MAHENDRA COLLEGE OF ENGINEERING

Approved by AICTE and Affiliated by Anna University, Chennai NAAC
Accredited– Recognized U/S 2 (F) & 12 (B) of UGC Act 1956

Salem-Chennai Highway NH 79, Minnampalli, Salem-636106

DEPARTMENT OF INFORMATION TECHNOLOGY

REGISTER NUMBER:

STUDENT NAME :

SEMESTER :

YEAR :

SUBJECT CODE : CCS356

SUBJECT NAME : OBJECT ORIENTED SOFTWARE ENGINEERING LABORATORY

MAHENDRA COLLEGE OF ENGINEERING

SALEM-CAMPUS, ATTUR MAIN ROAD, MINNAMPALLI, SALEM -636 106.

INSTITUTION VISION AND MISSION

VISION

Mahendra College of Engineering is committed to be a leader in Higher Education achieving excellence through world class learning environment for Science and Technology with a blend of advanced research to create ethical and competent professionals.

MISSION

- To provide a conductive atmosphere to impart innovative knowledge and commendable skills through quality education by continuous improvement and customization of teaching.
- To nurture research attitude and bring about tangible developments with dynamic Industry - Institute Interaction.
- To create society oriented citizens with professional ethics.

MAHENDRA COLLEGE OF ENGINEERING

SALEM-CAMPUS, ATTUR MAIN ROAD, MINNAMPALLI, SALEM -636 106.
DEPARTMENT OF INFORMATION TECHNOLOGY

DEPARTMENT VISION AND MISSION

VISION

To become a department, producing graduates with good technical skills in emerging areas of Information Technology, through value based education and research.

MISSION

- To provide exposure to students to the emerging technologies in Hardware and Software.
- To inculcate students with sound application knowledge.
- To establish strong Industry- Institute Interaction.

PROGRAMME SPECIFIC OUTCOMES (PSOs)

To ensure graduates

- Have proficiency in programming skills to design, develop and apply appropriate techniques, to solve complex engineering problems.
- Have knowledge to build, automate and manage business solutions using cutting edge technologies.
- Have excitement towards research in applied computer technologies.

PROGRAM OUTCOMES (POs)

1. Engineering knowledge:

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

2. Problem analysis:

Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

3. Design/development of solutions:

Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal,

and environmental considerations.

4. Conduct investigations of complex problems:

Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

5. Modern tool usages:

Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

6. The engineer and society:

Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

7.Environment and sustainability:

Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

8. Ethics:

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

9. Individual and team work:

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

10. Communications:

Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

11. Project management and finance:

Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

12.Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

CCS356 –OBJECT ORIENTED SOFTWARE ENGINEERING LABORATORY

COURSE OBJECTIVES:

- To understand Software Engineering Lifecycle Models
- To Perform software requirements analysis
- To gain knowledge of the System Analysis and Design concepts using UML.
- To understand software testing and maintenance approaches
- To work on project management scheduling using DevOps

LIST OF EXPERIMENTS:

1. Identify a software system that needs to be developed.
2. Document the Software Requirements Specification (SRS) for the identified system.
3. Identify use cases and develop the Use Case model.
4. Identify the conceptual classes and develop a Domain Model and also derive a Class Diagram from that.
5. Using the identified scenarios, find the interaction between objects and represent them using UML Sequence and Collaboration Diagrams
6. Draw relevant State Chart and Activity Diagrams for the same system.
7. Implement the system as per the detailed design
8. Test the software system for all the scenarios identified as per the usecase diagram
9. Improve the reusability and maintainability of the software system by applying appropriate design patterns.
10. Implement the modified system and test it for various scenarios

TOTAL: 30 PERIODS

SOFTWARE AND HARDWARE REQUIREMENTS:

Software Requirements	Argo UML, Star UML- Opensource
Hardware Requirements	Desktop Computer

COURSE OUTCOMES:

At the end of the course, the student should be able to:

CO1: Compare various Software Development Lifecycle Models

CO2: Evaluate project management approaches as well as cost and schedule estimation strategies.

CO3: Perform formal analysis on specifications.

CO4: Use UML diagrams for analysis and design.

CO5: Architect and design using architectural styles and design patterns, and test the system

CO's-PO & PSO's MAPPING:

CO's	PO's												PSO's		
	1	2	3	4	5	6	7	8	9	10	11	12	1	2	3
1	2	2	1	2	2	-	-	-	-	1	1	2	2	2	1
2	2	3	2	3	2	-	-	-	2	2	3	2	3	2	1
3	2	3	2	1	1	-	-	-	2	2	3	2	2	3	1
4	2	3	2	2	3	-	-	-	2	2	3	2	2	3	1
5	2	3	1	2	2	-	-	-	-	-	-	1	3	2	2
AVg.	2	2	1	2	2	-	-	-	-	1	1	2	2	2	1

1-low,2-medium,3-high,'-'-nocorrelation

INDEX

S.NO	DATE	LIST OF EXPERIMENTS	PAGE NO	MARK	SIGN

S.NO	INDEX
1.	Study of ARGOUML & UML Diagrams
2.	PROBLEM STATEMENT - QUIZ SYSTEM
3.	USE CASE MODELING
4.	UML Class Diagram
5.	UML Interaction Diagram
6.	UML State Chart Diagram and Activity Diagram
7.	Logical Architecture with UML Package diagram
8.	Project Development & Testing
	PROBLEM STATEMENTS A. Passport automation system. B. Book bank C. Exam Registration D. Stock maintenance system. E. Online course reservation system F. E-ticketing G. Software personnel managementsystem H. Credit card processing I. e-book management system J. Recruitment system K. Foreign trading system L. Conference Management System M. BPO Management System N. Library Management System O. Student Information System

INTRODUCTION TO OOAD & ArgoUML

What is OOAD ?

Object-oriented analysis and design (OOAD) is a software engineering approach that models a system as a group of interacting objects. Each object represents some entity of interest in the system being modeled, and is characterized by its class, its state (data elements), and its behavior. There are a number of different notations for representing these models, such as the Unified Modeling Language (UML).

Object-Oriented analysis

Object-oriented analysis (OOA) looks at the problem domain, with the aim of producing a conceptual model of the information that exists in the area being analyzed. Implementation constraints are dealt during object-oriented design (OOD).

Object-oriented design

Object-oriented design (OOD) transforms the conceptual model produced in object-oriented analysis to take account of the constraints imposed by the chosen architecture and any non-functional – technological or environmental – constraints, such as transaction throughput, response time, run-time platform, development environment, or programming language.

ArgoUML

ArgoUML is an UML diagramming application written in Java and released under the open source Eclipse Public License. By virtue of being a Java application, it is available on any platform supported by Java SE.

ArgoUML is different: i) it makes use of ideas from cognitive psychology, ii) it is based on open standards; iii) it is 100% pure Java; and iv) it is an open source project.

CASE Tools

Computer-aided software engineering (CASE) is the scientific application of a set of tools which is meant to result in high-quality, defect-free, and maintainable software products.

Ex: No: 1 Study of ArgoUML & UML Diagrams

Aim:

To study about case tools and ArgoUML

ArgoUML

ArgoUML is an open source Unified Modeling Language (UML) modeling tool that includes support for all standard UML 1.4 diagrams. It runs on any Java platform and is available in ten languages.

FEATURES

- Support open standards extensively: UML, XMI, SVG, OCL and others.
- 100% Platform independent thanks to the exclusive use of Java
- Open Source, which allows extending or customizing.
- Cognitive features like
  - reflection-in-action
    - Design Critics
    - Corrective Automations (partially implemented)
    - "To Do" List
    - User model (partially implemented)
  - opportunistic design
    - "To Do" List
    - Checklists
  - Comprehension and Problem Solving
    - Explorer Perspectives
    - Multiple, Overlapping Views
    - Alternative Design Representations: Graphs, Text, or Table

Supported Diagrams by ArgoUML

- Use Case Diagrams
- Class Diagrams
- Behavior Diagrams
  - State chart Diagrams
  - Activity Diagrams
  - Interaction Diagrams
    - » Sequence Diagrams
    - » Collaboration Diagrams
- Implementation Diagrams
  - Component Diagrams
  - Deployment Diagrams

Use Case Diagrams

- Present a high-level view of system usage
- These diagrams show the functionality of a system or a class and how the system interacts with the outside world.
- During analysis to capture the system requirements and to understand how the system should work.

During the design phase, use-case diagrams specify the behavior of the system as implemented.

Class Diagram

- Helps you visualize the structural or static view of a system.

- Class diagrams show the relationships among class.
- Foundation for component and deployment diagrams.

Sequence Diagram

- Illustrates object interacts arranged in a time sequence
- This shows step-by-step what has to happen to accomplish something in the use case and emphasize the sequence of events.

Collaboration Diagram

- Provides a view of the interactions or structured relationships between objects in current model.
- Emphasizes relation between objects.
- Used as the primary vehicle to describe interactions that express decision about system behavior.

State Chart Diagram

- To model the dynamic behaviors of individual classes or objects
- Used to model the discrete stages of an objects lifetime.

Activity Diagram

- Model the workflow of a business process and the sequence of activities in a process.
- Similar to a flowchart, l a workflow from activity to activity or from activity to state.
- It is help to understand the overall process.

Component Diagram

- A physical view of the current model and Show the organization and dependencies of software components, including source code, binary code, and executable components

Deployment Diagram

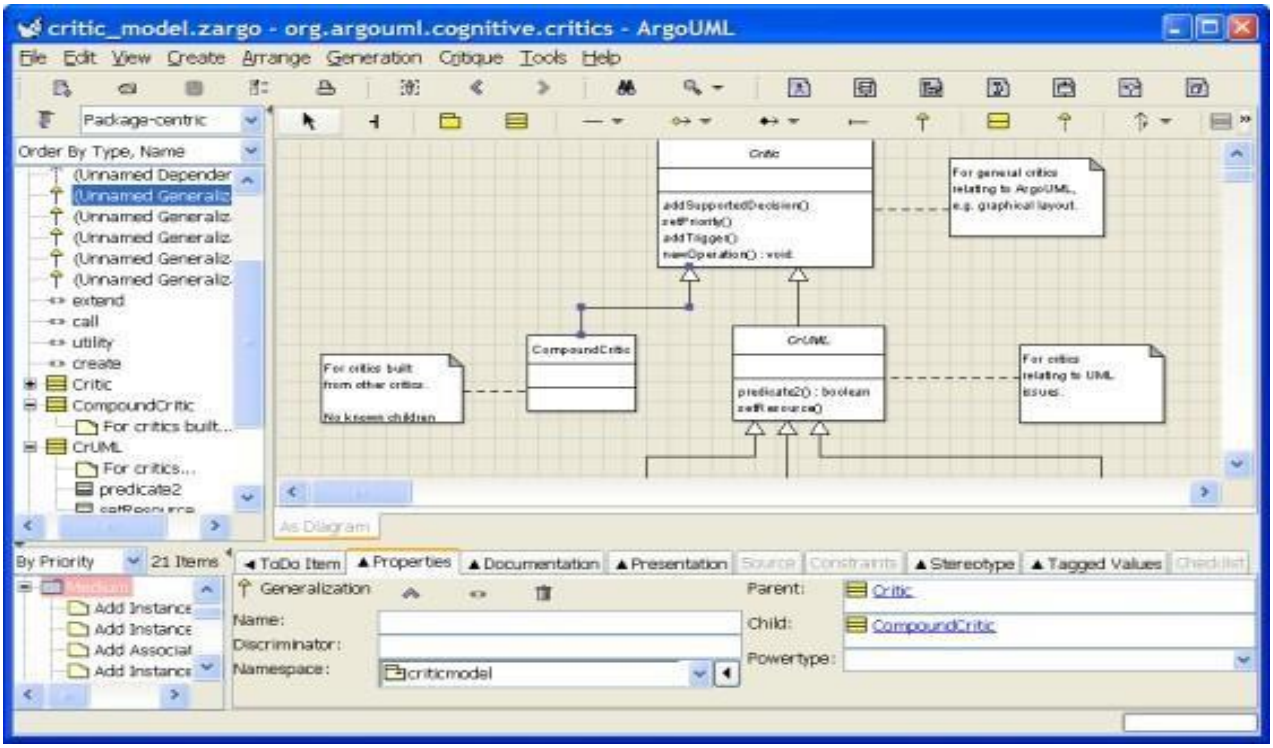
- Each model contains a single deployment diagram that shows the mapping of processes to hardware.

