their expectation and requirements from the upcoming system.

❖ **How Should the Information Be Captured?**

- There are many ways to capture the information, from a simple Word document, spreadsheet, or presentation to sophisticated modelling diagrams. We recommend that the initial high-level brainstorming and requirements discovery be done on a whiteboard to foster collaboration.
- Once the initial ideas have crystallized, we recommend using a formal Requirements Management System to record the information from the whiteboard and drill-down the functional requirements in smaller focus-groups to arrive at the use-cases and system requirements.

H. Experimental set up/ Program Logic-Flow chart:

N/A

I. Resources/Equipment Required

Sr. No.	Instrument/Equipment /Components/Trainer kit	Specification	Quantity
1	Laptop/Desktop with Internet Connection	Any processor above i3 and any operating system	1

J. Safety and necessary Precautions followed.

1. Keep all computers in the lab area clean and free from dust.
2. Ensure all computers are properly grounded and plugged into surge protectors.
3. Use antivirus software to protect computers from malicious software and viruses.
4. Provide clear instructions and demonstration on how to use the tool.
5. Establish a safe and comfortable environment for students to work in.
6. Refrain from using computers in the lab for activities that could be deemed as inappropriate or illegal.

K. Procedure to be followed/Source code/Lab Task:

a) Collect the functional requirements for the selected project and write down techniques used for collecting data.

L. Observations and Calculations/Input-Output:

N/A

M. Practical Related Quiz/Questions:

1. List out methods used to gather functional requirements.
2. What techniques are being employed to prioritize functional requirements?
3. Define: Requirement Gathering

N. References / Suggestions

1. Software Engineering by press man
2. https://www.tutorialspoint.com/business_analysis/business_analysis_requirement_gathering_techniques.htm

O. Assessment-Rubrics

	Excellent (5)	Very Good (4)	Good (3)	Satisfactory (2)	Poor (1)
Engagement	Students proactively contributes to practical session by offering thoughts & asking questions more than once per session	Students proactively contributes to practical session by offering thoughts & asking questions only once per session	Students actively contributes to practical session but rarely ask question.	Students rarely contributes to practical session and rarely ask question.	Students never contributes to practical session and rarely ask question.
Understanding & Explanation	Fully understood the practical and able to explain it precisely	Fully understood the practical but not able to explain it precisely	Understood the practical and able to explain it little.	Barely understood the practical and not able to explain it.	Not able to understand correctly and not able to explain it.
Time	Completed work within the deadline	Completed work one week later than the deadline	Completed work two week later than the deadline	Completed work three week later than the deadline	Completed work during submission.
Documentation	Documentation is done precisely.	Documentation is done properly with little bit formatting mistakes.	Documentation is done properly with little bit formatting, grammatical and spelling mistakes.	Documentation is done with too much formatting, grammatical and spelling mistakes.	Documentation is not done properly.

Viva (Q&A)	Answered all the questions with proper justification.	Answered 60% of questions properly.	Answered 40% of questions properly.	Answered 20% of questions properly.	Not answered any of questions properly.

Sign with Date

Date: ……

Practical No:3	**Analyse functional and non-functional requirement and prepare SRS document for the project.**

A. Objective: To Classify the requirement into functional and non-function requirements and prepare SRS document for the selected project.

B. Expected Program Outcomes (POs)

Po1 - Basic Knowledge.
Po2 - Discipline Knowledge.
Po3 - Individual & Teamwork.
Po4 - Communication.
Po5 - Lifelong learning.

C. Expected Skills to be developed based on competency:

This practical is expected to develop the following skills for the industry identified competency: 'Analyse functional and non-functional requirement and prepare SRS document for the project'.

1. Prepare broad SRS (Software requirement Software) for the above selected project.

D. Expected Course Outcomes (Cos)

CO-3: Prepare software requirement specification (SRS) document for a software project.

E. Practical Outcome (PRo)

After completion of this practical student will be able to understand user requirement and prepare SRS.

F. Expected Affective domain Outcome (ADos)

1. Work as a leader/ a team member
2. Follow the project title.
3. Prepare SRS for the selected project.

G. Prerequisite Theory:

- **Requirement Gathering: -**
  - Requirement engineering produces a specification of what a system should do.
  - The intention of requirement engineering is to provide a clear definition of requirement of the systems.

- This phase is a very important phase because, if the customer requirements are not clearly understood, the ambiguity can get into the other phase of the development.
- To avoid such issues, requirement has to be elicited using the right elicitation techniques, to be analysed effectively, specified clearly and verified thoroughly. All activities are collectively termed as requirement development activities.
- Requirement Analysis, also known as Requirement Engineering, is the process of defining user expectations for a new software being built or modified.
- Requirements analysis encompasses those tasks that go into determining the needs or conditions to meet for a new or altered product or project, taking account of the possibly conflicting requirements of the various stakeholders, analyzing, documenting, validating and managing software or system requirements. Here are the objectives for performing requirement analysis in the early stage of a software project.
- Requirements elicitation (also known as Requirements Gathering or Capture) is the process of generating a list of requirements (functional, system, technical, etc.) from the various stakeholders (customers, users, vendors, IT staff, etc.) that will be used as the basis for the formal Requirements Definition.

❖ **Techniques Used for Requirement Gathering: -**

- **Interviews -** These are an invaluable tool at the beginning of the process for getting background information on the business problems and understanding a current-world perspective of what the system being proposed needs to do. You need to make sure that your interviews cover a diverse cross-section of different stakeholders, so that the requirements are not skewed towards one function or area.
- **Questionnaires -** One of the challenges with interviews is that you will only get the information that the person is consciously aware of. Sometimes there are latent requirements and features that are better obtained through questionnaires. By using carefully chosen, probing questions (based on the information captured in prior interviews), you can drill-down on specific areas that the stakeholders don't know are important, but can be critical to the eventual design of the system.
- **User Observation -** One of the best ways to determine the features of a system, that does not result in "paving the cowpath" (i.e. building a slightly improved version of the current state) is to observe users actually performing their daily tasks, and ideally recording the actions and activities that take place. By understanding the holistic context of how they perform the tasks, you can write requirements that will reinvent the processes rather than just automating them, and will ensure that usability is paramount.
- **Brainstorming -** This is a powerful activity, which can be performed either in the context of a work shop or on its own. By considering different parts of the system and considering 'what-if' scenarios, or 'blue-sky' ideas, you can break

out of the context of the current-state and consider visionary ideas for the future. Tools such as whiteboards or mind-mapping software can be very helpful in this phase.

- **Surveys-** Organization may conduct surveys among various stakeholders by querying about their expectation and requirements from the upcoming system.

❖ **How Should the Information Be Captured?**

- There are many ways to capture the information, from a simple Word document, spreadsheet, or presentation to sophisticated modelling diagrams. We recommend that the initial high-level brainstorming and requirements discovery be done on a whiteboard to foster collaboration.
- Once the initial ideas have crystallized, we recommend using a formal Requirements Management System to record the information from the whiteboard and drill-down the functional requirements in smaller focus-groups to arrive at the use-cases and system requirements.

H. Experimental set up/ Program Logic-Flow chart:

N/A

I. Resources/Equipment Required

Sr. No.	Instrument/Equipment /Components/Trainer kit	Specification	Quantity
1	Laptop/Desktop with Internet Connection	Any processor above i3 and any operating system	1

J. Safety and necessary Precautions followed.

1. Keep all computers in the lab area clean and free from dust.
2. Ensure all computers are properly grounded and plugged into surge protectors.
3. Use antivirus software to protect computers from malicious software and viruses.
4. Provide clear instructions and demonstration on how to use the tool.
5. Establish a safe and comfortable environment for students to work in.
6. Refrain from using computers in the lab for activities that could be deemed as inappropriate or illegal.

K. Procedure to be followed/Source code/Lab Task:

a) Collect the functional requirements for the selected project and write down techniques used for collecting data.

L. Observations and Calculations/Input-Output:

N/A

M. Practical Related Quiz/Questions:

1. List out methods used to gather functional requirements.
2. What techniques are being employed to prioritize functional requirements?
3. Define: Requirement Gathering

N. References / Suggestions

1. Software Engineering by press man
2. https://www.tutorialspoint.com/business_analysis/business_analysis_requirement_gathering_techniques.htm

O. Assessment-Rubrics

	Excellent (5)	Very Good (4)	Good (3)	Satisfactory (2)	Poor (1)
Engagement	Students proactively contributes to practical session by offering thoughts & asking questions more than once per session	Students proactively contributes to practical session by offering thoughts & asking questions only once per session	Students actively contributes to practical session but rarely ask question.	Students rarely contributes to practical session and rarely ask question.	Students never contributes to practical session and rarely ask question.
Understanding & Explanation	Fully understood the practical and able to explain it precisely	Fully understood the practical but not able to explain it precisely	Understood the practical and able to explain it little.	Barely understood the practical and not able to explain it.	Not able to understand correctly and not able to explain it.
Time	Completed work within the deadline	Completed work one week later than the deadline	Completed work two week later than the deadline	Completed work three week later than the deadline	Completed work during submission.
Documentation	Documentation is done precisely.	Documentation is done properly with little bit formatting mistakes.	Documentation is done properly with little bit formatting, grammatical and spelling mistakes.	Documentation is done with too much formatting, grammatical and spelling mistakes.	Documentation is not done properly.

