**6.SYSTEM DESIGN**

**6.1 Work flow:**

Below architecture diagram represents mainly flow of requests from users to database through servers. In this scenario overall system is designed in three tires separately using three layers called presentation layer, business logic layer and data link layer. This project was developed using 3-tier architecture.

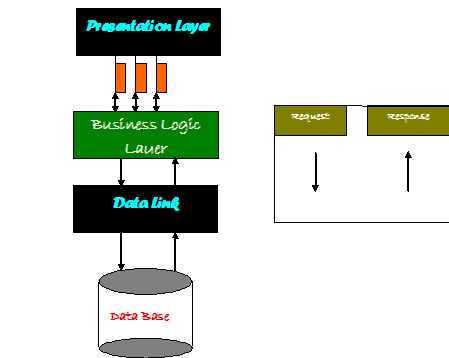
****

Figure 6.1: Architecture flow

**Introduction of UML**

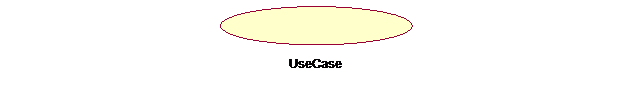
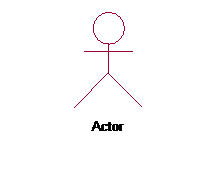
The Unified Modelling Language (UML) is a standard language for specifying, visualizing, constructing and documenting the artefacts’ of software systems, as well as for business modelling and other non-software systems. The UML represents a collection of best engineering practices that have proven successful in the modelling of large and complex systems. The UML is a very important part of developing objects oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects. Using the UML helps project teams communicate, explore potential designs, and validate the architectural design of the software.

**UML Diagrams:**

UML diagrams are designed to let developers and customers view a software system from a different perspective and in varying degrees of abstraction. UML diagrams commonly created using visual modelling tools.

**6.2Use case Diagram:**

A use case is a set of scenarios that describing an interaction between a user and a system. A use case diagram displays the relationship among actors and use cases. The two main components of a use case diagram are use cases and actors.



An actor is represents a user or another system that will interact with the system you are modelling. A use case is an external view of the system that represents some action the user might perform in order to complete a task.

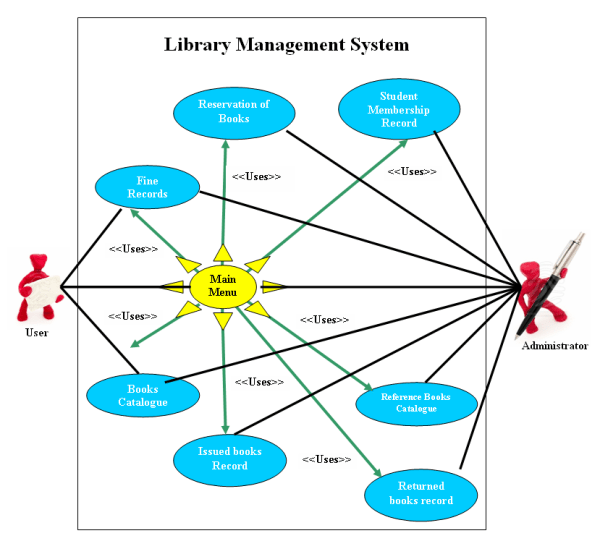
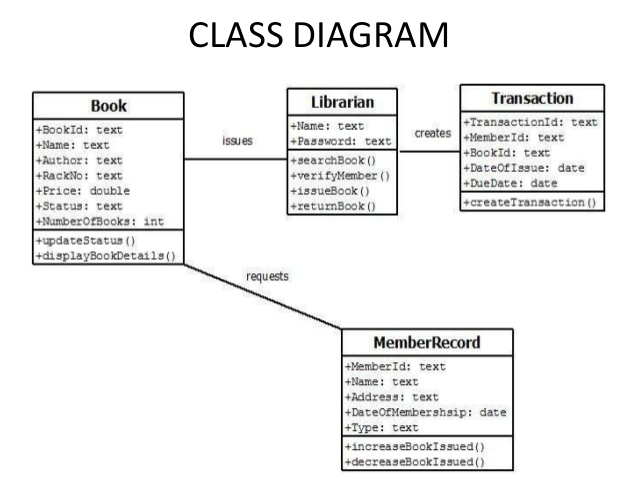


Figure:6.2 Use case diagram

**6.3 Class Diagram**

Class diagrams are widely used to describe the types of objects in a system and their relationships. Class diagrams model class structure and contents using design elements such as classes, packages and objects. Class diagrams describe three different perspectives when designing a system, conceptual, specification and implementation. These perspectives become evident as the diagram is created and help solidify the design. This example is only meant as an introduction to the UML and class diagrams. If you would like to learn more see the resources page for more detailed resources on UML.