THE ArgoUML USER INTERFACE

Overview of the Window

The title bar of the window shows the following 4 parts of information, separated from each other by a dash.

- The current filename.
- The name of the currently active diagram.
- The name "ArgoUML".
- An asterisk (*). This item is only present if the current project file is "dirty", i.e. it is altered, but not yet saved. In other words, if the asterisk is absent, then the current file has not been altered.



The window comprises four sub-windows or panes namely,

- The Explorer pane,
- The Editing pane
- The Details Pane and
- The To-Do Pane.

THE EDITING PANE

The Toolbar

The toolbar at the top of the editing pane provides the main functions of the pane.

The Toolbar consists of the following menus:

- File operations
- Edit operations
- View operations
- Create operations

File operations

- New
- Open Project...
- Save Project
- Project Properties
- Print

Edit operations

- Remove From Diagram

gram ".

- Delete From Model
- Configure Perspectives
- Settings

View operations

- Find...
- Zoom

Create operations

- New Use Case Diagram
- New Class Diagram
- New Sequence Diagram
- New Collaboration Diagram
- New Statechart Diagram
- New Activity Diagram
- New Deployment Diagram

The tools fall into four categories.

- ***Layout tools.*** Provide assistance in laying out model elements on the diagram.
- ***Annotation tools.*** Used to annotate model elements on the diagram.
- ***Drawing tools.*** Used to add general graphic objects to diagrams.
- ***Diagram specific tools.*** Used to add UML model elements specific to a particular diagram type to the diagram.

- ***Layout Tools***

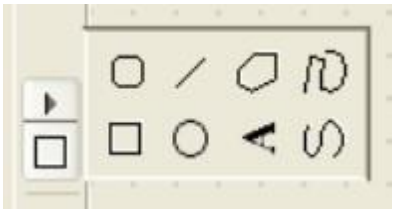
↑ **Select** - This tool provides for general selection of model elements on the diagram.

Broom Tool – This tool is used to sweep all model elements along.

- ***Annotation Tools***

Annotation tool - It is used to add a comment to a selected UML model element.

- ***Drawing Tools***



- Rectangle. Provides a rectangle.
- Rounded Rectangle. Provides a rectangle with rounded corners. There is no control over the degree of rounding.
- Circle. Provides a circle.
- Line. Provides a line.
- Text. Provides a text box. The text is entered by selecting the box and typing. Text is centered horizontally and after typing, the box will shrink to the size of the text. However it can be re-sized by dragging on the corners.
- Polygon. Provides a polygon. The points of the polygon are selected by button 1 click and the polygon closed with button 1 double click (which will link the final point to the first point).
- Spline. Provide an open spline. The control points of the spline are selected with button 1 and the last point selected with button 1 double click.
- Ink. Provide a polyline. The points are provided by button 1 motion.

USE CASE DIAGRAM SPECIFIC TOOLS

Actor - Add an actor to the diagram.

Use Case – Add a use case to the diagram.

Association - Add an association between two model elements selected using button 1 motion (from the first model element to the second).

The association tool selector

-  `Dependency`. Add a dependency between two model elements selected using button 1 motion (from the dependent model element).
-  `Generalization`. Add a generalization between two model elements selected using button 1 motion (from the child to the parent).
-  `Extend`. Add an extend relationship between two model elements selected using button 1 motion (from the extended to the extending use case).
-  `Include`. Add an include relationship between two model elements selected using button 1 motion (from the including to the included use case).
-  `Add Extension Point`. Add an extension point to a selected use case.

CLASS DIAGRAM SPECIFIC TOOLS

Class diagrams are used for only one of the UML static structure diagrams, the class diagram itself. Object diagrams are represented on the ArgoUML deployment diagram.

ArgoUML uses the class diagram to show model structure through the use of packages.

-  `Package`. Add a package to the diagram.
-  `Class`. Add a class to the diagram. For convenience, when the mouse is over a selected class it displays two handles to left and right which may be clicked or dragged to form association relationships (or composition in case SHIFT has been pressed) and two handles top and bottom which may be dragged or clicked to form generalization and specialization relationships respectively.
- `Association`. Add an association between two model elements selected using button 1 motion (from the first model element to the second). There are 2 types of association offered here, `bi-directional` or `unidirectional`.
-  `Aggregation`. Add an aggregation between two model elements selected using button 1 motion (from the first model element to the second). There are 2 types of aggregation offered here, `bi-directional` or `unidirectional`.
-  `Composition`. Add an composition between two model elements selected using button 1 motion (from the first model element to the second). There are 2 types of composition offered here, `bidirectional` or `unidirectional`.

- Association-end. Add another end to an already existing association using button 1 (from the association middle to a class, or vice versa). This is the way to create so-calld N-ary associations.
- Generalization. Add a generalization between two model elements selected using button 1 (from the child to the parent).
- Interface. Add an interface to the diagram. For convenience, when the mouse is over a selected interface it displays a handle at the bottom which may be dragged to form a realization relationship (the target being the realizing class).
- Realization. Add a realization between a class and an interface selected using button 1 motion (from the realizing class to the realized interface).
- Dependency. Add a dependency between two model elements selected using button 1 motion (from the dependent model element). There are also 2 special types of dependency offered here, Permission (P) and Usage (U). A Permission is created by default with stereotype Import, and is used to import elements from one package into another.
- Attribute. Add a new attribute to the currently selected class. The attribute is given the default name newAttr of type int and may be edited by button 1 double click and using the keyboard, or by selecting with button 1 click (after the class has been selected) and using the property tab.

SEQUENCE DIAGRAM SPECIFIC TOOLS

- ClassifierRole. Add a classifierrole to the diagram.
- a() Message with Call Action. Add a call message between two classifierroles selected using button 1 motion (from the originating classifierrole to the receiving classifierrole).
- a Message with Return Action. Add a return message between two classifierroles selected using button 1 motion (from the originating classifierrole to the receiving classifierrole).
- a Message with Create Action. Add a create message between two classifierroles selected using button 1 motion (from the originating classifierrole to the receiving classifierrole).
- a Message with Destroy Action. Add a destroy message between two classifierroles selected using button 1 motion (from the originating classifierrole to the receiving classifierrole).
-

 Add Vertical Space to Diagram. Add vertical space to a diagram by moving all messages below this down. Click the mouse at the point where you want the space to be added and drag down the screen vertically the distance which matches the height of the space you'd like to have added.

-  Remove Vertical Space in Diagram. Remove vertical space from diagram and move all elements below up vertically. Click and drag the mouse vertically over the space that you want deleted.

COLLABORATION DIAGRAM SPECIFIC TOOLS

-  Classifier Role. Add a classifier role to the diagram.
- Association Role. Add an association role between two classifier roles selected using button 1 motion (from the originating classifier role to the receiving classifier role). There are 6 types of association roles offered here, see Figure 12.4, " The association tool selector. ": association, aggregation and composition, and all these three can be bidirectional or unidirectional.
-  Generalization. Add a generalization between two model elements selected using button 1 (from the child to the parent).
-  Dependency. Add a dependency between two model elements selected using button 1 motion (from the dependent model element).
-  Add Message. Add a message to the selected association role.

STATE CHART DIAGRAM SPECIFIC TOOLS

-  Simple State. Add a simple state to the diagram.
-  Composite State. Add a composite state to the diagram.
-  Transition. Add a transition between two states selected using button 1 motion
-  Synch State. Add a synchstate to the diagram.
-  Submachine State. Add a submachinestate to the diagram.
-  Stub State. Add a stubstate to the diagram.
-  Initial. Add an initial pseudostate to the diagram.

- Shallow History. Add a shallow history pseudostate to the diagram.
- Deep History. Add a deep history pseudostate to the diagram.
- Call Event. Add a Call Event as trigger to a transition. There are 4 types of events offered here: Call Event, Change Event, Signal Event and Time Event.
- Guard. Add a guard to a transition.
- Call Action. Add a call action (i.e. the effect) to a transition.

ACTIVITY DIAGRAM SPECIFIC TOOLS

- Action State. Add an action state to the diagram.
- Transition. Add a transition between two action states selected using button 1 motion (from the originating action state to the receiving action state).
- Initial. Add an initial pseudostate to the diagram.
- Final State. Add a final state to the diagram.
- Junction. Add a junction (decision) pseudostate to the diagram.
- Fork. Add a fork pseudostate to the diagram.
- Join. Add a join pseudostate to the diagram.
- CallState. Add a callstate to the diagram. A call state is an action state that calls a single operation. Hence, the name of the operation being called is put in the symbol, along with the name of the classifier that hosts the operation in parentheses under it.
- ObjectFlowState. Add a objectflowstate to the diagram. An objectflowstate is an object that is input to or output from an action.

DEPLOYMENT DIAGRAM SPECIFIC TOOLS

-  `Node`. Add a node to the diagram. For convenience, when the mouse is over a selected node it displays four handles to left, right, top and bottom which may be dragged to form association relationships.
-  `Node Instance`. Add a node instance to the diagram. For convenience, when the mouse is over a selected node instance it displays four handles to left, right, top and bottom which may be dragged to form link relationships.

 `Component`. Add a component to the diagram. For convenience, when the mouse is over a selected component it displays four handles to left, right, top and bottom which may be dragged to form dependency relationships.

-  `Component Instance`. Add a component instance to the diagram. For convenience, when the mouse is over a selected component instance it displays four handles to left, right, top and bottom which may be dragged to form dependency relationships.
-  `Generalization`. Add a generalization between two model elements selected using button 1 (from the child to the parent).
-  `Realization`. Add a realization between a class and an interface selected using button 1 motion (from the realizing class to the realized interface).
-  `Dependency`. Add a dependency between two model elements selected using button 1 motion (from the dependent model element).
- `Association`. Add an association between two model elements

EX NO: 2 PROBLEM STATEMENT - QUIZ SYSTEM

Aim: To develop a problem statement for QUIZ system

PROBLEM STATEMENT

Online Quiz system is to be designed for the students of XYZ college of Technology. The questions should be of objective type with multiple options. The system should be able to handle errors such as 2 options for one answer. After answering the questions in the preliminary round, the system verifies whether the student has answered the minimum number of questions in each level. If not, a message is displayed to make the student answer the required number of questions within an allotted time. If the student qualifies the prelims , he/she can proceed to the finals in the same way. At the end of the quiz the users' score along with information whether he has been selected or not has to be displayed. The quiz system has two levels of questions.

LEVELS	TOTAL NO. OF QUESTIONS	MINIMUM NO. OF QUESTIONS TO BE ATTENDED	MARKS FOR EACH QUESTION	DURATION
LEVEL1- PRELIMINARY	30	25	1	15 Min
LEVEL2- FINALS	15	10	2	10 Min

EX NO: 3 **USE CASE MODELING**

Aim: To identify Use Cases and develop the Use Case model for QUIZ system

Description:

Guideline: To Find Use Cases

Use cases are defined to satisfy the goals of the primary actors. Hence, the basic procedure is:

Step 1: Choose the system boundary. Is it just a software application, the hardware and application as a unit, that plus a person using it, or an entire organization?

Step 2: Identify the primary actors those that have goals fulfilled through using services of the system.The following questions help identify others that may be missed:

Who starts and stops the system?	Who does system administration?
Who does user and security management?	Is "time" an actor because the system does something in response to a time event?
Is there a monitoring process that restarts the system if it fails?	Who evaluates system activity or performance?
How are software updates handled? Push or pull update?	Who evaluates logs? Are they remotely retrieved?
In addition to human primary actors, are there any	Who gets notified when there are

Who starts and stops the system?	Who does system administration?
external software or robotic systems that call upon services of the system?	errors or failures?

Step 3: Identify the goals for each primary actor.

For example, use this table to prepare actor goal list

Actor	Goal
Participant	**Play level 1**
	Play level 2
	Submit answers
...	**...**

Step 4: Define Use Cases

In general, define one use case for each user goal. Name the use case similar to the user goal. Start the name of use cases with a verb.

A common exception to one use case per goal is to collapse CRUD (create, retrieve, update, delete) separate goals into one CRUD use case, idiomatically called Manage <X>. For example, the goals "edit user," "delete user," and so forth are all satisfied by the Manage Users use case.

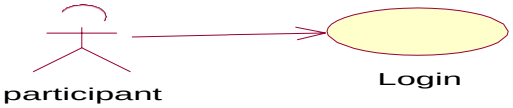
UML Notation:

Actor : An Actor, as mentioned, is a user of the system, and is depicted using a stick figure. The role of the user is written beneath the icon. Actors are not limited to humans. If a system communicates with another application, and expects input or delivers output, then that application can also be considered an actor.

