**Chapter 5**

**Software Design**

Software design is the act of determining the user‘s experience with a piece of software. It has nothing to do with how the code works inside, or how big or small the code is. The designer’s task is to specify unambiguous.

It serves as the foundation in software development. Without design, we risk building an unstable system. A design that fails when small changes are made may be difficult to test. The initial and primary goal of a system design is to ensure that the structure of the system supports the right function and organizes them to let sure work efficiently.

**5.1 Visual Modeling Using UML**

Visual modeling is a way of thinking about problems using models organized around real-world ideas. Models reduce the amount of time it tasks to learn and it improves safety, performance, consistency and reduces errors. It communicates effectively with process, maps, charts and diagrams of all types.

The unified modeling language (UML) offers standard semantics and notations for describing objects structure and behavior and has emerged as the design medium of choice for developing large scale distributed object applications.

UML allow people to develop several different types of visual diagrams that represent various aspects of the system. Rational rose supports the development of the majority of these models as follows:-

* Use case Diagram
* Sequence Diagram
* Activity Diagram
* Collaboration Diagram

**Use case Diagram**

It shows the interaction between use cases, which represent the system functionality, and actors or the people or system that provide or receive information from the system. This one shows the overall functionality of the system. One, who interested in the system as a whole can view these diagrams and understand what the system is supposed to accomplish.

**Sequence Diagram**

It shows the object iteration in the use case. It depicts the objects and the classes involved in the scenario and sequence of messages exchanged between the objects to carry out the functionality of the scenario. These are, typically, associated with the use case realizations in the logical view of the system under development.

**Collaboration Diagram**

These are used to show the flow through a specific scenario of a sequence diagram. It focuses on the relationships between the objects. Collaboration diagram show the same information as the sequence diagram, but people look at collaboration diagram for different reasons. Quality assurance engineers and system architecture look at the distribution of processing between objects.

**5.2 Use Case Diagram**:

Student

**5.2.1 Student use case Diagram**

Clerk

**5.2.2 Clerk use case Diagram**

Teacher

**5.2.3 Teacher use case Diagram**

**5.3 Sequence Diagrams**

Sign up form

Security

Data base

student

1:enter info()

5:confirmation()

3: verify()

4: update database

2:check validation()

**5.3.1 Student signup sequence diagram**

Login form

Authentication

Tag menu

Update db

Student

1:enter id and password

2:get id and password

3:verify id and password()

5:upload assignment()

Upload assignment form

6:Update db

4:login form()

7: confirmation ()

**5.3.2 Student upload Assignment sequence diagram**

Login form

Authentication

Tag menu

Log out

Student

1:enter id and password()

2:get id and password()

3:verify id and password()

4:login form()

5:view data ()

View updates

6:logout()

7: confirmation ()

**5.3.3 Students view updates sequence diagram**

Login form

Authentication

View assignments

Log out

Teacher

1: enter id and password ()

2:get id and password()

3: verify id and password ()

4: login form ()

5: give marks ()

Add result

6: logout ()

7: confirmation

**5.3.4 Teacher view Assignment Answer sequence diagram**

Login form

Authentication

Upload assignments

Log out

Teacher

1:enter id and password()

2:get id and password()

3:verify id and password()

4:login form()

5:update db ()

Data base

6:logout()

**5.3.5 Teacher Upload Assignment sequence diagram**

Login form

Authentication

Upload results

Log out

clerk

1:enter id and password()

2:get id and password()

3:verify id and password()

4:login form()

5:update db ()

Data base

6:logout()

7: confirmation

**5.3.6 Clerk Upload Result Sequence Diagram**

Login form

Authentication

Add subjects

Log out

clerk

1:enter id and password()

2:get id and password()

3:verify id and password()

4:login form()

5:update db ()

Data base

6:logout()

7:confirmation

**5.3.7 Clerk Add Subject Sequence Diagram**

Login form

Authentication

Add updates

Log out

clerk

1:enter id and password()

2:get id and password()

3:verify id and password()

4:login form()

5:update db ()

Data base

6:logout()

7:confirmation

**5.3.8 Clerk Add Updates Sequence Diagram**