E-CONTENT DEVELOPMENT FOR ENGINEERING COURSES

A PROJECT REPORT

submitted by

SNEHAA.R **212211104081**

SUDHA RANI.M **212211104087**

in partial fulfilment for the award of the degree

of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

SAVEETHA ENGINEERING COLLEGE,THANDALAM

ANNA UNIVERSITY: CHENNAI 600 025

APRIL 2015

ANNA UNIVERSITY: CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this project report **"E-CONTENT DEVELOPMENT FOR ENGINEERING COURSES"** is the bonafide work of **"SNEHAA R (212211104081) and SUDHA RANI M (212211104087)"** carried out the project under my supervision.

SIGNATURE

Mr.R.SARAVANAN, M.E.,(Ph.D.,)

Associate Professor,

HEAD OF THE DEPARTMENT

Dept. of Computer Science and Engineering,

Saveetha Engineering College,

Thandalam,

Chennai-602 105.

SIGNATURE

Dr.P.Valarmathie,M.E., Ph.D

Professor,

SUPERVISOR

Dept. of Computer Science and Engineering,

Saveetha Engineering College,

Thandalam,

Chennai-602 105.

Submitted for the Project work and Viva voce Examination held on _____

Internal Examiner **External Examiner**

ACKNOWLEDGEMENT

We express our deep sense of gratitude to our honorable and beloved Founder President ***Dr. N. M. Veeraiyan,*** our President ***Dr. Saveetha Rajesh,*** our Director ***Dr. S. Rajesh*** and other management members for providing the infrastructure needed.

We express our whole-hearted gratitude to our Principal ***Dr. R. Venkatasamy,*** and ***Prof. R. Dheenadayalu,*** Dean (ICT) for their whole-hearted encouragement in completing this project.

We convey our thanks to ***Mr. R. Saravanan,*** Associate Professor and Head, Department of Computer Science and Engineering, Saveetha Engineering College, for his kind support and for providing necessary facilities to carry out the project work.

We would like to express our sincere thanks and deep sense of gratitude to our guide ***Dr. P. Valarmathie ,*** Professor, Department of Computer Science and Engineering, Saveetha Engineering College, for her valuable guidance, suggestions and constant encouragement. We express our sincere thanks to the project coordinator ***Dr. G. Nagappan,*** Associate Professor, Department of Computer Science and Engineering, Saveetha Engineering College, for providing us necessary support and details at the right time and during the progressive reviews.

We owe our thanks to all the members of our college, ***faculty, staff and technicians*** for their kind and valuable co-operation during the course of the project. We are pleased to acknowledge our sincere thanks to our ***beloved parents, friends*** and ***well-wishers*** who encouraged us to complete this project successfully.

ABSTRACT

Efficiency and effectiveness of learning process can be improved by adaptations to learners' learning styles. But for the time being, most of existing education systems lack of adaptation or personalization; every learner is delivered the same learning contents. Many researchers have been studying to find out an efficient way of students' learning style identification for a better personalization. In our study, we concentrate on intelligent agents that can provide the learners with personal assistants to carry out learning activities according to their learning styles and knowledge level. In this project, a new literature-based method is preferred that uses learners' behaviors on learning objects as indicators for estimating students' learning styles during an online course conducted in the learning management system. The evaluation of learning style estimation and adaptation from the experiment show a high precision. Together with the mentioned benefits of learning style adaptation, this result indicates that the method is capable for wide use.

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.

CHAPTER NO.	TITLE	PAGE NO.

CHAPTER NO.	TITLE	PAGE NO.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

- ILT -INSTRUCTOR LED TRAINING
- LMS -LEARNING MANAGEMENT STYLES
- IDP -INDIVIDUAL DEVELOPMENT PLAN
- FSLSM -FELDER-SILVERMAN LEARNING STYLE MODEL
- CMS -OURSE MANAGEMENT SYSTEM
- UML -UNIFIED MODELING LANGUAGE
- DFD -DATA FLOW DIAGRAM
- SysML -SYSTEMS MODELING LANGUAGE
- ER -ENTITY-RELATIONSHIP
- ERM -ENTITY-RELATIONSHIP MODEL
- EEN -EXECUTION ENVIRONMENT NODE
- ERD -ENTITY-RELATIONSHIP DIAGRAM

CHAPTER 1

INTRODUCTION

1.1 AIM

The main scope of this project is to Developing an e-learning initiative is a typically much larger endeavor than that of an instructor-led training (ILT) program. Consider the increased expenses, number of people involved, development time, technological requirements, and delivery options. This additionally includes a cognitive test in the beginning to test the skills of the learner. The skills are categorized into 4 sets: Basic skills, planning skills, education skills and knowledge skills.