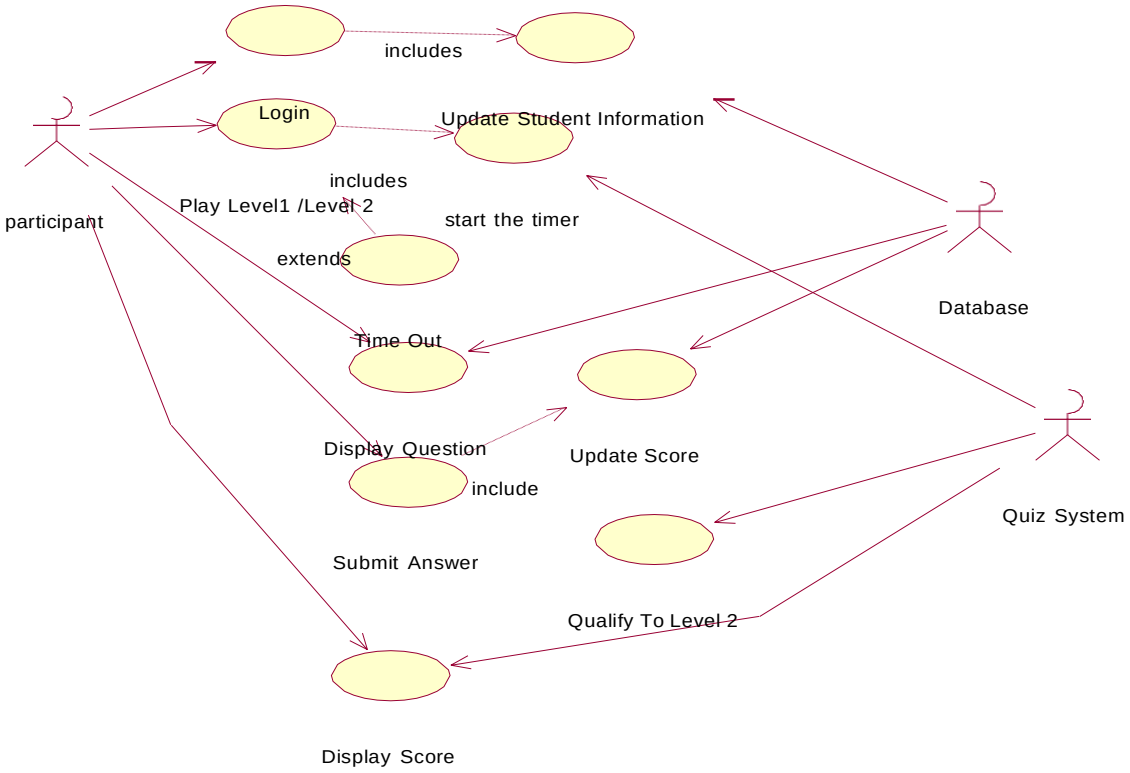
Use Case: A Use Case is functionality provided by the system, typically described as verb+object (e.g. Register Car, Delete User). Use Cases are depicted with an ellipse. The name of the use case is written within the ellipse.

Association: Associations are used to link Actors with Use Cases, and indicate that an Actor participates in the Use Case in some form. Associations are depicted by a line connecting the Actor and the Use Case.

Actor Association Use case



Use-case Diagram: Quiz System



1.1 Use cases

The system will consist of Login screen to authenticate the participants whose information is updated in the database. On starting the quiz , timer is started to maintain the timings. In Level 1 /Level 2 Questions with four options are displayed sequentially .User select the answer and move to the next question . Finally he/she selects the Submit answers which updates the marks and displays the score to the participants. If he is playing Level 1 and qualified for level 2 , then next level questions are displayed otherwise Not Qualified Message is displayed.

Functional Requirements

1.1.1. Use Case: Login

Brief Description

The use case describes how a Participant logs into the Quiz System

Use Case Section	Comment
Use Case Name	Login
Scope	Quiz System
Level	"user-goal"

Use Case Section	Comment
Primary Actor	Participant
Stakeholders and Interest list	- Participant: logs into the Quiz System ...
Preconditions	None
Success Guarantee/ Post condition	If the use case was successful, the actor is now logged into the system. If not, the system State is unchanged.
Main Success Scenario	1. The System requests the actor to enter his/her name and password 2. The actor enters his/her name and password 3. The System validates the entered name and password and logs the actor into the System
Extensions	3a.If the pwd is wrong, user is allowed for 3 attempts
Special Requirements	Timer
Technology and Data Variations List	-
Frequency of Occurrence	Could Be nearly Continuous
Miscellaneous	If the connection is terminated before the form is submitted, the fields are cleared and the Server is returned to the wait state. The Administrator can make the system not to get updated by others.

1.1.2. Use Case: Play Level 1 /Level 2 (provide the content same as above

1.1.3. Use Case: Display Questions (provide the content same as above)

1.1.4. Use Case: Submit Answers (provide the content same as above)

1.1.5. Use Case: Qualify to Level 2 (provide the content same as above)

1.1.6. Use Case: Display Score (provide the content same as above)

3.2. Non-functional requirements

Non-functional requirements are often called qualities of a system. Other terms for non-functional requirements are "constraints", "quality attributes", "quality goals", "quality of service requirements" and "non-behavioral requirements".

Non-functional requirements, can be divided into two main categories:

1) Execution qualities, such as security and usability, which are observable at run time.

2) Evolution qualities, such as testability, maintainability, extensibility and scalability, which are embodied in the static structure of the software system.

3.2.1. Functionality : Multiple users must be able to perform their work concurrently. If the participant has completed 30 Minutes allotted for him/her, he or she should be notified with the message "your time slot is over".

3.2.2. Usability: The desktop user-interface shall be Windows 95/98/2000/xp compliant.

3.2.3. Reliability: The System should function properly for allotted time slot and produces score card with no more than 10% down time.

3.2.4. Performance

1. The System shall support up to 100 simultaneous users against the central database at any given time and up to 100 simultaneous users against the local servers at any one time.
2. The System must be able to complete 80% of all transactions within 2 minutes.

3.2. 5. Supportability: None.

3.2. 6. Security

1. The System should secure so that only registered participants can take part in Quiz.

2. Once the participant had submitted a answer, he/she can't change the answer later.

3.2.7. Design Constraints: The system shall provide a window-based desktop interface

VIVA QUESTIONS:

1.What is an Actor?

2.Define Usecase?

3.What is Generalization?

4.Difference between include and extend?

5.What is object?

Ex No: 4 **UML Class Diagram**

Aim: To Identify the Conceptual Classes and to develop a domain model with UML class diagram for QUIZ system

Description

A class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, and the relationships between the classes.

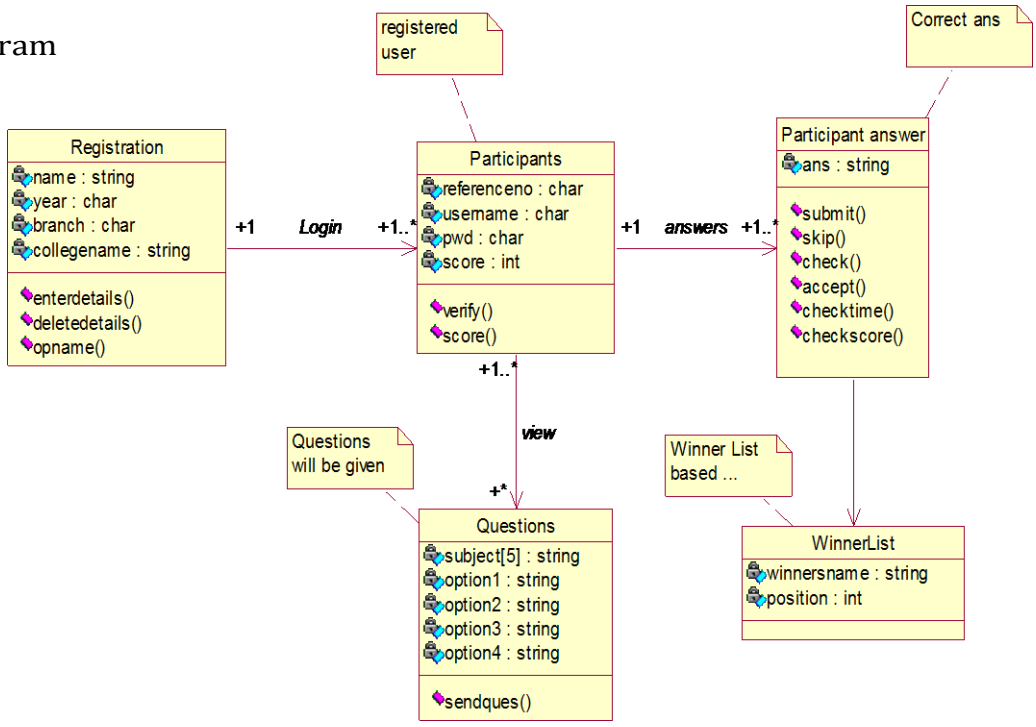
The class diagram is the main building block in object oriented modeling. They are being used both for general conceptual modeling of the systematic of the application, and for detailed modeling translating the models into programming code. The classes in a class diagram represent both the main objects and or interactions in the application and the objects to be programmed. In the class diagram these classes are represented with boxes which contain three parts

A class with three sections.

- The upper part holds the name of the class
- The middle part contains the attributes of the class
- The bottom part gives the methods or operations the class can take or undertake

In the system design of a system, a number of classes are identified and grouped together in a class diagram which helps to determine the statically relations between those objects. With detailed modeling, the classes of the conceptual design are often split in a number of subclasses.

Class Diagram



VIVA QUESTIONS:

1.What is four levels of visibility?
2.Difference between Aggregation and Composition?
3.What is Class and Object?
4.What is Association?
5.What is Inheritance?

Ex No: 5 UML Interaction Diagram

Aim: To Identify the interaction between objects and to develop using UML Interaction diagram for QUIZ system

Description

Sequence diagrams document the interactions between classes to achieve a result, such as a use case. Because UML is designed for object-oriented programming, these communications between classes are known as messages. The Sequence diagram lists objects horizontally, and time vertically, and models these messages over time.

Notation In a Sequence diagram, classes and actors are listed as columns, with vertical lifelines indicating the lifetime of the object over time.

Object :Objects are instances of classes, and are arranged horizontally. The pictorial representation for an Object is a class (a rectangle) with the name prefixed by the object name (optional) and a semi-colon. **: Object1**

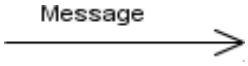
Actor: Actors can also communicate with objects, so they too can be listed as a column. An Actor is modeled using the ubiquitous symbol, the stick figure.

Lifeline: The Lifeline identifies the existence of the object over time. The notation for a Lifeline is a vertical dotted line extending from an object.

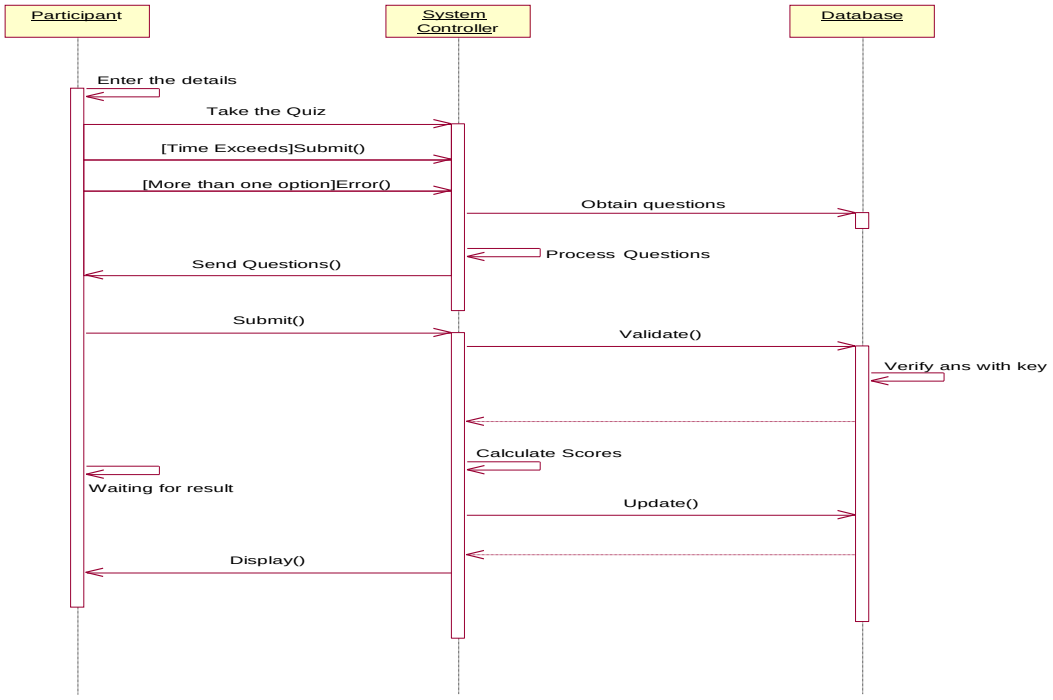
Activation: Activations, modeled as rectangular boxes on the lifeline, indicate when the object is performing an action.