Viva (Q&A)	Answered all the questions with proper justification.	Answered 60% of questions properly.	Answered 40% of questions properly.	Answered 20% of questions properly.	Not answered any of questions properly.

Sign with Date

Date: ……

Practical No:4	**Prepare GANTT chart for selected system.**

A. Objective: Apply planning, scheduling, and monitoring of any project.

B. Expected Program Outcomes (POs)

Po1 - Basic Knowledge.
Po2 - Discipline Knowledge.
Po3 - Individual & Teamwork.
Po4 - Communication.
Po5 - Lifelong learning.

C. Expected Skills to be developed based on competency:

This practical is expected to develop the following skills for the industry identified competency: 'Prepare software requirement specification (SRS) document for a software project.'

1. Apply software planning using Gantt chart.

D. Expected Course Outcomes (Cos)

CO4- Organize software project development schedule.

E. Practical Outcome (PRo)

After completion of this practical student will be able to create gantt chart for given project definition.

F. Expected Affective domain Outcome (ADos)

1. Demonstrate working as a leader or two members.
2. Follow ethical practices.

G. Prerequisite Theory:

❖ **key components and concepts of a Gantt chart:**

- **Tasks**: Each task in the project is listed as a separate row in the chart. Tasks are typically represented by horizontal bars or blocks on the chart, with each bar representing the duration of a task.

- **Time Scale**: The Gantt chart has a horizontal axis that represents time, usually divided into days, weeks, or months. The time scale helps in visualizing the project's timeline and the start and end dates of each task.

- **Duration**: The duration of each task is represented by the length of its corresponding bar on the chart. It indicates the time required to complete the task.

- **Dependencies**: Gantt charts also depict task dependencies, which are the relationships between tasks that determine their sequencing. Dependencies are shown through arrows or lines connecting the bars of dependent tasks. These arrows indicate which tasks must be completed before others can start.

- **Progress Tracking**: As the project progresses, the Gantt chart can be updated to reflect the actual start and end dates of tasks. This allows project managers to track the progress of each task, compare it with the planned schedule, and adjust if necessary.

Gantt charts provide a clear and visual representation of the project schedule, making it easier to understand the timing and dependencies of tasks. They help in effective project planning, communication, and coordination among team members and stakeholders. By using a Gantt chart, project managers can monitor progress, identify potential delays or bottlenecks, and make informed decisions to keep the project on track.

H. Experimental set up/ Program Logic-Flow chart:

N/A

I. Resources/Equipment Required

Sr. No.	Instrument/Equipment /Components/Trainer kit	Specification	Quantity
1	Laptop/Desktop with Internet Connection	Any processor above i3 and any operating system	1

J. Safety and necessary Precautions followed.

1. Keep all computers in the lab area clean and free from dust.
2. Ensure all computers are properly grounded and plugged into surge protectors.
3. Use antivirus software to protect computers from malicious software and viruses.
4. Provide clear instructions and demonstration on how to use the tool.
5. Establish a safe and comfortable environment for students to work in.
6. Refrain from using computers in the lab for activities that could be deemed as inappropriate or illegal.

K. Procedure to be followed/Source code/Lab Task:

a) Prepare Gantt chart for any one project definition.

L. Observations and Calculations/Input-Output:

N/A

M. Practical Related Quiz/Questions:

1. List out methods used to gather functional requirements. How does a Gantt chart help in visualizing a project's timeline?
2. What is the significance of task dependencies in a Gantt chart?
3. What are the advantages of using a Gantt chart in project planning and execution?

4. Gantt charts are used for_____

A. Forecasting sales

B. Production schedule

C. Scheduling and routing

D. Linear programming

5. Which of the following is a characteristic of a Gantt chart?

a. It is a normal graph.

b. Project activities are listed.

c. Shows progress to date for a particular activity.

d. All of the above.

N. References / Suggestions

1. https://www.canva.com/graphs/gantt-charts/

O. Assessment-Rubrics

	Excellent (5)	Very Good (4)	Good (3)	Satisfactory (2)	Poor (1)
Engagement	Students proactively contributes to practical session by offering thoughts & asking questions more than once per session	Students proactively contributes to practical session by offering thoughts & asking questions only once per session	Students actively contributes to practical session but rarely ask question.	Students rarely contributes to practical session and rarely ask question.	Students never contributes to practical session and rarely ask question.

Understanding & Explanation	Fully understood the practical and able to explain it precisely	Fully understood the practical but not able to explain it precisely	Understood the practical and able to explain it little.	Barely understood the practical and not able to explain it.	Not able to understand correctly and not able to explain it.
Time	Completed work within the deadline	Completed work one week later than the deadline	Completed work two week later than the deadline	Completed work three week later than the deadline	Completed work during submission.
Documentation	Documentation is done precisely.	Documentation is done properly with little bit formatting mistakes.	Documentation is done properly with little bit formatting, grammatical and spelling mistakes.	Documentation is done with too much formatting, grammatical and spelling mistakes.	Documentation is not done properly.
Viva (Q&A)	Answered all the questions with proper justification.	Answered 60% of questions properly.	Answered 40% of questions properly.	Answered 20% of questions properly.	Not answered any of questions properly.

Sign with Date

Date: ……

Practical No:5	**Prepare PERT chart for selected system.**

A. Objective: To organize and review different tasks of in project.

B. Expected Program Outcomes (POs)

Po1 - Basic Knowledge.
Po2 - Discipline Knowledge.
Po3 - Individual & Teamwork.
Po4 - Communication.
Po5 - Lifelong learning.

C. Expected Skills to be developed based on competency:

This practical is expected to develop the following skills for the industry identified competency: 'Prepare software requirement specification (SRS) document for a software project.'

1. Review software planning using PERT chart.

D. Expected Course Outcomes (Cos)

CO4- Organize software project development schedule.

E. Practical Outcome (PRo)

After completion of this practical student will be able to create scheduling and review different tasks of project.

F. Expected Affective domain Outcome (ADos)

3. Demonstrate working as a leader or two members.
4. Follow ethical practices.

G. Prerequisite Theory:

A PERT (Program Evaluation and Review Technique) chart, also known as a PERT diagram, is a project management tool used to schedule, organize, and coordinate tasks within a project. It provides a graphical representation of the project's timeline and helps project managers break down each task for analysis.

Here's an explanation of the key components and concepts of a PERT chart:

- **Tasks**: Each task in the project is represented by a node or box in the chart. Tasks are specific activities or work units that need to be completed to achieve the project's objectives.

- **Arrows**: Arrows are used to represent the dependencies between tasks. They indicate the flow and sequence of tasks, showing which tasks must be completed before others can start. Arrows connect the nodes, forming a network of tasks.

- **Estimation of Task Durations**: Each task in the PERT chart is associated with an estimated duration or time required to complete it. This estimation can be represented as optimistic, pessimistic, and most likely durations. These estimates help in calculating the expected duration of the project and assessing project risks.

- **Critical Path**: The critical path is the longest sequence of dependent tasks that determines the overall duration of the project. It represents the shortest possible time to complete the project. Any delay in tasks on the critical path will directly impact the project's completion date. Identifying the critical path helps project managers prioritize tasks and allocate resources effectively.

Overall, a PERT chart facilitates effective project planning, coordination, and monitoring by providing a visual representation of tasks and their relationships within the project.

H. Experimental set up/ Program Logic-Flow chart:

N/A

I. Resources/Equipment Required

Sr. No.	Instrument/Equipment /Components/Trainer kit	Specification	Quantity
1	Laptop/Desktop with Internet Connection	Any processor above i3 and any operating system	1

J. Safety and necessary Precautions followed.

7. Keep all computers in the lab area clean and free from dust.
8. Ensure all computers are properly grounded and plugged into surge protectors.
9. Use antivirus software to protect computers from malicious software and viruses.
10. Provide clear instructions and demonstration on how to use the tool.
11. Establish a safe and comfortable environment for students to work in.
12. Refrain from using computers in the lab for activities that could be deemed as inappropriate or illegal.

K. Procedure to be followed/Source code/Lab Task:

a) Prepare PERT chart for any one project.