1.2 PROBLEM DEFINITION

The main Objective of this paper is to aim at the automatic detection of learning styles to avoid intentional or unintentional wrong answers, and to save students' time on filling in a questionnaire. In the world of digitalization it is a challenging act for students to perfectly learn and understand a concept using diversely available e-learning content. The curriculum available should be common to students and the teachers so that there is a connectivity and easy understanding of the course material. The problem that is faced during such learning method is that the tutor finds it hard to keep track of the performance of the students and the communication is mostly depended over the other sources. Thus e-learning software with more features developed to analyze the performance level of the students and where they lag or where they need more input is analyzed through repetitive cognitive tests and such. It is seen that usually this type of

assessment is done through answering a long questionnaire at some time during the start of the course commencement. But the real problem is that the skill level of the students may change in time and assessing feature had to keep up with that. And also the learning software is yet to prepared for the subjects in engineering colleges. These problems are dealt with in the proposed project.

1.3 DESCRIPTION

ELECTRONIC LEARNING SYSTEM

Electronic-learning is a state-of-the-art technology of education that implies self-motivation, communication, efficiency, and technology.

Because there is limited social interaction, students must keep themselves motivated. E-learning is effective as it eliminates distances and subsequent commutes. Distance is eliminated because the e-learning content is designed with media that can be accessed from properly equipped computers, and other means of internet accessible technology.

E-learning has its own advantages and disadvantages. However, the most important advantages lay in the reduction of time, efforts and cost, adding too which the capability of the computer in improving the general level of studying achievement and in helping the teacher and the student in providing an attractive educational environment which doesn’t depend on place or time.

Furthermore, we can talk about the following advantages of E-learning with the increasing of communication between the student and themselves and between the student and the school a fast and easy way is needed to bring everybody

together. So from the midst of confusion came e-learning to make their lives easy. Throughout different directions such as discussing forums, e-mail, and chat rooms. Moreover, researchers think that these things increase and motivate students to participate and react with the subjects in question. There is also the matter of sharing points of views. So the discussing forums and chat rooms provide opportunities to exchange and share the point of views in respect of the subjects in question, the matter which increases the opportunities of making use of the opinions and suggestions of the others, which, in turn, will help in making a strong background at the learner through what he /she got of knowledge and skills from the chat rooms.

1.4 ADVANTAGES OF USING E-LEARNING SYSTEM

- Class work can be scheduled around work and family
- Reduces travel time and travel costs for off-campus students
- Students may have the option to select learning materials that meets their level of knowledge and interest
- Students can study anywhere they have access to a computer and Internet connection
- Self-paced learning modules allow students to work at their own pace
- Flexibility to join discussions in the bulletin board threaded discussion areas at any hour, or visit with classmates and instructors remotely in chat rooms

- Instructors and students both report eLearning fosters more interaction among students and instructors than in large lecture courses
- eLearning can accommodate different learning styles and facilitate learning through a variety of activities
- Develops knowledge of the Internet and computers skills that will help learners throughout their lives and careers
- Successfully completing online or computer-based courses builds self-knowledge and self-confidence and encourages students to take responsibility for their learning
- Learners can test out of or skim over materials already mastered and concentrate efforts in mastering areas containing new information and/or skills

1.5 SCOPE OF THE PROJECT

The main scope of this project is to Developing an e-learning initiative is a typically much larger endeavor than that of an instructor-led training (ILT) program. Consider the increased expenses, number of people involved, development time, technological requirements, and delivery options.