Message

Messages, modeled as horizontal arrows between Activations, indicate the communications between objects.



Sequence diagram



Ex No : 6 UML State Chart Diagram and Activity Diagram

Aim: To develop using UML State Chart diagram and Activity diagrams for QUIZ system

State Chart Diagram Description

UML preserves the general form of the traditional state diagrams. The UML state diagrams are directed graphs in which nodes denote states and connectors denote state transitions. For example, Figure 1 shows a UML state diagram corresponding to the computer keyboard state machine. In UML, states are represented as rounded rectangles labeled with state names. The transitions, represented as arrows, are labeled with the triggering events followed optionally by the list of executed actions. The initial transition originates from the solid circle and specifies the default state when the system first begins. Every state diagram should have such a transition, which should not be labeled, since it is not triggered by an event. The initial transition can have associated actions.

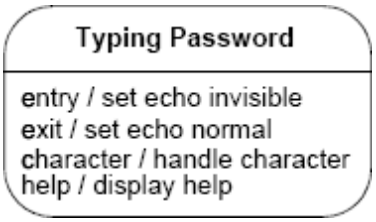
The main purposes of using Statechart diagrams:

- To model dynamic aspect of a system.
- To model life time of a reactive system.
- To describe different states of an object during its life time.
- Define a state machine to model states of an object

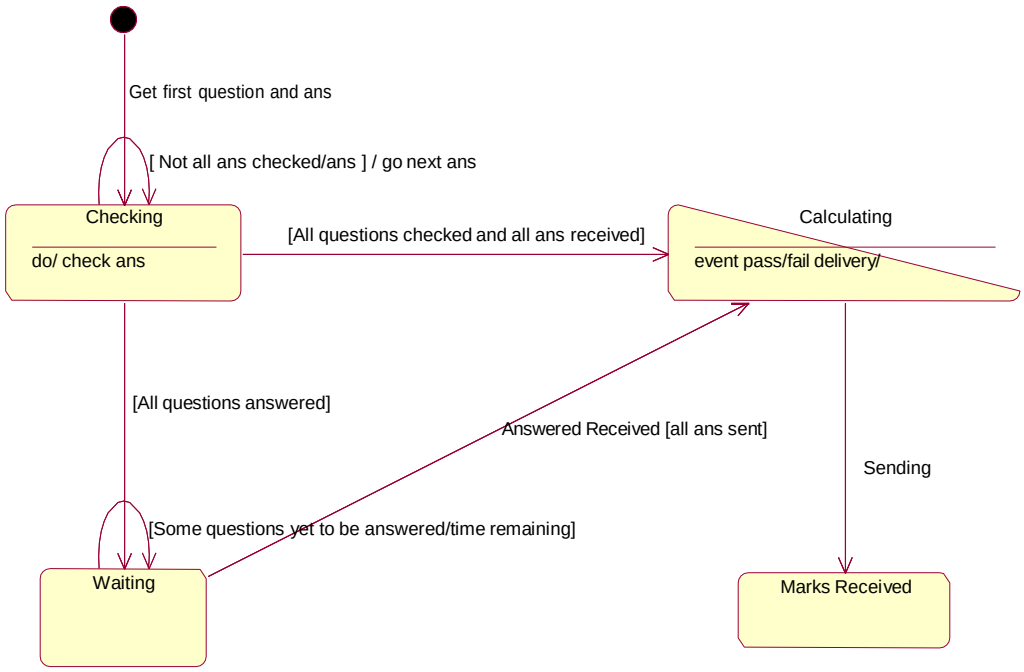
Steps to prepare state chart diagram

- Identify important objects to be analyzed.
- Identify the states.
- Identify the events.

State Notation:



State Chart Diagram



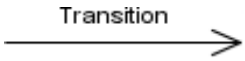
Activity Diagram Description

Activity diagrams are used to document workflows in a system, from the business level down to the operational level. Activity diagram is a variation of the state diagram where the "states" represent operations, and the transitions represent the activities that happen when the operation is complete. The general purpose of Activity diagrams is to focus on flows driven by internal processing vs. external events.

Activity States

Activity states mark an action by an object. The notations for these states are rounded rectangles, the same notation as found in State chart diagrams.

Transition: When an Activity State is completed, processing moves to another Activity State. Transitions are used to mark this movement. Transitions are modeled using arrows.



Swim lane: Swim lanes divide activities according to objects by arranging objects in column format and placing activities by that object within that column. Objects are listed at the top of the column, and vertical bars separate the columns to form the swim lanes.

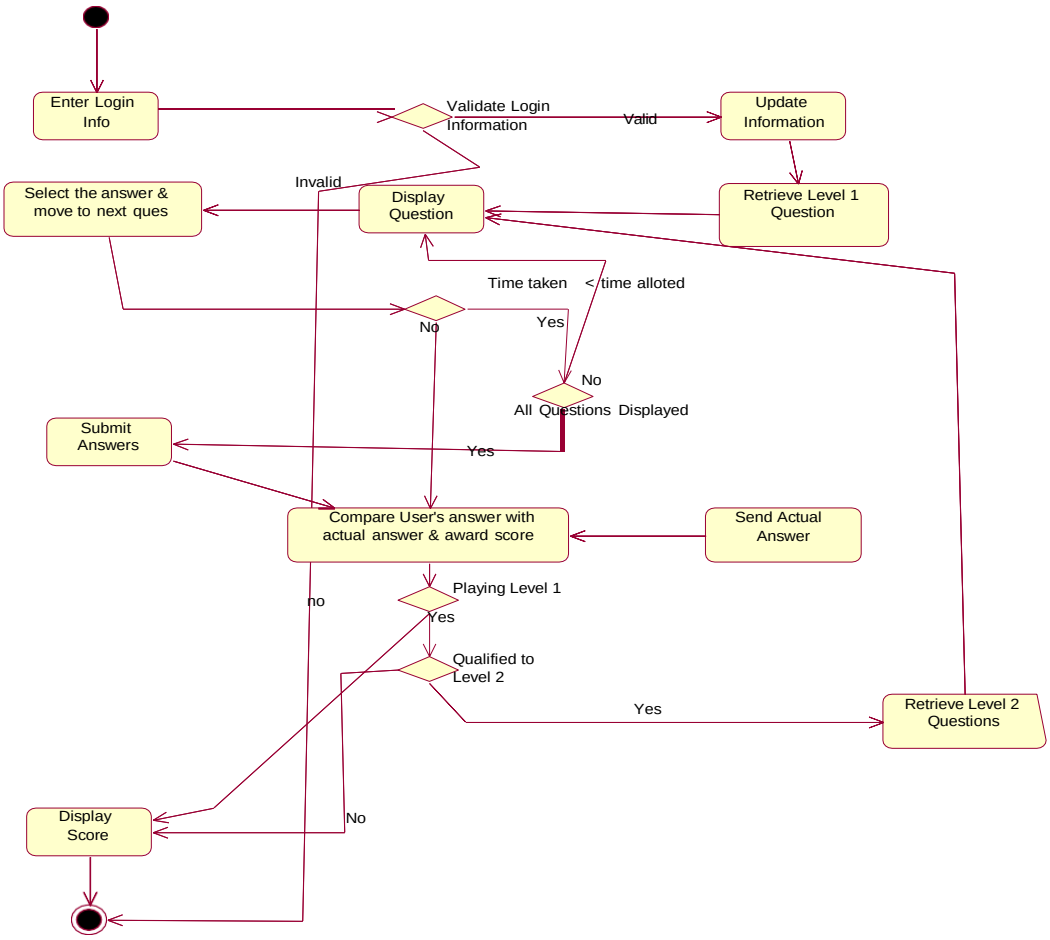
Initial State: The Initial State marks the entry point and the initial Activity State. The notation for the Initial State is the same as in State chart diagrams, a solid circle. There can only be one Initial State on a diagram. 

Final State: Final States mark the end of the modeled workflow. There can be multiple Final States, and these are modeled using a solid circle surrounded by another circle. 

Synchronization Bar: Activities often can be done in parallel. To split processing ("fork"), or to resume processing when multiple activities have been completed ("join"), Synchronization Bars are used. These are modeled as solid rectangles, with multiple transitions going in and/or out.

Activity diagram : Quiz System

Participant Quiz System Database



VIVA QUESTIONS:

1. What is Activity?
2. What is Fork and Join?
3. What is Event, State, Transitions?
4. Difference between Data flow Diagram and State Transition diagram?
5. Define Petrinet.

Ex No : 7 Logical Architecture with UML Package diagram

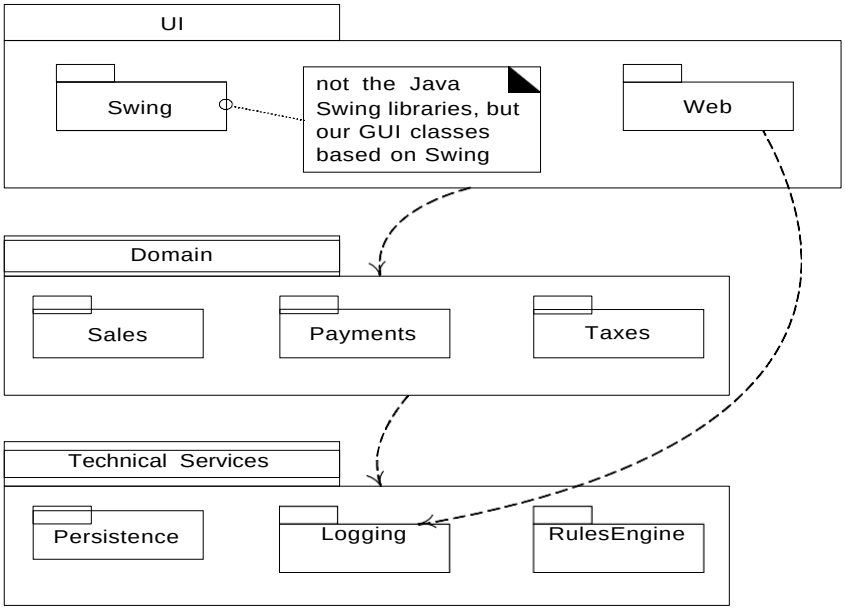
Aim: To draw the partial layered, logical architecture diagram with UML package diagram notation.

Logical Architecture and Layers

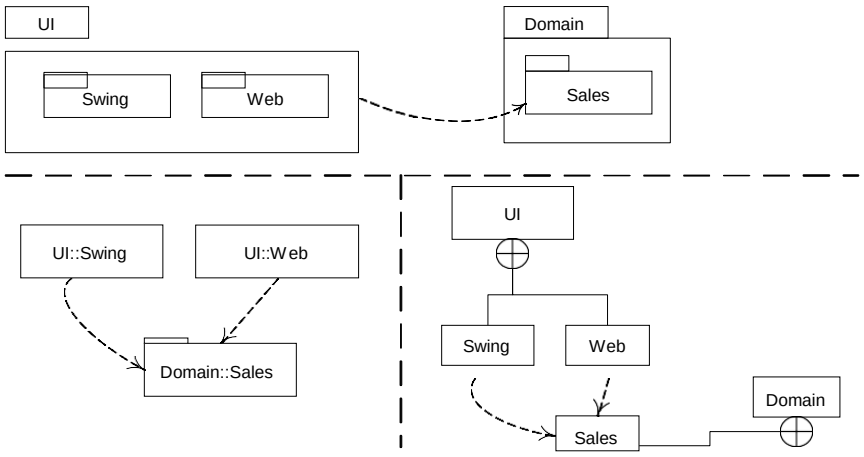
- Logical architecture: the large-scale organization of software classes into packages, subsystems, and layers.
  - "Logical" because no decisions about deployment are implied.
- Layer: a very coarse-grained grouping of classes, packages, or subsystems Typical layers in an OO system:
  - User Interface
  - Application Logic and Domain Objects
  - Technical Services
    - Application-independent, reusable across systems.
- Relationships between layers:

- Strict layered architecture: a layer only calls upon services of the layer directly below it.
- Relaxed layered architecture: a higher layer calls upon several lower layers.