L. Observations and Calculations/Input-Output:

N/A

M. Practical Related Quiz/Questions:

1. What is the purpose of a PERT chart in project management?
2. What are the main components of a PERT chart?
3. How are tasks represented in a PERT chart?

N. References / Suggestions

1. https://www.visme.co/pert-chart-generator/

O. Assessment-Rubrics

	Excellent (5)	Very Good (4)	Good (3)	Satisfactory (2)	Poor (1)
Engagement	Students proactively contributes to practical session by offering thoughts & asking questions more than once per session	Students proactively contributes to practical session by offering thoughts & asking questions only once per session	Students actively contributes to practical session but rarely ask question.	Students rarely contributes to practical session and rarely ask question.	Students never contributes to practical session and rarely ask question.
Understanding & Explanation	Fully understood the practical and able to explain it precisely	Fully understood the practical but not able to explain it precisely	Understood the practical and able to explain it little.	Barely understood the practical and not able to explain it.	Not able to understand correctly and not able to explain it.
Time	Completed work within the deadline	Completed work one week later than the deadline	Completed work two week later than the deadline	Completed work three week later than the deadline	Completed work during submission.
Documentation	Documentation is done precisely.	Documentation is done properly with little bit formatting mistakes.	Documentation is done properly with little bit formatting, grammatical and spelling mistakes.	Documentation is done with too much formatting, grammatical and spelling mistakes.	Documentation is not done properly.

Viva (Q&A)	Answered all the questions with proper justification.	Answered 60% of questions properly.	Answered 40% of questions properly.	Answered 20% of questions properly.	Not answered any of questions properly.

Sign with Date

Date: ……

Practical No:6	**Design DFD (context, level-1/2) for function-oriented design.**

A. Objective: Prepare DFD diagram up to level 2 and understand data flow in given project.

B. Expected Program Outcomes (POs)

Po1 - Basic Knowledge.
Po2 - Discipline Knowledge.
Po3 - Individual & Teamwork.
Po4 - Communication.
Po5 - Lifelong learning.

C. Expected Skills to be developed based on competency:

This practical is expected to develop the following skills for the industry identified competency: 'Prepare software requirement specification (SRS) document for a software project.'

1. Create different diagram related to data flow up to level 2 as per project definition.

D. Expected Course Outcomes (Cos)

CO5: Prepare a design of the software with user interface.

E. Practical Outcome (PRo)

After completion of this practical student will be able to create DFD Diagram level 0, 1 and 2.

F. Expected Affective domain Outcome (ADos)

1. Work as a leader/ a team member
2. Follow ethical practice.

G. Prerequisite Theory:

In a DFD, the system is represented as a set of interconnected components that exchange data.

- **Process**: A process represents a function or activity that transforms input data into output data. It can be a specific operation or a higher-level task within the system. Processes are typically depicted as circles or rectangles in a DFD.

- **Data Flow**: Data flows represent the movement of data between different components of the system. They show how data is input into the system,

processed, and output to various destinations. Data flows are represented by arrows connecting the processes, data stores, and external entities.

- **Data Store**: A data store is a repository where data is stored within the system. It can represent a physical database, a file, or any other form of data storage. Data stores are depicted as rectangles with two horizontal lines at the top and bottom.

- **External Entity**: An external entity is an external system, person, or organization that interacts with the system being analysed. It can be a user, another system, or an external data source or destination. External entities are represented as rectangles.

DFDs follow a top-down approach, meaning that they start with a high-level overview of the system and then progressively break down the processes into more detailed levels. This allows for a clear understanding of how data moves through the system and facilitates identification of potential bottlenecks or areas for improvement.

H. Experimental set up/ Program Logic-Flow chart:

N/A

I. Resources/Equipment Required

Sr. No.	Instrument/Equipment /Components/Trainer kit	Specification	Quantity
1	Laptop/Desktop with Internet Connection	Any processor above i3 and any operating system	1

J. Safety and necessary Precautions followed.

1. Keep all computers in the lab area clean and free from dust.
2. Ensure all computers are properly grounded and plugged into surge protectors.
3. Use antivirus software to protect computers from malicious software and viruses.
4. Provide clear instructions and demonstration on how to use the tool.
5. Establish a safe and comfortable environment for students to work in.
6. Refrain from using computers in the lab for activities that could be deemed as inappropriate or illegal.

K. Procedure to be followed/Source code/Lab Task:

a) Prepare DFD diagram level 0,1 and 2 for any project.

L. Observations and Calculations/Input-Output:

N/A

M. Practical Related Quiz/Questions:

1. Which of the following represents the movement of data between different components in a DFD diagram?
   a) Process
   b) Data flow
   c) Data store
   d) External entity

2. What is the purpose of a DFD diagram?
   a) To represent the physical layout of a database
   b) To depict the sequence of activities in a process
   c) To visualize the flow of data within a system
   d) To define the user interface of a software application

3. What does an external entity represent in a DFD diagram?
   a) A process within the system
   b) A repository where data is stored
   c) The movement of data between processes
   d) An external system, person, or organization interacting with the system

N. References / Suggestions

O. Assessment-Rubrics

	Excellent (5)	Very Good (4)	Good (3)	Satisfactory (2)	Poor (1)
Engagement	Students proactively contributes to practical session by offering thoughts & asking questions more than once per session	Students proactively contributes to practical session by offering thoughts & asking questions only once per session	Students actively contributes to practical session but rarely ask question.	Students rarely contributes to practical session and rarely ask question.	Students never contributes to practical session and rarely ask question.
Understanding & Explanation	Fully understood the practical and able to explain it precisely	Fully understood the practical but not able to explain it precisely	Understood the practical and able to explain it little.	Barely understood the practical and not able to explain it.	Not able to understand correctly and not able to explain it.

Time	Completed work within the deadline	Completed work one week later than the deadline	Completed work two week later than the deadline	Completed work three week later than the deadline	Completed work during submission.
Documentation	Documentation is done precisely.	Documentation is done properly with little bit formatting mistakes.	Documentation is done properly with little bit formatting, grammatical and spelling mistakes.	Documentation is done with too much formatting, grammatical and spelling mistakes.	Documentation is not done properly.
Viva (Q&A)	Answered all the questions with proper justification.	Answered 60% of questions properly.	Answered 40% of questions properly.	Answered 20% of questions properly.	Not answered any of questions properly.

Sign with Date

Date: ……

Practical No:7	**Design data dictionary of the selected system.**

A. **Objective:** To understand the overall database design, structure, relationships, and data flow.

B. Expected Program Outcomes (POs)

Po1 - Basic Knowledge.
Po2 - Discipline Knowledge.
Po3 - Individual & Teamwork.
Po4 - Communication.
Po5 - Lifelong learning.

C. Expected Skills to be developed based on competency:

This practical is expected to develop the following skills for the industry identified competency: 'Prepare software requirement specification (SRS) document for a software project.'

1. Create Data dictionary for above selected project.

D. Expected Course Outcomes (Cos)

CO5: Prepare a design of the software with user interface.

E. Practical Outcome (PRo)

Create Data Dictionary with detailed information for selected project.

F. Expected Affective domain Outcome (ADos)

1. Work as a leader/ a team member
2. Follow ethical practice.

G. Prerequisite Theory:

The main components of a data dictionary include the following:

- **Data Element**: A data element refers to a distinct piece of information within a system or database. It can be a field, attribute, column, or variable. The data dictionary provides detailed information about each data element, including its name, description, data type, size, format, constraints, and any associated business rules. This component helps in understanding the characteristics and properties of individual data elements.

- **Data Structure:** The data structure component of a data dictionary defines the organization and relationships of data elements within a system. It includes information about tables, entities, records, and other data structures used to store and organize data. This component specifies the primary keys, foreign keys, indexes, and relationships between different data elements. It helps in understanding the structure and integrity of the data.

- **Data Definitions:** Data definitions provide clear and concise explanations of the meaning, purpose, and usage of data elements. They ensure a common understanding of data across different stakeholders and help in eliminating ambiguity or misunderstandings. Data definitions typically include details such as synonyms, aliases, or alternate names used for the data elements.

- **Data Relationships and Dependencies:** This component describes the relationships and dependencies between data elements. It includes information about parent-child relationships, associations, or hierarchies among data elements. It helps in understanding how data elements are related and how changes in one data element may impact others.

- **Data Validation and Business Rules:** The data dictionary documents the validation rules, constraints, and business rules associated with data elements. It defines the allowable values, formats, ranges, and any other restrictions on data. This component ensures data integrity and helps in implementing data validation checks during data entry or data processing.

H. Experimental set up/ Program Logic-Flow chart:

N/A

I. Resources/Equipment Required

Sr. No.	Instrument/Equipment /Components/Trainer kit	Specification	Quantity
1	Laptop/Desktop with Internet Connection	Any processor above i3 and any operating system	1

J. Safety and necessary Precautions followed.

1. Keep all computers in the lab area clean and free from dust.
2. Ensure all computers are properly grounded and plugged into surge protectors.
3. Use antivirus software to protect computers from malicious software and viruses.