CHAPTER 2

LITERATURE SURVEY

2.1 CONTENT MANGEMENT SYSTEM

A course management system (CMS) is a collection of software tools providing an online environment for course interactions. A CMS typically includes a variety of online tools and environments, such as: An area for faculty posting of class materials such as course syllabus and handouts. This can be typically explained as a learning tool designed and made by the tutors for his/hers remote students and also for advanced teaching methods.

2.2 LEARNING MANAGEMENT SYSTEM

A learning management system (LMS) is a software application for the administration, documentation, tracking, reporting and delivery of electronic educational technology (also called e-learning) education courses or training programs.

LMS is the framework that handles all aspects of the learning process. An LMS is the infrastructure that delivers and manages instructional content, identifies and assesses individual and organizational learning or training goals, tracks the progress towards meeting those goals, and collects and presents data for supervising the learning process of the organization as a whole.

A Learning Management System delivers content but also handles registering for courses, course administration, skills gap analysis, tracking, and reporting.

2.3 SYSTEM FUNCTIONALITY

- Course Content Delivery
- Student Registration and Administration
- Training Event Management (i.e., scheduling, tracking)
- Curriculum and Certification Management
- Skills and Competencies Management
- Skill Gap Analysis
- Individual Development Plan (IDP)
- Reporting
- Training Record Management
- Courseware Authoring
- Resource Management
- Virtual Organizations
- Performance Management System Integration

2.4 EXISTING SYSTEM

Adaptive e-learning systems allow students learn by themselves so that it would improve learning effect and overcome the disadvantage of traditional class teaching. To implement adaptation or personalization in such learning management systems (LMSs), students' learning styles need to be identified first.

Beside the static approach that uses a questionnaire for identifying learning styles, there are two approaches that automatically detect learning styles: data-driven and literature-based. The literature-based approach is the newest and it is still studied by few researchers. It investigates learners' behaviors in their interactions with LMSs.

2.4.1 DRAWBACKS OF EXISTING SYSTEM

- It does not require learners to waste their time on completing a questionnaire.
- This approach has a noticeable promising result in identifying learning styles not only precisely but also automatically and dynamically.
- This character is worth considering because learning styles may change over time.

2.5 PROPOSED SYSTEM

To implement adaptation or personalization in such learning management systems (LMSs), students' learning styles need to be identified first. Beside the static approach that uses a questionnaire for identifying learning styles, there are two approaches that automatically detect learning styles: data-driven and literature-based. The literature-based approach is the newest and it is still studied by few researchers. It investigates learners' behaviours in their interactions with LMSs. It does not require learners to waste their time on completing a questionnaire. This approach has a noticeable promising result in identifying learning styles not only precisely but also automatically and dynamically. This character is worth considering because learning styles may change over time

We concentrate on the literature-based approach with respect to Felder-Silverman Learning Style Model (FSLSM). We promote a new method to estimate each student's learning styles based on the number of visits and the time that he spent on learning objects. The method was experimented in a web based LMS called POLCA. Their initial results in discovering learning styles and matching learning objects with suitable learners are promising.

In the proposed system we designed, the procedure starts with the initial cognitive test where the students had to answer simple questions about themselves. These data are stored and is compared along with the data collected during their study time and their performance in the periodic examinations conducted by the tutors. These results are then analyzed in four different skill areas such as basic skills, knowledge skills, education skills and planning skills.

2.5.1 MERITS OF THE PROPOSED SYSTEM

- Using an LMS helps keep all the information consistent. Each student is enrolled through the LMS, so no one gets left off an e-mail list or gets the wrong information.
- Consistency is an advantage for teachers on leave, sick days and vacation as well, because the students can receive study material in the absence of their teachers.
- A good LMS will also allow for the students to customize their options as well, which makes the coursework much more interactive and adaptable for them.

CHAPTER 3

SYSTEM ANALYSIS

3.1 FEASIBILITY STUDY

The main objective of feasibility study is to test the Technical, operational and economical for adding new modules and debugging old running system. All system is feasible for adding new modules and debugging old running system.