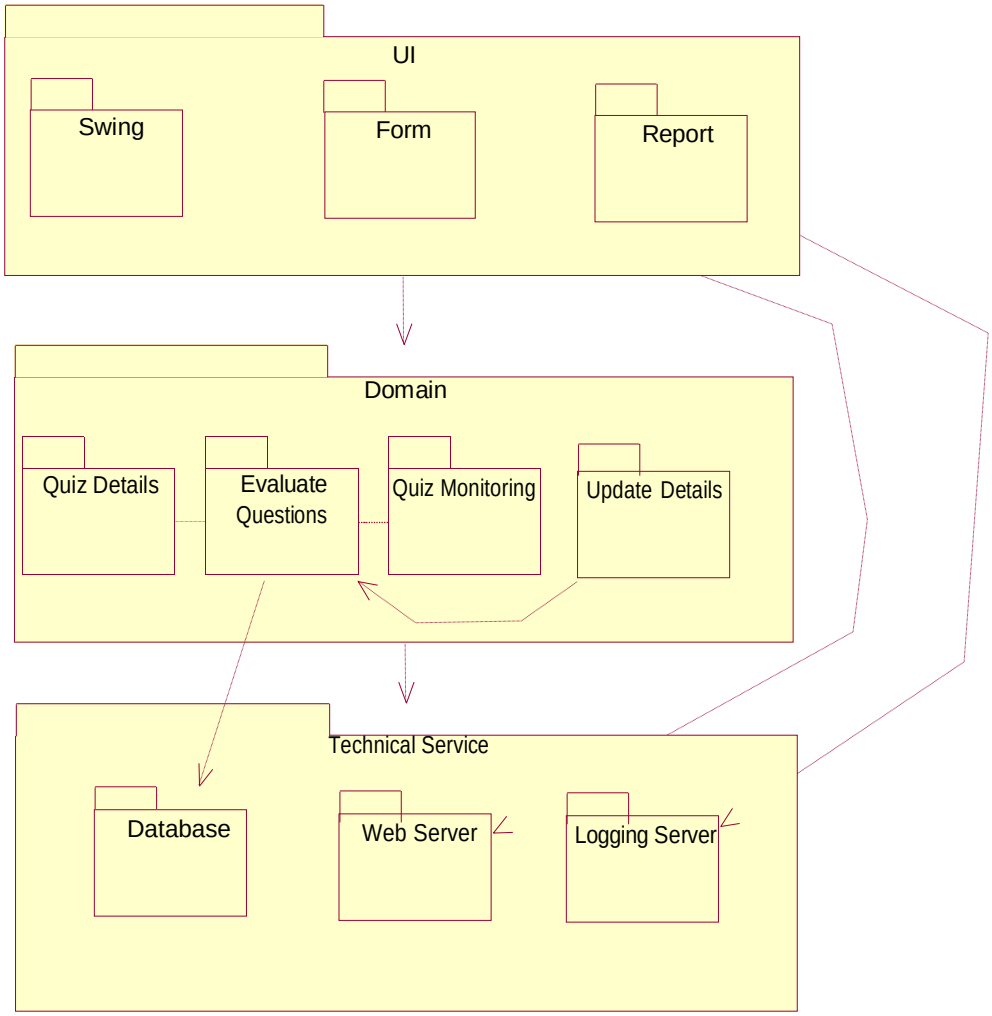
Layers shown with UML package diagram.



Various UML notations for package nesting



Package Diagram : Quiz System



Ex No: 8 Project Development & Testing: QUIZ SYSTEM

Aim: To Develop and test the Technical services layer, Domain objects layer and User interface layer.

Description:

In a quiz system we perform the following

1) Student registration
2) Quiz Participation
3) Result Processing

1) Student Registration

The inputs to the quiz system initially is the student details for the quiz which involves the following details like

1) name
2) year
3) branch

4) college name

When a student enters these details it will be updated in a database. After that the student will be allowed to participate.

2) Quiz Participation

When the participant begins to answer the questions, a clock is maintained to keep track of the time.

The correct answers for the questions are stored in a separate database .After answering the questions in the preliminary round, the system verifies whether the student has answered the minimum number of questions in each level. If not, a message is displayed to make the student answer the required number of questions within an allotted time.

If the student qualifies the prelims, he/she can proceed to the finals in the same way.

3) Result Processing

After all the students have participated in the quiz, the system processes the marks scored by all the students. The system sorts the marks and generates a final output, which displays the name and scores

SOFTWARE DEVELOPMENT AND TESTING

TECHNICAL SERVICES LAYER

Table Structure:

Student Registration

	Field Name	Data Type
▶	Name	Text
	Branch	Text
	College	Text

Level 1-Questions and Answers

	Field Name	Data Type
▶	qno	Number
	question	Text
	option1	Text
	option2	Text
	option3	Text
	option4	Text
	ans	Text
	rightans	Text

Level 2: Questions and answers

	Field Name	Data Type
▶	qno	Number
	question	Text
	option1	Text
	option2	Text
	option3	Text
	option4	Text
	ans	Text
	rightans	Text

UI LAYER AND DOMAIN LAYER

Testing UI Layer & Domain Layer

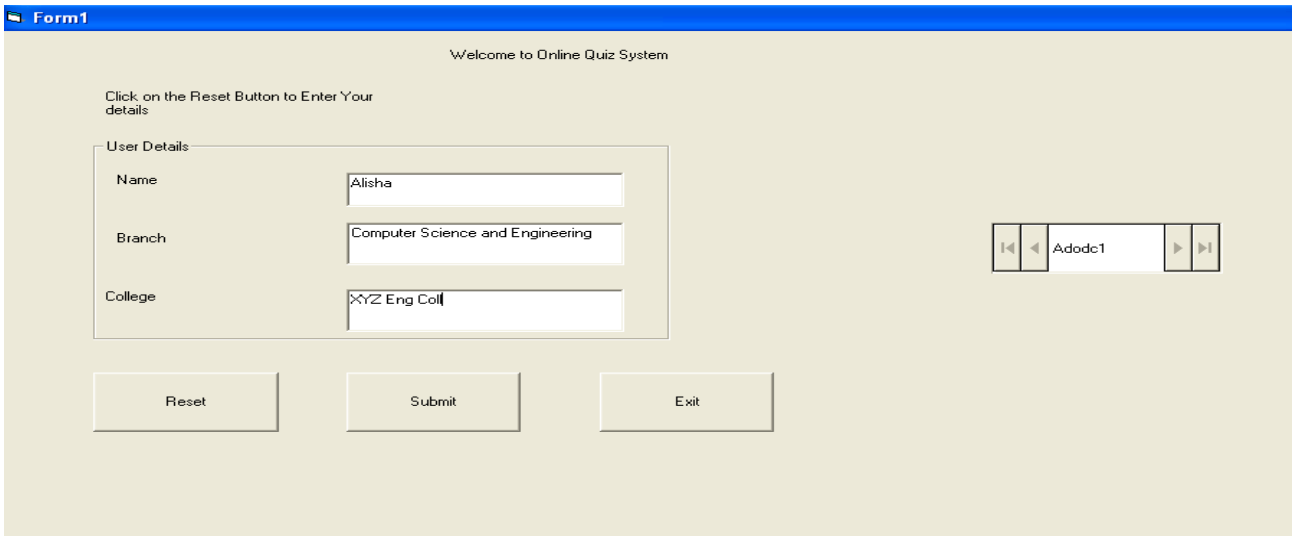
Test case Template

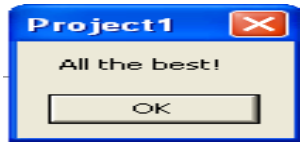
Test case Id : Screen001

Product **:** Quiz System

Module **:** Enter Student details

Form 1: Student Registration.





S.No	Action To Perform	Expected Result	Actual Result	Pass/Fail
1	Run the executable file	Enter Student details form appears	As expected	Pass
2	Click reset Button	Resets the text fields	-do-	Pass

3.	Enter “Alisha”, ”CSE”,”XYZ” in Name ,Branch and College Text Box	The Cursor moves to the submit button	-do-	Pass
4.	Click Submit Button	The Student details window is closed and goes to the level 1 of the quiz form	-do-	Pass
5.	Click Cancel Button	The Student details form is closed	-do-	Pass