4. Provide clear instructions and demonstration on how to use the tool.
5. Establish a safe and comfortable environment for students to work in.
6. Refrain from using computers in the lab for activities that could be deemed as inappropriate or illegal.

K. Procedure to be followed/Source code/Lab Task:

a) Design data dictionary of the selected Project.

L. Observations and Calculations/Input-Output:

N/A

M. Practical Related Quiz/Questions:

1. Write Uses of Data Dictionary.
2. Explain each data dictionary components.

N. References / Suggestions

O. Assessment-Rubrics

	Excellent (5)	Very Good (4)	Good (3)	Satisfactory (2)	Poor (1)
Engagement	Students proactively contributes to practical session by offering thoughts & asking questions more than once per session	Students proactively contributes to practical session by offering thoughts & asking questions only once per session	Students actively contributes to practical session but rarely ask question.	Students rarely contributes to practical session and rarely ask question.	Students never contributes to practical session and rarely ask question.
Understanding & Explanation	Fully understood the practical and able to explain it precisely	Fully understood the practical but not able to explain it precisely	Understood the practical and able to explain it little.	Barely understood the practical and not able to explain it.	Not able to understand correctly and not able to explain it.
Time	Completed work within the deadline	Completed work one week later than the deadline	Completed work two week later than the deadline	Completed work three week later than the deadline	Completed work during submission.
Documentation	Documentation is done precisely.	Documentation is done properly with little bit formatting mistakes.	Documentation is done properly with little bit formatting, grammatical and spelling mistakes.	Documentation is done with too much formatting, grammatical and spelling mistakes.	Documentation is not done properly.

Viva (Q&A)	Answered all the questions with proper justification.	Answered 60% of questions properly.	Answered 40% of questions properly.	Answered 20% of questions properly.	Not answered any of questions properly.

Sign with Date

Date: ……

Practical No:8	**Prepare User-case diagrams of the selected system for object-oriented design.**

A. Objective: The objective of a use case diagram in UML is to demonstrate the different ways that a user might interact with a system.

B. Expected Program Outcomes (POs)

PO2: Problem analysis
PO3: Design/ development of solutions
PO4: Engineering Tools, Experimentation and Testing

C. Expected Skills to be developed based on competency:

This practical is expected to develop the following skills for the industry identified competency:

1. Identifying primary users and their use cases based on requirements.

D. Expected Course Outcomes (Cos)

CO5: Prepare a design of the software with user interface.

E. Practical Outcome (PRo)

Prepare usecases and draw usecase diagram for any System.

F. Expected Affective domain Outcome (ADos)

1. Work as a leader/ a team member
2. Follow ethical practice.

G. Prerequisite Theory:

- **What is Usecase Diagram?**

  In the Unified Modeling Language (UML), a use case diagram can summarize the details of your system's users (also known as actors) and their interactions with the system.

- **When to use usecase diagram?**

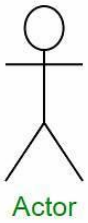
  use case diagram depicts a high-level overview of the relationship between use cases, actors, and systems. Experts recommend that use case diagrams be used to supplement a more descriptive textual use case.

  Use cases are represented with a labeled oval shape. Stick figures represent actors in the process, and the actor's participation in the system is modeled with a line between

the actor and use case. To depict the system boundary, draw a box around the use case itself.

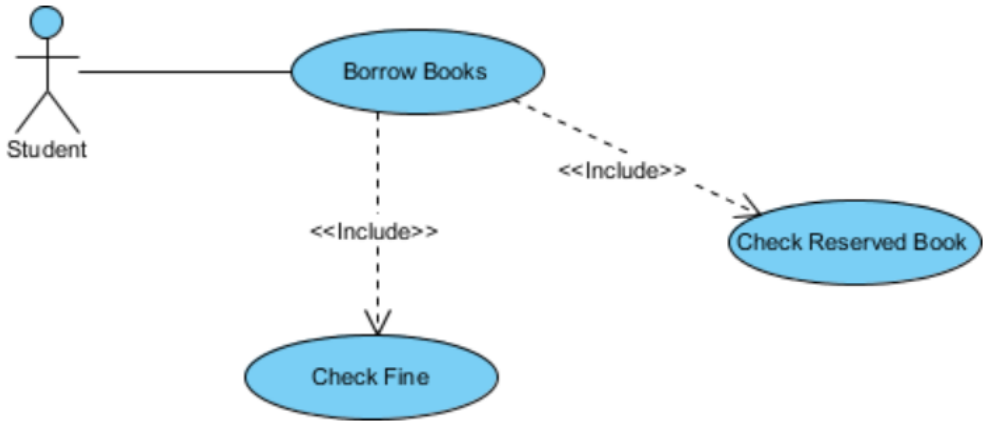
- **UML use case diagrams are ideal for:**
  1. Representing the goals of system-user interactions
  2. Defining and organizing functional requirements in a system
  3. Specifying the context and requirements of a system
  4. Modelling the basic flow of events in a use case

Usecase Diagram symbols and notations

Use cases	Horizontally shaped ovals that represent the different uses that a user might have.	
Actors	Stick figures that represent the people actually employing the use cases.	
Associations:	A line between actors and use cases. In complex diagrams, it is important to know which actors are associated with which use cases.	

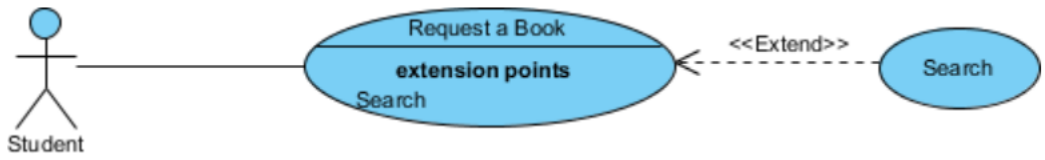
- **Include Relationship**

  The include relationship adds additional functionality not specified in the base use case. The <<Include>> relationship is used to include common behavior from an included use case into a base use case in order to support the reuse of common behaviour.

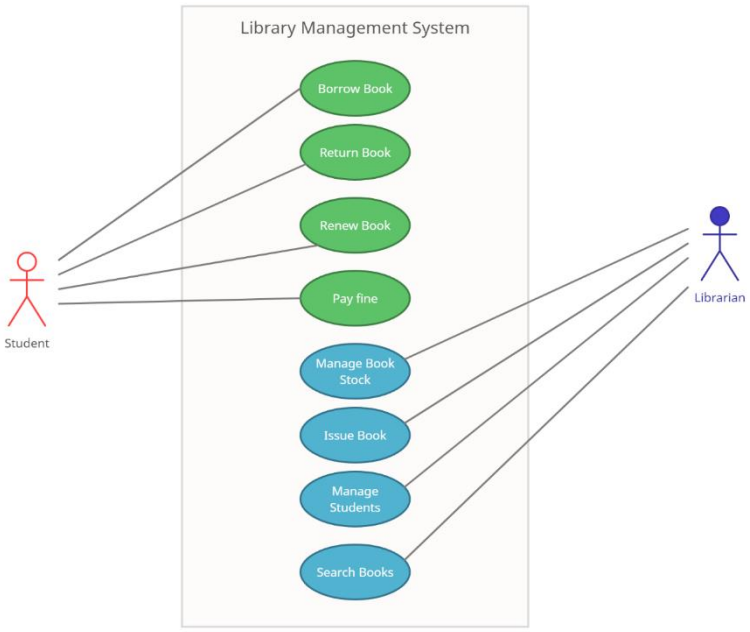


❖ **Extend Relationship**

The extend relationships are important because they show optional functionality or system behaviour. The <<extend>> relationship is used to include optional behaviours from an extending use case in an extended use case. Take a look at the use case diagram example below. It shows an extend connector and an extension point "Search".



An Example of Use case Diagram for Library Management system



H. Experimental set up/ Program Logic-Flow chart:

N/A

I. Resources/Equipment Required

Sr. No.	Instrument/Equipment /Components/Trainer kit	Specification	Quantity
1	Laptop/Desktop with Internet Connection	Any processor above i3 and any operating system	1

J. Safety and necessary Precautions followed.

1. Keep all computers in the lab area clean and free from dust.
2. Ensure all computers are properly grounded and plugged into surge protectors.
3. Use antivirus software to protect computers from malicious software and viruses.
4. Provide clear instructions and demonstration on how to use the tool.
5. Establish a safe and comfortable environment for students to work in.
6. Refrain from using computers in the lab for activities that could be deemed as inappropriate or illegal.