There are aspects in the feasibility study portion of the preliminary investigation:

- Technical feasibility
- Economic feasibility
- Operational feasibility

3.1.1 TECHNICAL FEASIBILITY

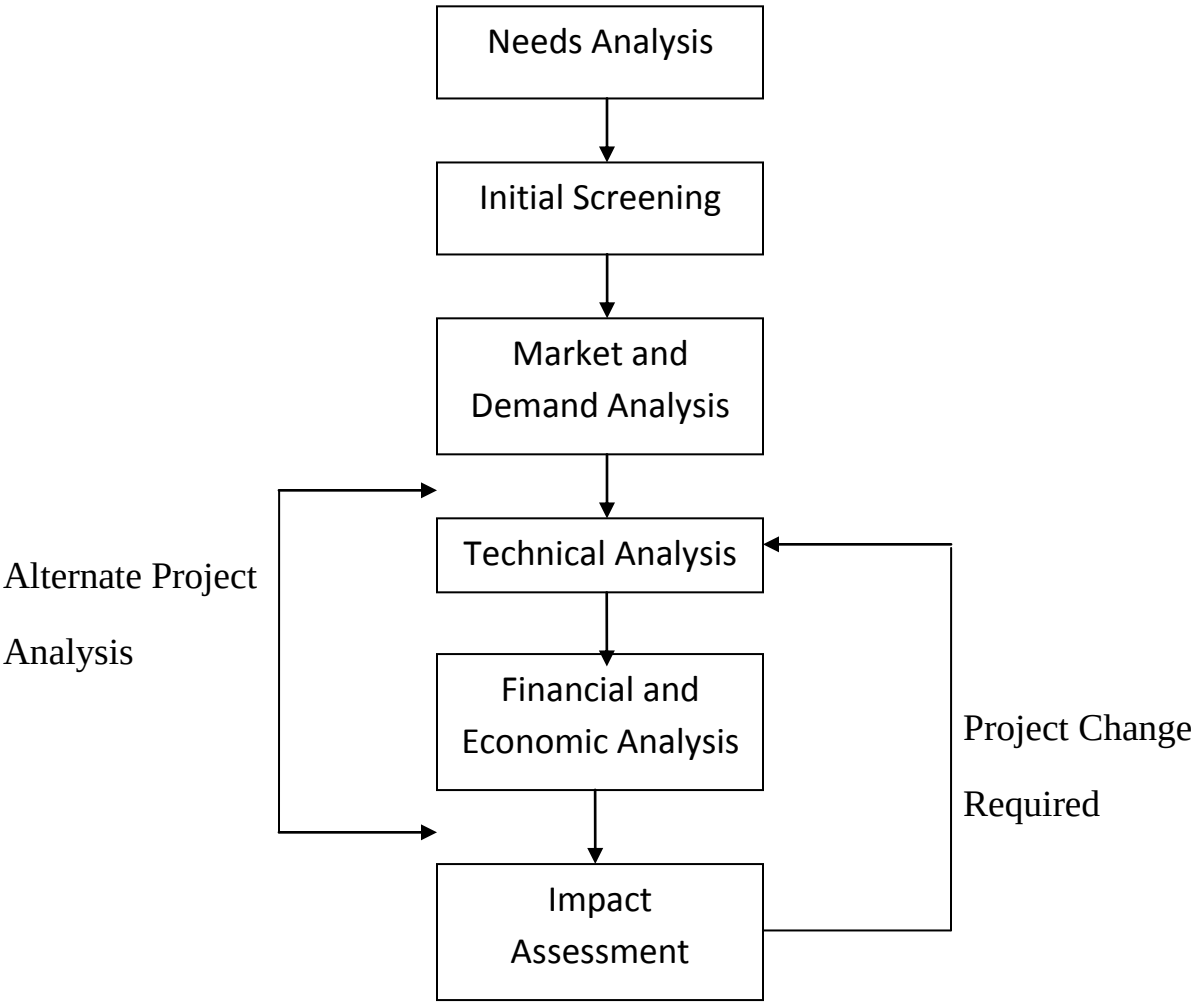
This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

3.1.2 ECONOMIC FEASIBILITY

Economic feasibility is the most frequently used method for evaluating the effectiveness of the proposed system. More commonly known as cost-benefit analysis, the procedure that costs for a proposed system and weights them against the tangible and intangible benefits of the system. If the benefit out weight costs,

then the system is designed and implemented. Otherwise further justification and alterations in the proposed system have to be made, if it has the chance of being approved. A system can be developed technically and that will be used if installed must still be an investment for the organization. In the economic feasibility, the development cost in creating the system must be evaluated against the ultimate benefits from the new system. Financial benefit must be equal or exceeds the costs derived.