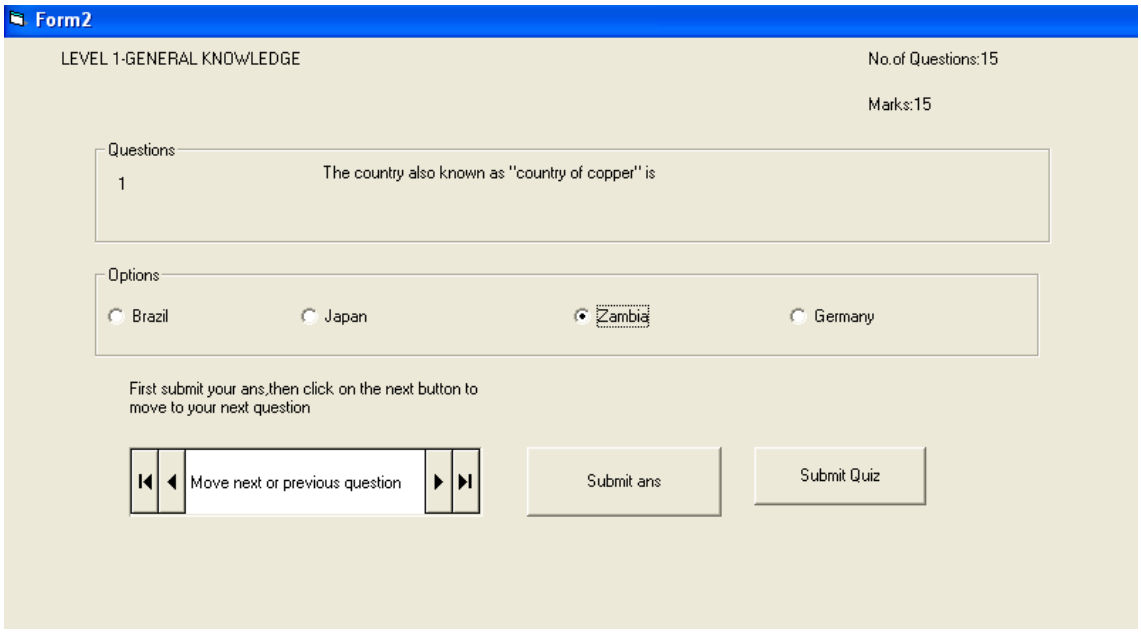
Test case Template

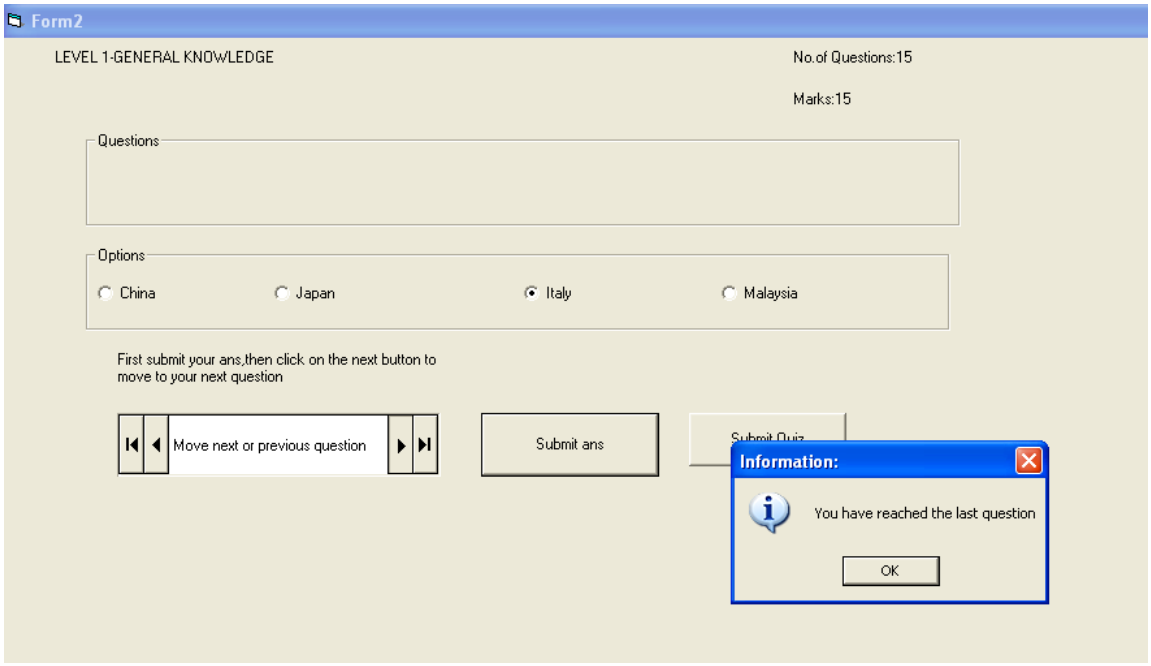
Test case Id : Screen002

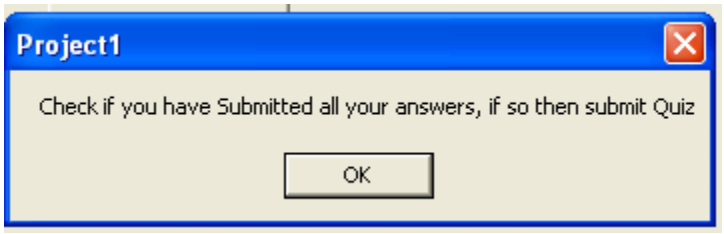
Product **:** Quiz System

Module **:** Level 1 of the Quiz

Form 2: Level 1







S.No	Action To Perform	Expected Result	Actual Result	Pass/Fail
1.	Select the suitable option for the required question.	1. The cursor moves to the Submit ans button. 2. The Cursor Moves to the next question using ADODC control	-do-	Pass
2.	Click submit ans Button	The ans is accepted and stored in the database.	-do-	Pass
3.	Click submit quiz Button	All the ans are accepted and the quiz is submitted. The Level 1 of the quiz form is closed.	-do-	Pass

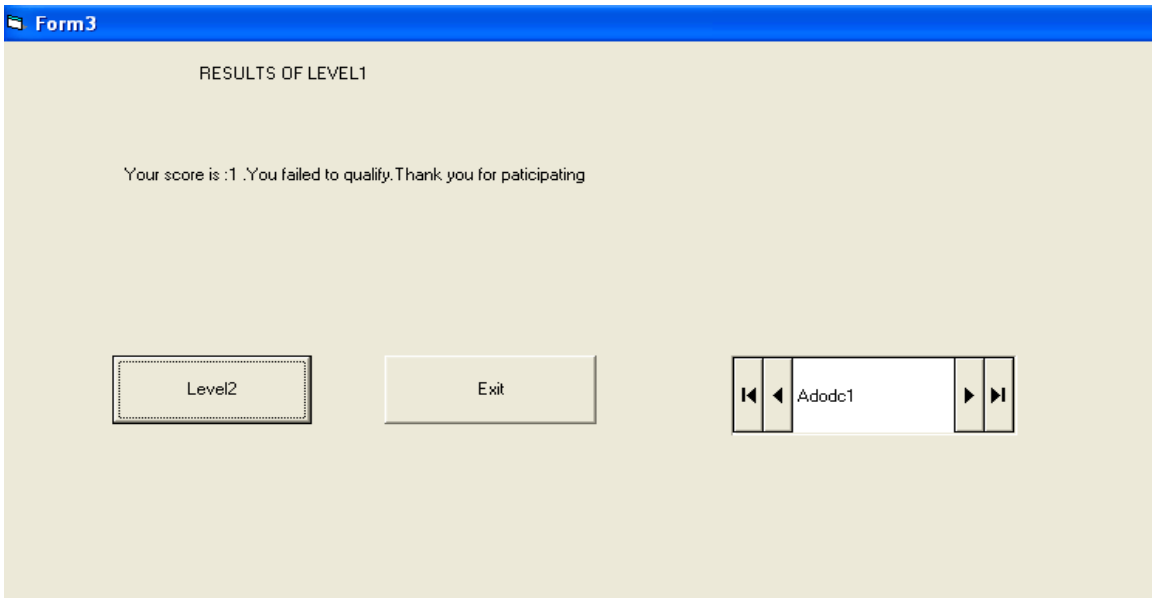
Test case Template

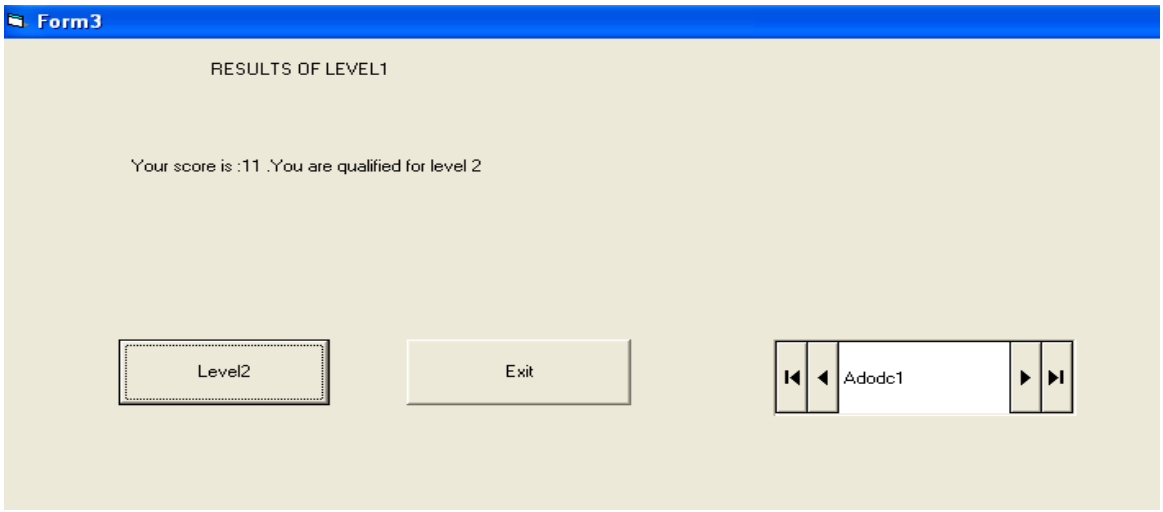
Test case Id : Screen003

Product : Quiz System

Module : Result Processing of Level 1 of the Quiz

Form 3: Result Processing of Level 1





S.No	Action To Perform	Expected Result	Actual Result	Pass/Fail
1.	Click the Level 2 Button	1. Error Message appears as "You can Exit, Click on the Exit Button if not qualified. 2. If qualified close result form and enter Level 2 Form of the quiz.	-do-	Pass
2.	Click Exit Button	The Result Processing Form is closed and the Level 2 Form appears	-do-	Pass

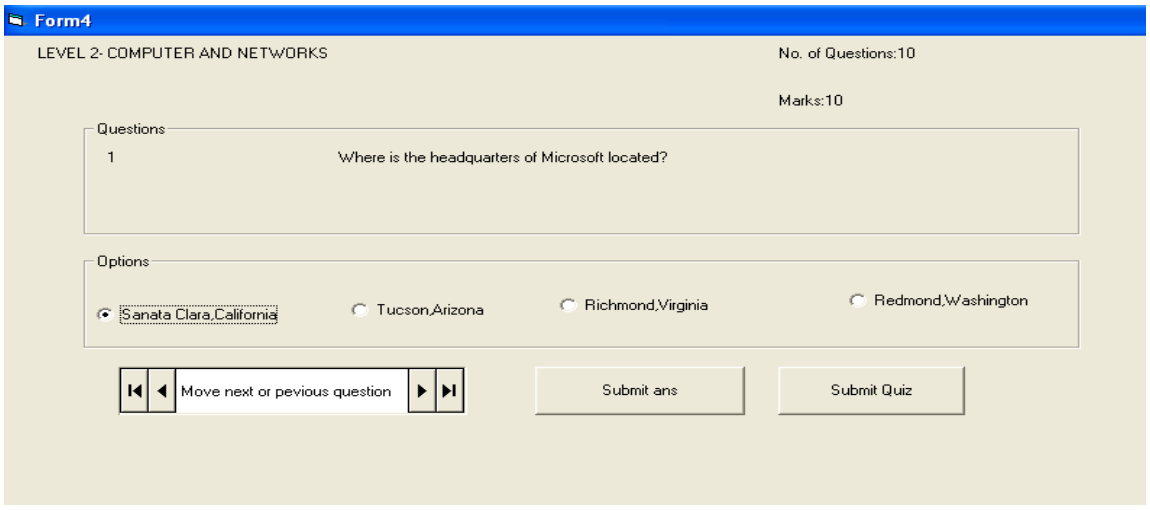
Test case Template

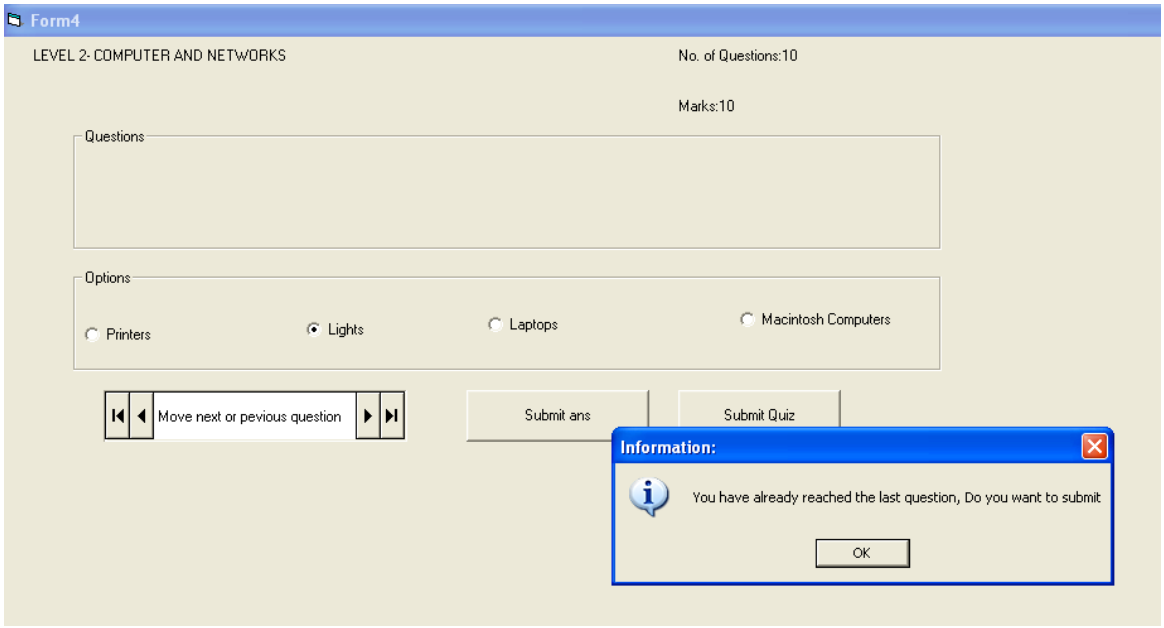
Test case Id : Screen004

Product : Quiz System

Module : Level 2 of the Quiz

Form 4: Level 2 of the quiz

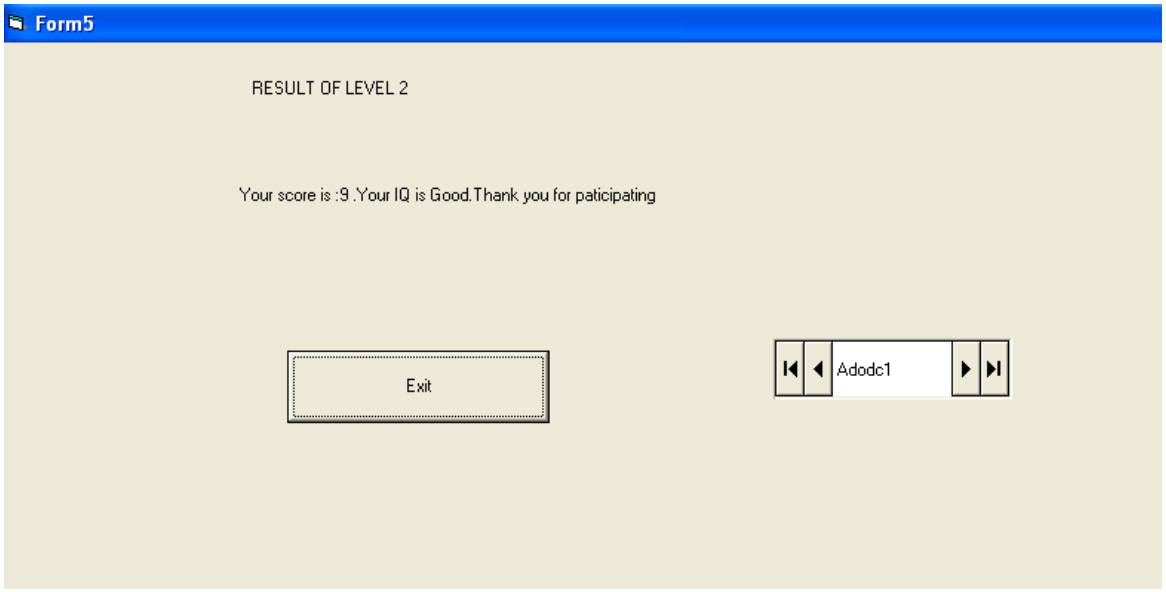




S.No	Action To Perform	Expected Result	Actual Result	Pass/Fail
1.	Select the suitable option for the required question.	1. The cursor moves to the Submit ans Button. 2. The cursor moves to the next question using ADODC control	-do-	Pass
2.	Click submit ans Button	The ans is accepted and stored in the database.	-do-	Pass
3.	Click submit quiz Button	The ans are accepted and the quiz is submitted The Level 2 of the quiz form is closed.	-do-	Pass