K. Procedure to be followed/Source code/Lab Task:

a) Prepare USE case diagram for selected system..

L. Observations and Calculations/Input-Output:

N/A

M. Practical Related Quiz/Questions:

1. UML diagram that shows the interaction between users and system, is known as _____.
2. which among the following can be heuristic for Use case diagram?

   a) The product can be made actor b) Never name actors with noun phrases.

   c) Name Use cases with verb phrases d) All of the mentioned

3. In a Usecase diagram for a system, an Actor can be placed inside the system boundary.

   a) True b) False

4. The <<include>> relationship shows.
   a) Behaviour of one usecase that does not stand alone but it is incorporated as part of some larger use case.
   b) One usecase inherits the behaviour of another use case.
   c) That a use case optionally provides additional functionality to another use case at certain points
   d) One use case includes the behaviour of an external actor in its behaviour.

N. References / Suggestions

1. www.visual-paradigm.com › what-is-use-case-diagram
2. https://www.lucidchart.com/pages/how-to-draw-a-use-case-diagram-in-UML
3. Software Engineering by Rajib Mall: https://docs.google.com/file/d/0B6JYmHy5NB8uQWxlaFhDenpHLWs/edit?resourcekey=0-jspVps0XuZQoIbHi5WtrCQ

O. Assessment-Rubrics

	Excellent (5)	Very Good (4)	Good (3)	Satisfactory (2)	Poor (1)
Engagement	Students proactively contributes to practical session by offering thoughts & asking questions more than once per session	Students proactively contributes to practical session by offering thoughts & asking questions only once per session	Students actively contributes to practical session but rarely ask question.	Students rarely contributes to practical session and rarely ask question.	Students never contributes to practical session and rarely ask question.
Understanding & Explanation	Fully understood the practical and able to explain it precisely	Fully understood the practical but not able to explain it precisely	Understood the practical and able to explain it little.	Barely understood the practical and not able to explain it.	Not able to understand correctly and not able to explain it.
Time	Completed work within the deadline	Completed work one week later than the deadline	Completed work two week later than the deadline	Completed work three week later than the deadline	Completed work during submission.
Documentation	Documentation is done precisely.	Documentation is done properly with little bit formatting mistakes.	Documentation is done properly with little bit formatting, grammatical and spelling mistakes.	Documentation is done with too much formatting, grammatical and spelling mistakes.	Documentation is not done properly.
Viva (Q&A)	Answered all the questions with proper justification.	Answered 60% of questions properly.	Answered 40% of questions properly.	Answered 20% of questions properly.	Not answered any of questions properly.

Sign with Date

Date: ……

Practical No:9	**Prepare Activity diagrams of the selected system for object-oriented design.**

A. Objective: The objective of Activity diagram is to describe behaviours of a system by depicting the sequencing of events through workflow. They illustrate what happens in workflow, what activities can be done in parallel and whether there are alternative paths through the workflow.

B. Expected Program Outcomes (POs)

PO2: Problem analysis
PO3: Design/ development of solutions
PO4: Engineering Tools, Experimentation and Testing

C. Expected Skills to be developed based on competency:

This practical is expected to develop the following skills for the industry identified competency:

1. Analysing and organizing system process flow.

D. Expected Course Outcomes (Cos)

CO5: Prepare a design of the software with user interface.

E. Practical Outcome (PRo)

Design an activity diagram that illustrates a business process or workflow of the system.

F. Expected Affective domain Outcome (ADos)

1. Work as a leader/ a team member
2. Follow ethical practice.

G. Prerequisite Theory:

❖ **What is Activity Diagram?**

Activity diagrams, along with use case and state machine diagrams, are considered behaviour diagrams because they describe what must happen in the system being modelled.

❖ **Why Activity Diagram?**

Activity diagrams help people on the business and development sides of an organization come together to understand the same process and behaviour. You'll use

a set of specialized symbols—including those used for starting, ending, merging, or receiving steps in the flow—to make an activity diagram.

❖ **Activity diagrams present several benefits to users. Consider creating an activity diagram to:**

1. Demonstrate the logic of an algorithm.
2. Describe the steps performed in a UML use case.
3. Illustrate a business process or workflow between users and the system.
4. Simplify and improve any process by clarifying complicated use cases.
5. Model software architecture elements, such as method, function, and operation.

❖ **Basic Components of an Activity Diagram**

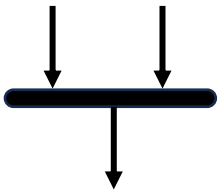
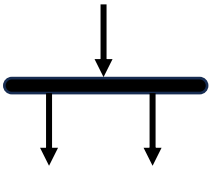
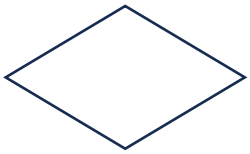
Some of the most common components of an activity diagram include:

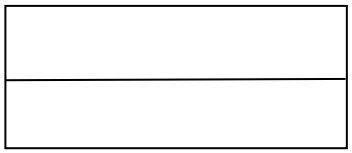
- **Action:** A step in the activity wherein the users or software perform a given task. actions are symbolized with round-edged rectangles.
- **Decision node:** A conditional branch in the flow that is represented by a diamond. It includes a single input and two or more outputs.
- **Control flows:** Another name for the connectors that show the flow between steps in the diagram.
- **Start node:** Symbolizes the beginning of the activity. The start node is represented by a black circle.
- **End node:** Represents the final step in the activity. The end node is represented by an outlined black circle.

❖ **Activity diagram symbols**

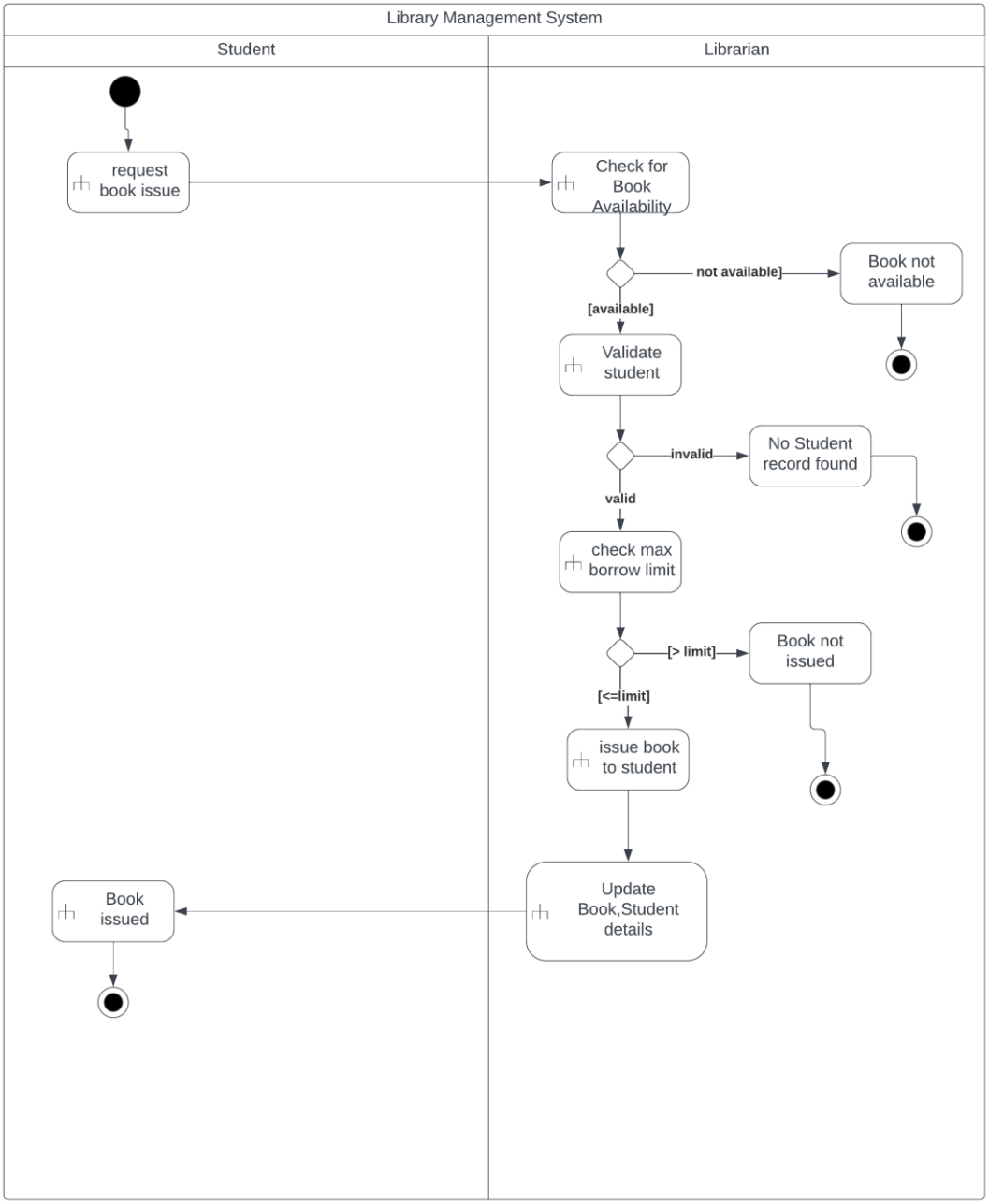
These activity diagram shapes and symbols are some of the most common types you'll find in UML diagrams.

