# **DEPARTMENT OF INFORMATION TECHNOLOGY**

**LAB MANUAL FILE**

**Name :** Software Engineering

**Course Code:** IT-6002

**Course :**  B.TECH

**Session :**

**Prepared By Prof. nAME**

**Student Name Submitted to**

**RollNo**

**LIST OF EXPERIMENTS**

|  |  |
| --- | --- |
| **S.No.** | **EXPERIMENTS** |
| **1.** | Identifying the Requirements from Problem Statements. |
| **2.** | How to Choose right Software development life cycle model? |
| **3.** | To perform the system analysis: Requirement analysis, SRS for Library Management System. |
| **4.** | To perform the function oriented diagram: DFD. |
| **5.** | To perform the user’s view analysis: Use case diagram. |
| **6.** | To draw the structural view diagram: Class diagram. |
| **7.** | To draw the behavioral view diagram: Sequence diagram. |
| **8.** | Explain testing Strategies. |
| **9.** | To design a test case for square root problem. |
| **10.** | To compute cyclometic complexity for any flow graph. |

Note: Implementation of the programs is to be done using star UML 5.02 or rationale rose.

**INDEX**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No.** | **EXPERIMENTS** | **DATE** | **Signature** |
| **1.** | Identifying the Requirements from Problem Statements. |  |  |
| **2.** | How to Choose right Software development life cycle model? |  |  |
| **3.** | To perform the system analysis: Requirement analysis, SRS for Library Management System. |  |  |
| **4.** | To perform the function oriented diagram: DFD. |  |  |
| **5.** | To perform the user’s view analysis: Use case diagram. |  |  |
| **6.** | To draw the structural view diagram: Class diagram. |  |  |
| **7.** | To draw the behavioral view diagram: Sequence diagram. |  |  |
| **8.** | Explain testing Strategies. |  |  |
| **9.** | To design a test case for square root problem. |  |  |
| **10.** | To compute cyclometic complexity for any flowgraph. |  |  |

**EXPERIMENT-1**

**Aim:** Identifying the Requirements from Problem Statements.

**A Library Information System**

The Oriental Group of Institutes has been recently setup to provide state-of-the-art research facilities in the field of Software Engineering. Apart from research scholars (students) and professors, it also includes quite a large number of employees who work on different projects undertaken by the institution.

As the size and capacity of the institute is increasing with the time, it has been proposed to develop a Library Information System (LIS) for the benefit of students and employees of the institute. LIS will enable the members to borrow a book (or return it) with ease while sitting at his desk/chamber. The system also enables a member to extend the date of his borrowing if no other booking for that particular book has been made. For the library staff, this system aids them to easily handle day-to-day book transactions. The librarian, who has administrative privileges and complete control over the system, can enter a new record into the system when a new book has been purchased, or remove a record in case any book is taken off the shelf. Any non-member is free to use this system to browse/search books online. However, issuing or returning books is restricted to valid users (members) of LIS only.

The final deliverable would a web application, which should run only within the institute LAN. Although this reduces security risk of the software to a large extent, care should be taken no confidential information (eg., passwords) is stored in plain text.

**EXPERIMENT-2**

**Aim**: How to choose right Software development life cycle model?

Selecting a Software Development Life Cycle (SDLC) methodology is a challenging task for many organizations and software engineers. What tends to make it challenging is the fact that few organizations know what are the criteria to use in selecting a methodology to add value to the organization. Fewer still understand that a methodology might apply to more than one Life Cycle Model. Before considering a framework for selecting a given SDLC methodology, we need to define the different types and illustrate the advantages and disadvantages of those models.

**EXPERIMENT-3**

**Aim**: To perform the system analysis: Requirement analysis, SRS for Library Management System.

**EXPERIMENT-4**

**Aim**: To perform the function oriented diagram: Data Flow Diagram.

**Tools/Apparatus** Smart Draw **Procedure**:

1. Identify various processes, data store, input output etc. of the system and ask students to analyze.
2. Use processes at various levels to draw the DFDs.
3. Identify various modules, input, output etc. of the system and ask students to analyze.