Test case Template
Test case Id : Screen005
Product **:** Quiz System
Module **:** Result Processing of Level 2 of the Quiz

Form 5: Result Processing of Level 2



S.No	Action To Perform	Expected Result	Actual Result	Pass/Fail
1.	Click Exit Button	The Result Processing Form is closed and a Message Window Appears as “You can exit”	-do-	Pass

VIVA QUESTIONS:

1.List the Levels of Testing?
2.What is test case?
3.Difference between Black box testing and White box testing?
4.What is Validation Testing?
5. What is big-bang approach?

PROBLEM STATEMENTS

1. PASSPORT AUTOMATION SYSTEM

Passport Automation System is used in the effective dispatch of passport to all of the applicants. This system adopts a comprehensive approach to minimize the manual work and schedule resources, time in a cogent manner. The core of the system is to get the online registration form (with details such as name, address etc.,) filled by the applicant. Testament is verified for its genuineness by the Passport Automation System with respect to the already existing information in the database. This forms the first and foremost step in the processing of passport application. The information is in turn forwarded to the regional administrator's (Ministry of External Affairs) office.

The application is then processed manually based on the report given by the system, and any forfeiting identified can make the applicant liable to penalty as per the law. The system also provides the applicant the list of available dates for appointment to 'document verification' in the administrator's office, from which they can select one. The system forwards the necessary details to the police for its separate verification whose report is then presented to the administrator.

2. BOOK BANK SYSTEM

The book bank management system is an software in which a member can register themselves and then he can borrow books from the book bank. It mainly concentrates on providing books for engineering students.

The Book-Bank system was established in XYZ University in 1990 with an initial book stock of 44,000. The stock has steadily increased and stands at over 221,696 books , distributed in all departments, schools and institutes of the University. Departmental Book-Banks are managed at departmental level.

The process of members registering and purchasing books from the book bank are described sequentially through following steps:

a. First the member registers himself if he was new to the book bank.
b. Old members will directly select old member button..
c. They select their corresponding year.
d. After selecting the year they fill the necessary details and select the book and he will be directed towards administrator
e. The administrator will verify the status and issue the book.

3. EXAM REGISTRATION SYSTEM

The main objective of Exam Registration System is to make applicants register themselves and apply for the exam. Exam Registration System provides

easy interface to all the users to apply for the exam easily. The Exam Registration is an application in which applicant can register themselves for the exam. The details of the students who have registered for the examination will be stored in a database and will be maintained. The registered details can then be verified for any fraudulent or duplication and can be removed if found so. The database which is verified can be used to issue hall tickets and other necessary materials to the eligible students.

The process of students accessing the registration application and applying for the examination by filling out the form with proper details and then the authorities verify those details given for truth and correctness are sequenced through steps

a. The students access exam registration application.
b. They fill out the form with correct and eligible details.
c. They complete the payment process.
d. The authorities verify or check the details.
e. After all verification the exam registration database is finalized.

The overview of the project is to design an exam registration tool for the registration process which makes the work easy for the applicant as well as the Authorities of Exam.

4. STOCK MAINTENANCE SYSTEM

Stock Maintenance System is a real time application used in the merchant's day to day system. This is a database to store the transaction that takes places between the Manufacturer, Dealer and the Shop Keeper that includes stock inward and stock outward with reference to the dealer.

Here we assume our self as the Dealer and proceed with the transaction as follows:

- The Manufacturer is the producer of the items and it contains the necessary information of the item such as price per item, Date of manufacture, best before use, Number of Item available and their Company Address.
- The Dealer is the secondary source of an Item and he purchases Item from the manufacturer by requesting the required Item with its corresponding Company Name and the Number of Items required.
- The Dealer is only responsible for distribution of the Item to the Retailers in the Town or City.
- The Shop Keeper or Retailer is the one who is prime source for selling items in the m a r k e t .
- The customers get item from the Shop Keeper and not directly from the Manufacturer or the Dealer.
- The Stock is the database used in our System which records all transactions that takes place between the Manufacturer and the Dealer and the Dealer and the Retailer.

The process of stock maintenance system is that the customer login to the particular site to place the order for the customer product. The stock maintenance system are described sequentially through steps

a. The customer login to the particular site.
b. They fill the customer details.
c. They place the orders for their product.
d. The vendor login and views the customer details and orders.

5. ONLINE COURSE RESERVATION SYSTEM

The aim is to design a course registration system for the admission of students in XYZ College of Technology for the academic year.All the processes involved in the course registration system are computerized.

It must contain all the details about:

- Student,
- Course offerings ,
- Registering a course ,
- Availability of courses ,
- Eligibility for Admission .

This registration system enables the admission of candidates into the institution on the basis of their educational and social background.This software will be accessible to the candidates who are short listed by the institution with their respective application numbers. The candidate has to register their desired branch from the available courses.

The registration must include the following details:

- Application No ,
- Student information ,
- Course selected,
- course Availability ,
- Eligibility

If a student wishes to change his/ her course, then the schedule is updated. To update a schedule, the student needs to be logged in. Then the student can select any other course from the available list of courses. And the schedule is submitted.

If a student wishes to cancel his/her course, the student's current schedule is retrieved and displayed first. Then the system prompts the user to confirm the deletion of schedule. If confirmed, the student is removed from the course offerings.

6. E- TICKETING SYSTEM

Online ticket reservation system has to be developed for Railway department.The System should provide information about Train Details (arrival and departure of trains along with information about stations through which it passes).Search about train passing through stations can be obtained either by means of train no, train name or specifying the source and destination stations.

While displaying information about train, it has to provide following information's

a) Stations through which train passes along with arrival and departure time.

b) Availability of seats in different classes along with waiting list and RAC (Reservation Against Cancellation)information.

While reserving ticket online the system obtain following information's from the user

a) Passenger name, Sex, Age, Address

b) Credit Card No, Bank Name

c) Class through passenger is going to travel i.e First class or Second class or AC

d) Train no and Train name, Date of Journey and number of tickets to be booked.

The ticket has to be issued based on availability of tickets in different classes along with the waiting list.The ticked issued should contain the following information's PNR NO, Train No, Date, No. of adults and children, Class, Coach, Seat/Berth, Sex, Age, Concession, Reservation fee, Total Cash, Train Name, Departure time. Before issuing ticket the amount from customer account has to be transferred to railway account. Cancellation of booked tickets should be available.

7. SOFTWARE PERSONNEL MANAGEMENT SYSTEM

This software is designed for the process of knowing the details of a person (i.e. Employee) works in a software company The employee management system is used to manage our personnel things such as maintaining databases in offices etc. this project is easy for the CEO to handle the details. This is personally used for CEO.

The CEO must enter the name and password to login the form and select the particular employee to view the details about that employee and maintaining the employee details personally. This process of employee management system are described sequentially through following steps,

- The CEO login to the employee management system.
- He/she search for the list of employees.
- Then select the particular employee.
- Then view the details of that employee.
- After displaying the employee details then logout.

8. CREDIT CARD PROCESSING

The Credit Card Processing System which is use to purchasing an item from any shop mall, and it is used to maintain the limitation of credit card balance and current transaction process could be update via credit card machine.

The customer should select the item to be purchased from the shop by using credit card payment then the vendor should give a bill for the selected item. The customer should give his card to swap and request for the kind of amount transaction. After processing the transaction, the CREDIT

CARD MACHINE should give the balance print statement or receipt. The Credit Card Processing System are described sequentially through steps

- Customer should select the item from the shop
- Vendor makes the bill for the selected item.
- Customer gives the credit card to the vendor to swap the card.
- They required amount transaction is done by the card reader.
- Vendor will issue the balance statement to the customer.
- Customers put the signature in the receipt and return to the vendor.

9. E-BOOK MANAGEMENT SYSTEM

E-book Management System gives an idea about how books are maintained in the particular websites. The books that are to be purchased, the books that are to be sold are maintained here. Further some additional details of the current books that is available in the store are also given. EBook Management System in this project is done in an authorized way. The password and user id has been set here. The website has to be maintained properly since the whole eBook purchase process can be improved. EBook management in this project gives the idea about how eBooks are maintained in a particular concern. The book details which includes the number of books available, no of pages and price. EBook management system the EBook management in this project is understood by going through the modules that is being involved.

The main objective of this project is to overcome the work load and time consumption which makes the maintenance of the eBook in an organization as a tedious process. This project provides complete information about the details of the eBook to the customers. This project identifies the amount of book available. Separate modules have been created for purchasing, viewing book details, and delivery details.

10. RECRUITMENT SYSTEM

The recruitment system allows the job seekers to enroll their names through the process of registration. The employee also can get the list of available candidates and shortlist for their company requirement. Once the applicant enrolls he receives an id, which helps him in further correspondence. A fees amount is received from the job seekers for enrollment. This system makes the task of the job seeker easier rather than waiting in queue for enrollment. This also reduces the time consumption for both for the job seeker and employee. The recruitment system will select the candidate for an organization based on aptitude test, Interview. It generates results for the test taken up the candidates and reports to view the systems usage by the graduates and the employers in the recruitment process in a periodical base. Access rights are allocated in following order.

Administrator->Organization/Company->Candidates

- The administrator is given rights only to add or delete Candidate profiles, provide information about company and job conformation and publish results for aptitude test.But has no rights to access the recruiting information.
- Organization or HR Department is given rights to access the recruiting information and if needed, make changes to them.
- They can also access the report database to view number of applied and selected graduates for all the designations with the details in brief.
- All the reports and queries are at their disposal. Candidates are the end users of the system.
- They login the recruiting session and attend the aptitude test.
- The candidates cannot access any of the databases including the questions, report or the profile database.
- The candidates can view their results through this system. Each Candidate is given a username and password to ensure the security.
- The aptitude test in recruitment system is basically objective questions.
- The session is fixed for each candidate and the questions carry a time limit within which the candidates are supposed to answer the questions.
- Otherwise the question lapses and no points are awarded for that question. The student has the ability to pass the question and answer the question later within the remaining time left.
- In recruitment system All the Graduates, Employer and Administrator information is maintained in normalized database instead of manual records.

11. FOREIGN TRADING SYSTEM

Foreign Trading System involves details about the export and import of materials between two or many countries.

The main objective of this project is to ensure globalization and thereby ensuring that the buyers all over the world get the desired product that helps in the development of their globalizations.

This software helps the trading system to be easy for the promoters and the buyers to maintain their export and import details.

The database contains all the details of the export and import procedures.

The system consists of two databases to maintain this software.

- The import session
- The export session.

Export session :

- ✓ The promoter has to maintain the details of the products that are been exported to other countries.
- ✓ The bank's certification should be issued for the export products.