6.3: Class diagram

**6.4 Sequence Diagram**

Sequence Diagrams demonstrate the behaviour of objects in a use case by describing the objects and the messages they pass. The diagrams are read left to right and descending. The example below shows an object of class 1 start the behaviour by sending a message to an object of class 2. Messages pass between the different objects until the object of class 1 receives the final message.

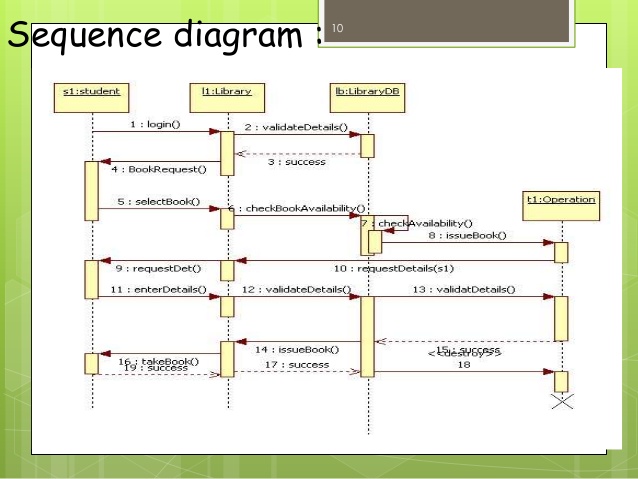
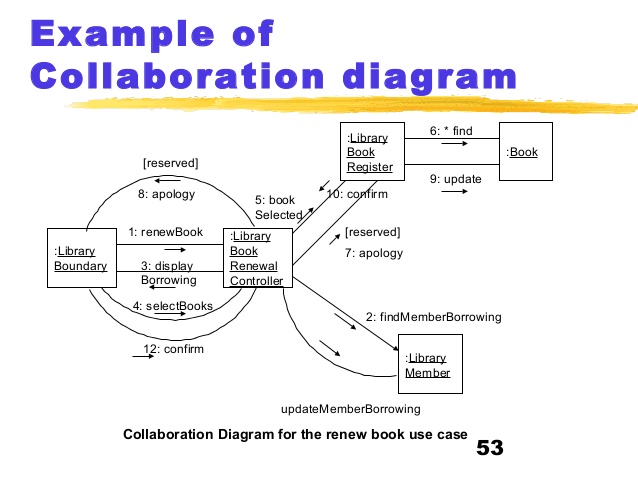


Figure:6.4 Sequence diagram

**6.5.Collaboration diagram**

A collaboration diagram show the objects and relationships involved in an interaction, and the sequence of messages exchanged among the objects during the interaction. It can be the decomposition of a use case, use case diagram, or part of a use case diagram. The collaboration diagram shows messages being sent between classes and object (instances). A diagram is created for each system operation that relates to the current development cycle (iteration).



6.4.1: Collaboration Diagram

**6.6 Activity Diagram**

Activity diagram are a loosely defined diagram to show workflows of stepwise activities and actions, with support for choice, iteration and concurrency. UML, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system.UML activity diagram could potentially model the internal logic of a complex operation. In many ways UML, activity diagrams are object-oriented equivalent of flow charts and data flow diagrams (DFDs) from structural development.

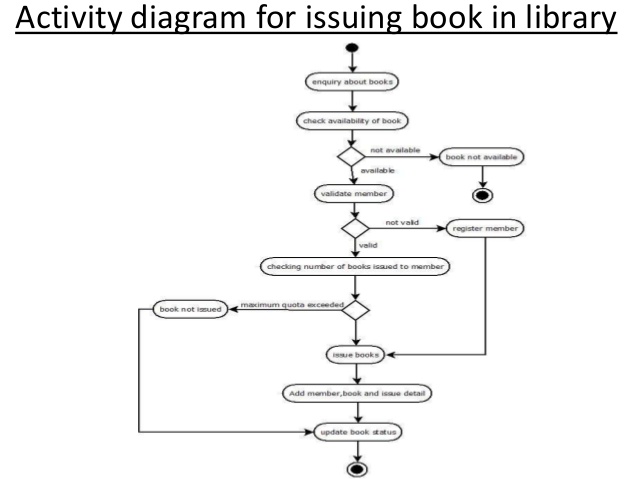
****

Figure 6.5: Activity Diagram