* **Data flow diagrams:** illustrate how data is processed by a system in terms of inputs and outputs. A **Date Flow Diagram (DFD)** is a diagrammatic representation of the information (data) flow within a system.
  + **Process**: A process transforms incoming data flow into outgoing data flow.
  + **Data Store:** Data stores are repositories of data in the system. They are sometimes also referred to as files.
  + **Dataflow:** Dataflows are pipelines through which packets of information flow. Label the arrows with the name of the data that moves through it.
  + **External Entities:** External entities are objects outside the system, with which the system communicates. External entities are sources and destinations of the system's inputs and outputs.

**Description:**

* **Data Flow:**

|  |  |  |
| --- | --- | --- |
| **S.No** | **Data Flow** | **Description** |
| 1 | Login | Login data usually contains the ATM card and the PIN number. |
| 2 | Accept Info | If the Account is verified then the user control can be accept inside the transaction procedure. |
| 3 | Reject Info | It the account details are not correct,do not grant the permission to access transaction and reject the card. |
| 4 | Account Details | The details of account are forwarded as the data to be printed in the receipt. |
| 5 | Money | The Money is given to the user. |

Figure Details of Data Flow

* **Function:**

|  |  |  |  |
| --- | --- | --- | --- |
| **S no** | **Function** | **Description** | |
| 1 | Check Account | **Input:** **a**. Login Details. | Compare the data with the Bank’s Database and grant access. |
| **Output: a**. Accept Info. **b**. Reject Info. |
| 2 | Prompts Amount | **Input: a.** accept info. | If the user is valid, then prompt user to enter the amount. |
| **Output: a.** account info. |
| 3 | Update Database | **Input: a.** account info. | When user enters the amount then update the database and perform the transaction. |
| **Output: a.** account details. |
| 4 | Print Details | **Input: a.** account details. | The details of the changes are printed as the new details of the account. |
| **Output: a.** Money. |

Figure Functions of Data Flow diagram

**EXPERIMENT-5**

**Aim**: To perform the user’s view analysis: Use case diagram

**Tools/Apparatus**: Rational rose/Star UML 5.02

**Procedure**:

1. Identify various processes, use-cases, actors etc. of the system and ask students to analyze.
2. Use processes at various levels to draw the use-case diagram.

A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different [use cases](https://en.wikipedia.org/wiki/Use_case) in which the user is involved. A use case diagram can identify the different types of users of a system and the different use cases and will often be accompanied by other types of diagrams as well.

**EXPERIMENT-6**

**Aim:** To draw the structural view diagram: Class diagram.

**Tools/Apparatus**:

Rational rose/Star UML 5.02 **Procedure**:

1. Identify various elements such as classes, member variables, member functions etc. of the class diagram
2. Draw the class diagram as per the norms.
3. Identify various elements such as various objects of the object diagram
4. Draw the object diagram as per the norms.

Class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing, and documenting different aspects of a system but also for constructing executable code of the software application.

Class diagram describes the attributes and operations of a class and also the constraints imposed on the system. The class diagrams are widely used in the modeling of objectoriented systems because they are the only UML diagrams, which can be mapped directly with object-oriented languages.

Class diagram shows a collection of classes, interfaces, associations, collaborations, and constraints. It is also known as a structural diagram.

**EXPERIMENT-7**

**Aim:** To draw the behavioral view diagram: Sequence diagram.

**Tools/Apparatus**:

Rational rose/Star UML 5.02

**Procedure**:

1. Identify various elements such as controller class, objects, boundaries, messages etc. of the sequence diagram
2. Draw the sequence diagram as per the norms.
3. Identify various elements such as for the sequence diagram of the collaboration diagram
4. Draw the collaboration diagram as per the norms.

A sequence diagram is an interaction diagram that shows how objects operate with one another and in what order. It is a construct of a message sequence chart. A sequence diagram shows object interactions arranged in time sequence.

**EXPERIMENT-8**

**Aim:** Explain testing Strategies.

**EXPERIMENT-9**

**Aim:** To design a test case for square root problem.

**EXPERIMENT-10**

**Aim:** To compute cyclometic complexity for any flowgraph.

Cyclomatic complexity is a source code complexity measurement that is being Cyclomatic complexity is a source code complexity measurement that is being correlated to a number of coding errors. It is calculated by developing a Control Flow Graph of the code that measures the number of linearly-independent paths through a program module.

Lower the Program's cyclomatic complexity, lower the risk to modify and easier to understand. It can be represented using the below formula:

Cyclomatic complexity = E - N + 2\*P

where,

E = number of edges in the flow graph.

N = number of nodes in the flow graph.

P = number of nodes that have exit points