Symbol	Name	Description
	Start symbol	Represents the beginning of a process or workflow in an activity diagram. It can be used by itself or with a note symbol that explains the starting point.
	Activity symbol	Indicates the activities that make up a modeled process. These symbols, which include short descriptions within the shape, are the main building blocks of an activity diagram.

	Connector symbol	Shows the directional flow, or control flow, of the activity. An incoming arrow starts a step of an activity; once the step is completed, the flow continues with the outgoing arrow.
	Joint symbol/ Synchronization bar	Combines two concurrent activities and re-introduces them to a flow where only one activity occurs at a time. Represented with a thick vertical or horizontal line.
	Fork symbol	Splits a single activity flow into two concurrent activities. Symbolized with multiple arrowed lines from a join.
	Decision symbol	Represents a decision and always has at least two paths branching out with condition text to allow users to view options. This symbol represents the branching or merging of various flows with the symbol acting as a frame or container.
[condition Text]	Condition text	Placed next to a decision marker to let you know under what condition an activity flow should split off in that direction.
	Note symbol	Allows the diagram creators or collaborators to communicate additional messages that don't fit within the diagram itself. Leave notes for added clarity and specification.
	End symbol	Marks the end state of an activity and represents the completion of all flows of a process.

	Swim Lanes	The swim lane is used to cluster all the related activities in one column or one row. It can be either vertical or horizontal. It used to add modularity to the activity diagram.

Example of Activity Diagram for Library management System.



H. Experimental set up/ Program Logic-Flow chart:

N/A

I. Resources/Equipment Required

Sr. No.	Instrument/Equipment /Components/Trainer kit	Specification	Quantity
1	Laptop/Desktop with Internet Connection	Any processor above i3 and any operating system	1

J. Safety and necessary Precautions followed.

1. Keep all computers in the lab area clean and free from dust.
2. Ensure all computers are properly grounded and plugged into surge protectors.
3. Use antivirus software to protect computers from malicious software and viruses.
4. Provide clear instructions and demonstration on how to use the tool.
5. Establish a safe and comfortable environment for students to work in.
6. Refrain from using computers in the lab for activities that could be deemed as inappropriate or illegal.

K. Procedure to be followed/Source code/Lab Task:

a) Design an activity diagram which depicts the process or workflow of the selected system.

L. Observations and Calculations/Input-Output:

N/A

M. Practical Related Quiz/Questions:

1. Which type of diagram is Activity diagram?

   a) Structural Diagram b) Behavioral UML Diagrams

   c) Both of these d) None of these

2. Which of the following Combines two concurrent activities and re-introduces them to a flow where only one activity can be performed at a time?

   a)Joint symbol b) Fork symbol c) Decision symbol d)Note symbol

3. Activity diagrams are used to model the processing of data.

   a)True b) False

N. References / Suggestions

1. www.visual-paradigm.com › what-is-use-case-diagram
2. https://www.lucidchart.com/pages/how-to-draw-a-use-case-diagram-in-UML
3. Software Engineering by Rajib Mall: https://docs.google.com/file/d/0B6JYmHy5NB8uQWxlaFhDenpHLWs/edit?resourcekey=0-jspVps0XuZQoIbHi5WtrCQ

O. Assessment-Rubrics

	Excellent (5)	Very Good (4)	Good (3)	Satisfactory (2)	Poor (1)
Engagement	Students proactively contributes to practical session by offering thoughts & asking questions more than once per session	Students proactively contributes to practical session by offering thoughts & asking questions only once per session	Students actively contributes to practical session but rarely ask question.	Students rarely contributes to practical session and rarely ask question.	Students never contributes to practical session and rarely ask question.
Understanding & Explanation	Fully understood the practical and able to explain it precisely	Fully understood the practical but not able to explain it precisely	Understood the practical and able to explain it little.	Barely understood the practical and not able to explain it.	Not able to understand correctly and not able to explain it.
Time	Completed work within the deadline	Completed work one week later than the deadline	Completed work two week later than the deadline	Completed work three week later than the deadline	Completed work during submission.
Documentation	Documentation is done precisely.	Documentation is done properly with little bit formatting mistakes.	Documentation is done properly with little bit formatting, grammatical and spelling mistakes.	Documentation is done with too much formatting, grammatical and spelling mistakes.	Documentation is not done properly.
Viva (Q&A)	Answered all the questions with proper justification.	Answered 60% of questions properly.	Answered 40% of questions properly.	Answered 20% of questions properly.	Not answered any of questions properly.

Sign with Date

Date: ……

Practical No:10	**Design appropriate User Interface based on the type of project.**

A. Objective: The objective of UI design is to focuses on anticipating what users might need to do and ensuring that the interface has elements that are easy to access, understand, and use to facilitate those actions.

B. Expected Program Outcomes (POs)

PO2: Problem analysis
PO3: Design/ development of solutions
PO4: Engineering Tools, Experimentation and Testing
PO5: Engineering practices for society, sustainability, and environment.

C. Expected Skills to be developed based on competency:

This practical is expected to develop the following skills for the industry identified competency:

1. Creativity and design abilities.
2. Technical, Planning, Team-based, communication skill.

D. Expected Course Outcomes (Cos)

CO5: Prepare a design of the software with user interface.

E. Practical Outcome (PRo)

Design command line or graphical user interface through which user can easily interact with system.

F. Expected Affective domain Outcome (ADos)

1. Work as a leader/ a team member
2. Follow ethical practice.

G. Prerequisite Theory:

- **What is User Interface?**

User interface is the front-end application view to which user interacts in order to use the software.

- **Characteristics of Good User Interface**
  a. Attractive
  b. Simple to use.