The following details are maintained in database are

- ❖ product name that is to be delivered,
- ❖ the number of items to be exported ,
- ❖ the date and time of the product to be delivered,
- ❖ the mode of transport through which the goods are sent
- ❖ the details of the transaction.

Import session:

- ✓ The buyer has to maintain the details of the products brought from the foreign countries.
- ✓ The bank's certification should be issued to the import products.

The following details are maintained in database are

- ❖ product name that is to be imported,
- ❖ the number of items to be imported ,
- ❖ the date and time of the product to be imported ,
- ❖ the mode of transport through which the goods are received
- ❖ the details of the transaction.

12. CONFERENCE MANAGEMENT SYSTEM

Our conference Management System handles

- ➢ the creation of the registration form/s,
- ➢ collection of online payments by credit card,
- ➢ invoicing,
- ➢ bookkeeping,
- ➢ email communications.

- Organizers can configure the submission form to include any number of items.
- Extended abstracts and full-papers can be uploaded in any specified format.
- Organizers can configure up to nine numerical criteria for the evaluation of submissions by the reviewers.
- Textual appraisals and feedback for the submitters can also be collected. Reviewers can perform the reviews online from their account.
- Organizers can track their progress and send them reminders by email. Abstract selection can be performed automatically or manually.
- The automatic option selects submissions on the value of the marks given by the reviewers and can be followed by a manual selection.
- Accepted abstracts can be assigned to sessions and ordered.
- Participants can be selected independently of the abstract selection process. Alternatively, their selection can be synchronized with the selection of abstracts.
- This system sends automatic emails, bulk emails and email alerts to organizers.
- Filters are supplied to allow the targeting of groups of users when sending bulk emails.
- Prefilled templates are supplied for the bulk and automatic mails.
- The registrations, payments, submissions, reviews and results of the selection process can be exported to CSV or Excel for further processing.
- Submitted abstracts can be styled and exported to HTML or PDF. Uploaded files, as well as the PDF files generated from the submissions can be downloaded from the FTP server.

13. BPO MANAGEMENT SYSTEM

A call center is a centralized office used for the purpose of receiving and transmitting a large volume of request by telephone. A BPO is operated by a company to administer product support or information inquiries from customers. Outgoing calls for telemarketing, client and dept collection are also

made. A BPO is often operated through an extensive open workspace for call center agents, with work stations that include a computer for each agent, a telephone set/headset connected to a telecom switch and one or more supervisor stations. It can be independently operated or networked with additional centers, often linked to a corporate computer network, including mainframes, micro computers and LANS.Increasingly the voice and data pathways into the centre are linked through a set of new technologies called Computer Telephony Integration. Most major business use call centers to interact with their customers. Examples include utility companies, mail order catalogue firms, and customer support for computer hardware and software. Some business even service internal functions through call centers.

14. LIBRARY MANAGEMENT SYSTEM

The purpose of the LIBRARY MANAGEMENT SYSTEM is mainly to provide automation to the library. The categories of users provided are:

LIBRARIAN: He can read or write the information about any member and can update, delete or create a membership plan.

MEMBER: He can get a book issued.

The three major components in the application are:

-Login
-Issue/Borrow Book
-Balance Dues

The Library Management System implements databases to make the existing system more efficient. It is difficult to catch defaulters in a usual library system, but Library Management System solves this problem by providing messages to the administrator about the fine to be paid and books to be returned. When a book is to be borrowed, its barcode is read and is fed to Library Management System .When the administrator tries to issue a book to a member, Library Management System checks whether the member is allowed to borrow books respective to his limit. Library Management System also checks whether the book to be issued is a reference book or not. If the book is a reference book, it cannot be issued. If the book has no restrictions it is issued to the member, and also the member's details is updated with the books that he has just borrowed. Any member can search a book with its title name, or author's name.

15. STUDENT MANAGEMENT SYSTEM

The Student Management System can handle all the details about a Student. The details include College details, Course details, Students personal details, Academic details etc., The Student Management System is an automated version of manual Student Management System. The student management system allows authorized members to access the record of academically registered students. This helps in maintaining the database of the students in any educational organization. We can easily access any student's information anytime and can be kept safely for long period of time without any damage.

1. Administrator

2. User

i) Administrator:

SMS is managed by Administrator. Administrator has to update and monitor the registered student details , provide register number for all students, generate hall tickets for all students, evaluate examination marks and allocate fees for each course etc.,

ii) User:

User can only view their personal details, fees details, examination schedule, mark sheet, download hall tickets for each semester.

- To update the information that is essential to transmit to the users.
- To have a centralized control over the records of the students, departments, teachers, staffs, library, etc., and monitor the changes in these records.

CONTENT BEYOND SYLLABUS

UML Component & Deployment Diagram with Code Generation

Component Diagram Description:

Component diagrams fall under the category of an implementation diagram, a kind of diagram that models the implementation and deployment of the system. A Component Diagram, in particular, is used to describe the dependencies between various software components such as the dependency between executable files and source files.

The following artifacts are to be identified clearly:

- Files used in the system.
- Libraries and other artifacts relevant to the application.
- Relationships among the artifacts.

Component Diagram Notation

Component: A component represents a software entity in a system. Examples include source code files, programs, documents, and resource files. A component is represented using a rectangular box, with two rectangles protruding from the left side, as seen in the image to the right.

Dependency: A Dependency is used to model the relationship between two components. The notation for a dependency relationship is a dotted arrow, pointing from a component to the component it depends on.

Component Diagram for Quiz System

Deployment Diagram Description:

A deployment diagram in the Unified Modeling Language models the physical deployment of artifacts on nodes. To describe a web site, for example, a deployment diagram would show what hardware components ("nodes") exist (e.g., a web server, an application server, and a database server), what software components ("artifacts") run on each node (e.g., web application, database), and how the different pieces are connected (e.g. JDBC, REST, RMI).

The nodes appear as boxes, and the artifacts allocated to each node appear as rectangles within the boxes. Nodes may have subnodes, which appear as nested boxes. A single node in a deployment diagram may conceptually represent multiple physical nodes, such as a cluster of database servers.

The following artifacts should be identified:

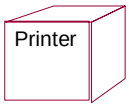
- Nodes
- Relationships among nodes

The usage of deployment diagrams can be described as follows:

- To model the hardware topology of a system.
- To model embedded system.
- To model hardware details for a client/server system.
- To model hardware details of a distributed application.
- Forward and reverse engineering.

Deployment Diagram Notation

Node: The element that provides the execution environment for the components of a system. Depicted by a cube with the name of the object in it, preceded by a colon, and underlined.

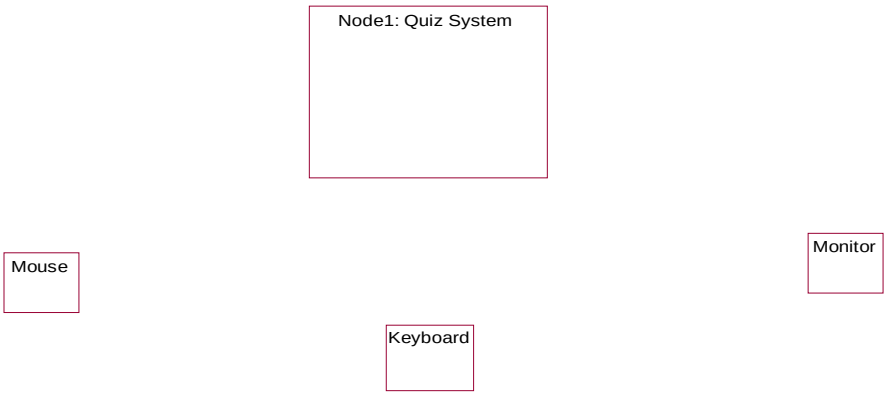


Node 1

Device :

Connection: Similar to the relation/association used in class diagrams to define the interconnection between nodes.interconnection between nodes.

Deployment Diagram for Quiz System



Code Generation

Steps:

1. Select the platform (Language) in the Component diagram 's Open Specification window
2. Select Realizes tab and assign classes to the component
3. Generate the code by selecting UpdateCode option of the Component.
4. A new window containing the newly generated code is opened.
5. Design the User Interface (forms)

CASE STUDY

1) ATM SYSTEM

The ATM will service one customer at a time. A customer will be required to insert an ATM card and enter a personal identification number (PIN) - both of which will be sent to the bank for validation as part of each transaction. The ATM will communicate each transaction to the bank and obtain verification that it was allowed by the bank. If the bank determines that the customer's PIN is invalid, the customer will be required to re-enter the PIN before a transaction can proceed. If the customer is unable to successfully enter the PIN after three tries, the card will be permanently retained by the machine, and the customer will have to contact the bank to get it back. The system shows the list of items such as balance enquiry, withdrawal, cancel options. When the customer selects the balance enquiry option, then the system shows the balance that left in the account and prints as receipt. When the customer selects the withdrawal option then the system should ask the amount and dispense the amount after enquiring the balance. After all transactions, the customer should collect the ejected card.

Entries will be made in the log when the ATM is started up and shut down, for each message sent to the Bank (along with the response back, if one is expected), for the dispensing of cash, and for the receiving of an envelope.

2) STUDENT MARK ANALYZING SYSTEM

Student marks analyzing system has to be developed for analyzing the marks obtained by the students who scored in Semester Examination.

The System should provide the following functionalities

- The System obtains the following information (Student No, Name, Department, Semester, Section, Marks obtained in each subject and No of students in each section) from the faculty and generates report.
- The total and average of each student should be calculated and ranked based on total and pass in all the subject appeared. If a student fails in at least in one subject then he/she should be given rank.
- The Final report should display the top 3 persons in each subject along with rank, percentage, Class, Pass/Fail Status for each student.
- The report should also contain information about no of students passed, failed, list of students who got more than 60% in each subject and overall list of students who got >=60%

3) TIC-TAC-TOE SYSTEM

Tic-tac-toe system is to play game in system. It should ensure that the game should accommodate only two players at a time. The player should be able to click on the labels and view their corresponding symbols. The software should be able to declare the winner when the required pattern of a particular symbol is found.

The players must first register themselves to access the system. While registering the players must give their details. Only two players can accommodate the game. The players are given alternate chances for clicking the boxes to place their corresponding labels in the tic-tac-toe square board. The software should be able to create a new play area if the players opt to play a new game. The software should be able to update the scores of individual players on the completion of each game. When the users opt to exit the software should log them out.

Login:

- The players must enter their names.
- It is the first step in this tic-tac-toe system.
- It should not exceed more than two.

Selecting the Icons:

- Two icons 'x' and 'o' is offered by the tic-tac-toe system.
- Two of them select either 'x' or 'o'
- But same icon should be selected by both players.
- These labels are placed in the game board by clicking the particular box in that game board, so that the corresponding icon or label is placed in that box

Clicking the Labels:

- The player should click the corresponding labels alternatively
- The system checks the winning condition and also he/she has chances to click the box still.
- If the required pattern for winning condition was found then display the winner name.
- If not, it allows the player to click the label and continue the game.
- Finally, the tic-tac-toe system displays the player name who won.

4) EXPERT SYSTEMS

An expert System has to be developed for prescribing medicines for given symptoms.

The System should contain the following features

- The user has to query the system by giving his name, age group, sex, symptoms, period (how long the user is having those symptoms). Based on these inputs, the system has to make a comparison with the symptoms in the database prescribing the medicine for the symptoms.
- Only the Registered users can query the system. So there should be registration form and after registration the user has to be given username and password. If the user enters the correct password then the system has to redirect to query form.
- In the query form, age group should be combo box and list of symptoms should be in the form of list box with options selecting multiple symptoms. While prescribing the medicine for the symptoms, corresponding disease, medicine name and medicine manufacturer name along with price has to be displayed.

5) PAYROLL SYSTEM

The payroll system is a new system that replaces the current manual process of calculating the salary. It enables to maintain employees details , the accountant calculates salary , generates pay slip and updates the database. Maintaining Employee Details (New Employee , Modification in Employee Details) The employee can login and to edit/view his profile .The system Admin / accountant maintains employee salary details based on his/her attendance and the payment type selected by the employee, Maintaining the Loan Details ,Payslip generation includes Company Details with Name , Department , Designation , Bank AccountNo ,Date , Basic pay, DA (30% of basic pay), HRA(8% of basicpay), Total Deductions(PF, IT, Loan , LOP) , Grosspay (total pay) , Netpay (Grosspay – Deduction) ,Access to database can be restricted to by providing multiple users and privileges