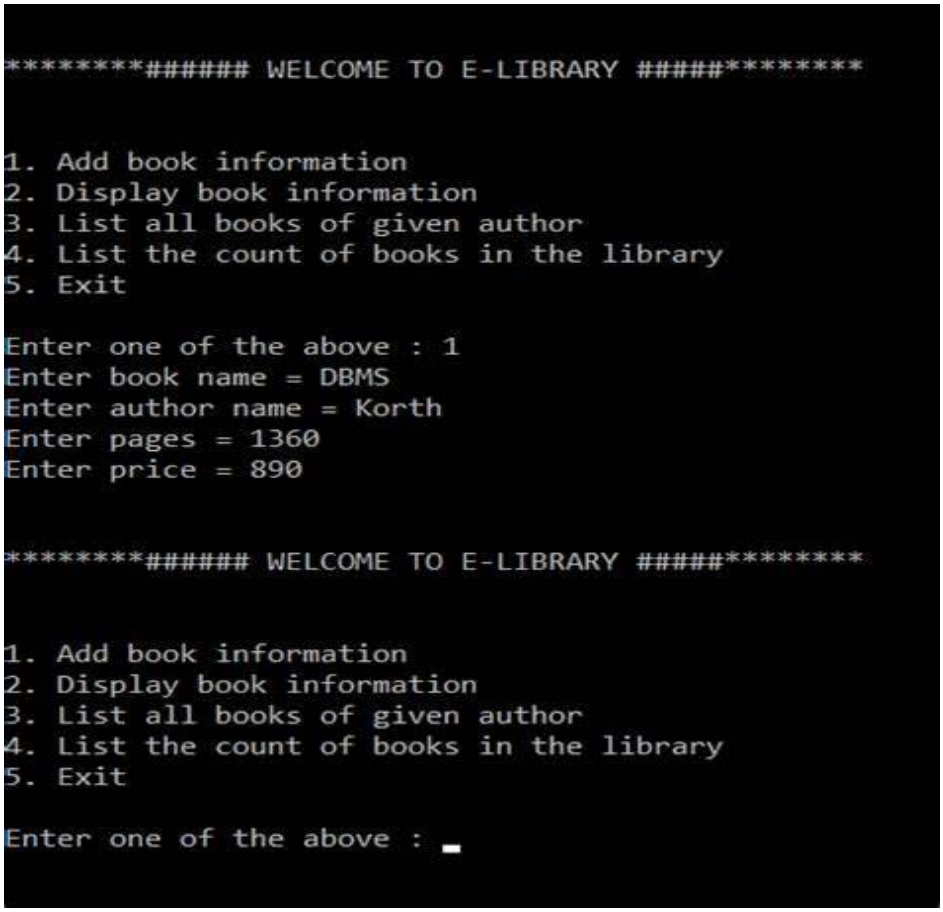
c. Responsive in short time

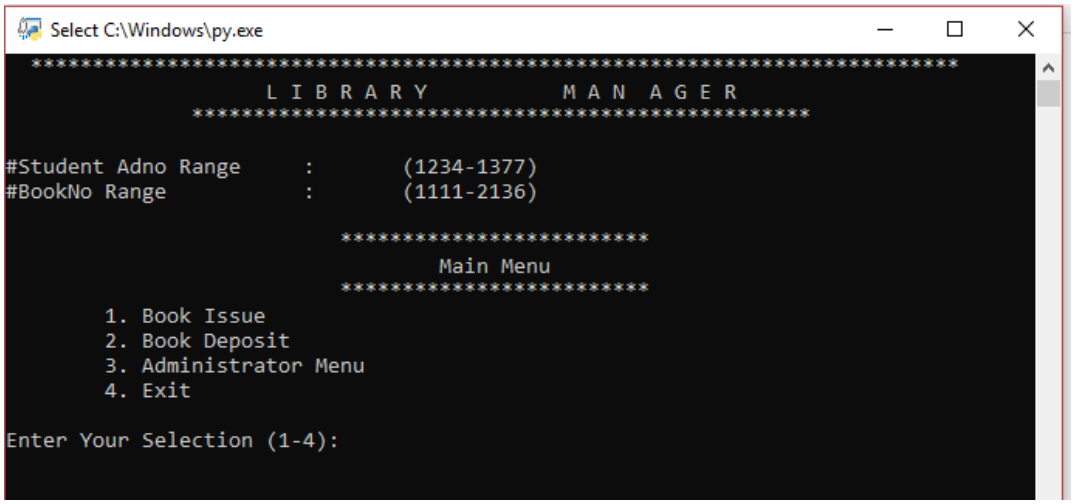
d. Clear to understand.

e. Consistent on all interface screens

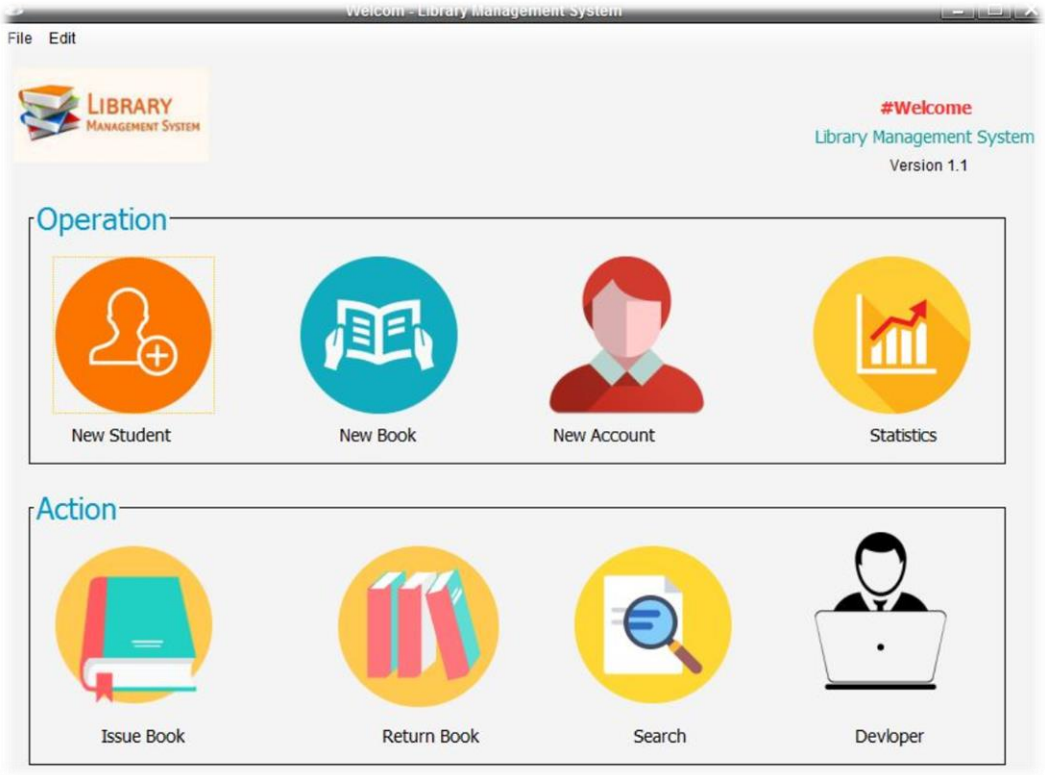
❖ **There are two types of User Interface:**

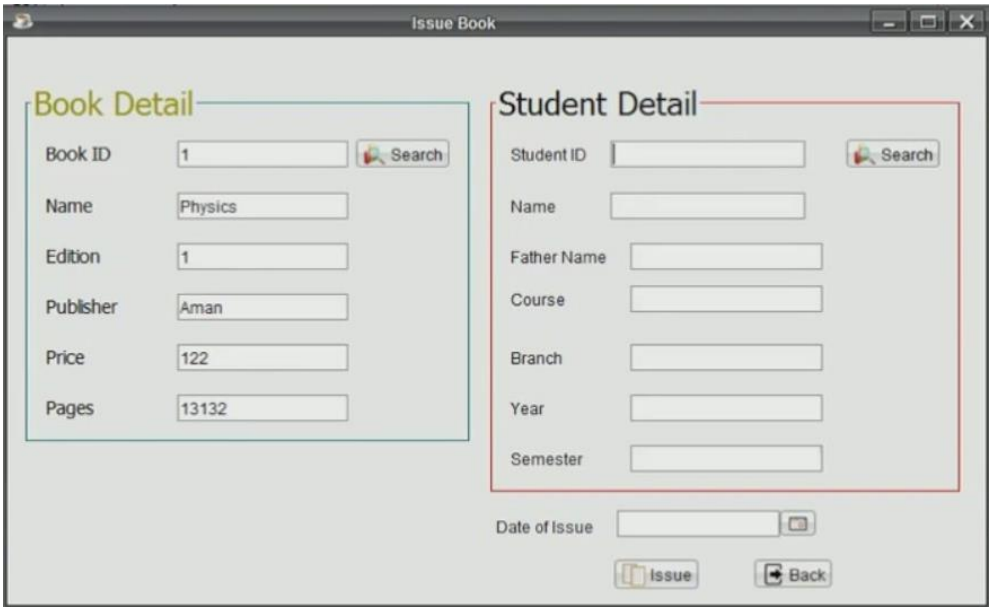
1. **Command Line Interface:** Command Line Interface provides a command prompt, where the user types the command and feeds to the system. The user needs to remember the syntax of the command and its use.





2. **Graphical User Interface:** Graphical User Interface provides a simple interactive interface to interact with the system. GUI can be a combination of both hardware and software. Using GUI, the user interprets the software.





H. Experimental set up/ Program Logic-Flow chart:

N/A

I. Resources/Equipment Required

Sr. No.	Instrument/Equipment /Components/Trainer kit	Specification	Quantity
1	Laptop/Desktop with Internet Connection	Any processor above i3 and any operating system	1

J. Safety and necessary Precautions followed.

1. Keep all computers in the lab area clean and free from dust.
2. Ensure all computers are properly grounded and plugged into surge protectors.
3. Use antivirus software to protect computers from malicious software and viruses.
4. Provide clear instructions and demonstration on how to use the tool.
5. Establish a safe and comfortable environment for students to work in.
6. Refrain from using computers in the lab for activities that could be deemed as inappropriate or illegal.

K. Procedure to be followed/Source code/Lab Task:

a) Design graphical user interface for selected system. [Students can design user interface using online designing tool or use (html +css) for designing web interface].

L. Observations and Calculations/Input-Output:

N/A

M. Practical Related Quiz/Questions:

1. A software might allow a user to interact via
   a) keyboard commands
   b) mouse movement
   c) voice recognition commands
   d) all the mentioned

2. List down User interface design principal.

   _____

3. Which of the following is golden rule for interface design?
   a) Place the user in control
   b) Reduce the user's memory load
   c) Make the interface consistent
   d) All of the mentioned.

N. References / Suggestions

1. https://www.javatpoint.com/software-engineering-user-interface-design
2. Software Engineering by Rajib Mall: https://docs.google.com/file/d/0B6JYmHy5NB8uQWxlaFhDenpHLWs/edit?resourcekey=0-jspVps0XuZQoIbHi5WtrCQ

O. Assessment-Rubrics

	Excellent (5)	Very Good (4)	Good (3)	Satisfactory (2)	Poor (1)
Engagement	Students proactively contributes to practical session by offering thoughts & asking questions more than once per session	Students proactively contributes to practical session by offering thoughts & asking questions only once per session	Students actively contributes to practical session but rarely ask question.	Students rarely contributes to practical session and rarely ask question.	Students never contributes to practical session and rarely ask question.
Understanding & Explanation	Fully understood the practical and able to explain it precisely	Fully understood the practical but not able to explain it precisely	Understood the practical and able to explain it little.	Barely understood the practical and not able to explain it.	Not able to understand correctly and not able to explain it.
Time	Completed work within the deadline	Completed work one week later than the deadline	Completed work two week later than the deadline	Completed work three week later than the deadline	Completed work during submission.
Documentation	Documentation is done precisely.	Documentation is done properly with little bit formatting mistakes.	Documentation is done properly with little bit formatting, grammatical and spelling mistakes.	Documentation is done with too much formatting, grammatical and spelling mistakes.	Documentation is not done properly.
Viva (Q&A)	Answered all the questions with proper justification.	Answered 60% of questions properly.	Answered 40% of questions properly.	Answered 20% of questions properly.	Not answered any of questions properly.

Sign with Date

Date: ……

Practical No:11	**Prepare the test cases to test the functionalities of the modules.**

A. Objective: The main objective of the Software Testing is to identify and report any errors or issues in software applications at the earliest.

B. Expected Program Outcomes (POs)

PO2: Problem analysis
PO3: Design/ development of solutions
PO4: Engineering Tools, Experimentation and Testing

C. Expected Skills to be developed based on competency:

This practical is expected to develop the following skills for the industry identified competency:

1. Analytical and logical thinking.
2. Project management, Team-based, communication skill.

D. Expected Course Outcomes (Cos)

CO6: Apply testing on software product with proper test cases.

E. Practical Outcome (PRo)

Prepare effective test cases to produce a good quality software product.

F. Expected Affective domain Outcome (ADos)

1. Work as a leader/ a team member
2. Follow ethical practice.

G. Prerequisite Theory:

❖ **What is Software testing?**

Software testing can be stated as the process of verifying and validating whether a software or application is bug-free, meets the technical requirements as guided by its design and development, and meets the user requirements effectively and efficiently by handling all the exceptional and boundary cases.

❖ **Why testing?**

Software testing is the end of application development through which software testers evaluate code by questioning it. This evaluation can be brief or proceed until all stakeholders are satisfied.

Software testing identifies bugs and issues in the development process so they're fixed prior to product launch.
This approach ensures that only quality products are distributed to consumers, which in turn elevates customer satisfaction and trust.

- **What is a Test Case?**

  A test case serves as a predefined set of conditions or activities designed to validate the functionality of a software program or system.
  It contains description of the inputs, preconditions, steps to be executed, and expected outcomes.
  Test cases are written to target specific aspects or functionalities of the software, ensuring that they align with the requirements and meet the desired objectives.

- **Parameters of a Test Case:**

Module Name:	Subject or title that defines the functionality of the test.
Test Case Id:	A unique identifier assigned to every single condition in a test case.
Test scenario:	The test scenario provides a small overview to know about what needs to be performed.
Test Case Description:	The condition required to be checked for a given software. for eg. Check if only numbers validation is working or not for an age input box.
Test Steps:	Steps to be performed for the checking of the condition.
Prerequisite:	The conditions required to be fulfilled before the start of the test process.
Test Data:	The inputs to be taken while checking for the conditions.
Test Expected Result:	The output which should be expected at the end of the test.
Actual Result:	The output that is displayed at the end.
Status:	The status of tests such as pass, fail, NA, etc.

Sample Test case Example:

1)Login functionality of Library Management System.

Module Name	Login Functionality(Librarian)					
Test Case ID	Test_M1_1					
Test Scenario	Testing Login functionality for Librarian					
Test Case Description	**Steps**	**Prerequisite**	**Test Data**	**Expected Result**	**Actual Result**	**Status**
Verify login functionality with valid username and password	Navigate to login page			able to see login page	as expected	pass
	Enter valid username	valid username	username:xyz.400@gmail.com	credential can be entered	as expected	pass
	Enter valid password	valid password	password:xxxx@1	credential can be entered	as expected	pass
	click on login button			user logged in	user loggen in successful	pass
Verify login functionality with valid username and invalid password	Navigate to login page			able to see login page	as expected	pass
	Enter valid username	valid username	username:xyz.400@gmail.com	credential can be entered	as expected	pass
	Enter valid password	in valid password	password:xxxx@2	credential can be entered	as expected	pass
	click on login button			user logged in	unsuccessful login	fail

2) Book Searching Functionality with book title

Module Name	Book Searching functionality(Librarian)					
Test Case ID	Test_M1_2					
Test Scenario	Testing book searching functionality with different options					
Test Case Description	**Steps**	**Prerequisite**	**Test Data**	**Expected Result**	**Actual Result**	**Status**
Perform book searching based on valid book title	Navigate to book search page					
	Enter valid book title	valid book title	ANSI C	Book details can be entered	as expected	pass
	click on search button			Book details should be displayed	book details display	pass

H. Experimental set up/ Program Logic-Flow chart:

N/A

I. Resources/Equipment Required

Sr. No.	Instrument/Equipment /Components/Trainer kit	Specification	Quantity
1	Laptop/Desktop with Internet Connection	Any processor above i3 and any operating system	1

J. Safety and necessary Precautions followed.

1. Keep all computers in the lab area clean and free from dust.
2. Ensure all computers are properly grounded and plugged into surge protectors.
3. Use antivirus software to protect computers from malicious software and viruses.
4. Provide clear instructions and demonstration on how to use the tool.
5. Establish a safe and comfortable environment for students to work in.
6. Refrain from using computers in the lab for activities that could be deemed as inappropriate or illegal.

K. Procedure to be followed/Source code/Lab Task:

a) Prepare test cases for different modules of selected system.

L. Observations and Calculations/Input-Output:

N/A

M. Practical Related Quiz/Questions:

1. Which of the following term describes testing?

   a) Finding broken code b) Evaluating deliverable to find errors

   c) A stage of all projects d) None of the mentioned

2. In software, a ____ occurred when the expected output does not match the output provided by the software.

   a) Correctness b) Defect c) Confusion

3. A set of inputs, execution preconditions and expected outcomes is known as

   a) Test Plan b) Test case c) Test document d) Test suit

N. References / Suggestions

1. Software Engineering by Rajib Mall: https://docs.google.com/file/d/0B6JYmHy5NB8uQWxlaFhDenpHLWs/edit?resourcekey=0-jspVps0XuZQoIbHi5WtrCQ
2. **https://www.geeksforgeeks.org/software-testing-test-case/**
3. **https://www.edureka.co/blog/test-case-in-software-testing/**

O. Assessment-Rubrics

	Excellent (5)	Very Good (4)	Good (3)	Satisfactory (2)	Poor (1)
Engagement	Students proactively contributes to practical session by offering thoughts & asking questions more than once per session	Students proactively contributes to practical session by offering thoughts & asking questions only once per session	Students actively contributes to practical session but rarely ask question.	Students rarely contributes to practical session and rarely ask question.	Students never contributes to practical session and rarely ask question.
Understanding & Explanation	Fully understood the practical and able to explain it precisely	Fully understood the practical but not able to explain it precisely	Understood the practical and able to explain it little.	Barely understood the practical and not able to explain it.	Not able to understand correctly and not able to explain it.
Time	Completed work within the deadline	Completed work one week later than the deadline	Completed work two week later than the deadline	Completed work three week later than the deadline	Completed work during submission.

Documentation	Documentation is done precisely.	Documentation is done properly with little bit formatting mistakes.	Documentation is done properly with little bit formatting, grammatical and spelling mistakes.	Documentation is done with too much formatting, grammatical and spelling mistakes.	Documentation is not done properly.
Viva (Q&A)	Answered all the questions with proper justification.	Answered 60% of questions properly.	Answered 40% of questions properly.	Answered 20% of questions properly.	Not answered any of questions properly.

Sign with Date

Fundamentals of Software Development

4331604

Lab manual

Prepared By:

Mr. Pragnesh Patel

Lecturer

Government Polytechnic for Girls, Surat

Mr. Dhaval Gandhi

Lecturer

Dr.S & S.S Gandhi College (Diploma), Surat

Mr. Umang Shukla

Lecturer

Dr.S & S.S Gandhi College (Diploma), Surat

Branch Coordinator

Mr. Nandu Fatak

HOD(IT)

Government Polytechnic for Girls, Ahmedabad

Committee Chairman

Shri R. D. Raghani

(HOD-EC)

Principal (I/C)

Government Polytechnic, Gandhinagar