Fig 3.1 Feasibility Analysis



3.1.3 OPERATIONAL FEASIBILITY

Operational feasibility is a measure of how well a proposed system solves the problems, and takes advantage of the opportunities identified during scope definition and how it satisfies the requirements identified in the requirements analysis phase of system development. The aspect of the study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system. efficiently. The user must not feel threatened by the system, instead must accept it as necessity. The level of acceptance by the user solely depends on the method that are employed to educate the user about the system and to make them familiar with it.

3.2 HARDWARE USED

Processor	: Pentium IV
Hard Disk	: 40GB
RAM	: 512MB or more

3.3 SOFTWARE USED

Operating System	: Windows XP/2007 or Linux
User Interface	: HTML, CSS
Client-side Scripting	: JavaScript
Programming Language	: Java
Web Applications	: JDBC, Servlets, JSP
IDE/Workbench	: Net beans 7.3
Database	: MySQL 5.1
Server Deployment	: Tomcat 7

3.4 FUNCTIONAL REQUIREMENT

3.4.1 NUMBER OF MODULES

The system after careful analysis has been identified to be presented with the following modules:

1. Content Management
2. Initial Evaluation
3. Record Activities
4. Periodic Evaluation
5. Periodic Comparison
6. Final Evaluation and Result

3.4.2 MODULE DESCRIPTION

3.4.2.1 Content Management

A course management system (CMS) typically includes a variety of online tools and environments, such as: An area for faculty posting of class materials such as course syllabus and handouts.These are the pages that are associated with the logic pages, and contain content specific to that page.

- Organizing the topics according to the category
- Organizing the subtopics according to the subcategory
- Providing appropriate links to specified category
- A mechanism to export and import data from and to XML files
- Generating maps for specific category
- Providing a quiz after completion of specified course

3.4.2.2 Initial Evaluation

Initial evaluation is done before the students enter into the learning process. A questionnaire is prepared which consists of different items that reflect the cognitive skills of students. These items are taken from different learning styles models. Pre-test is conducted for students using this questionnaire. This pre – test helps to find the initial cognitive level of students. The students have to first register their details to join in LMS. Then they have to answer these items using Likert scale. The Likert scale used here is 5 – point. These values are stored in the database.

3.4.2.3 Record Activities

During the learning process, the activities of the students such as submitting assignments, attending classes regularly, taking tests, test marks, their navigations in course, spending extra time for learning, asking questions in class, showing interest in learning other courses, etc., are recorded automatically by the system.

3.4.2.4 Periodic Evaluation

With the help of the recorded activities, periodic evaluation is done automatically by the system. If the students perform the activities in the same way as they have answered in the questionnaire, their scores will increase. Otherwise their scores will decrease. These updated scores are stored in another database. For example, if a student gives three rates for the item “Understanding the concept faster”, this rate will increase if he could really understand the concept that is completion of the course within the expected allotment of time. Otherwise this rate will decrease. This evaluation is done periodically and the data is maintained in the back end and is used for future and further calculations to be made.

3.4.2.5 Periodic Comparison

The initially evaluated scores are compared with the periodically evaluated scores to find the students development in cognitive skills. If the updated score is higher than the initial score, then the student is said to be improved in his cognitive skills.

3.4.2.6 Final Evaluation and Result

The final evaluation is done at the end of the course to find students final cognition level. No post-test is conducted to do the final evaluation. At last, the final result about their cognitive skill is given.

CHAPTER 4

SYSTEM DESIGN

4.1 SYSTEM ARCHITECTURE

Design Engineering deals with the various UML [Unified Modeling language] diagrams for the implementation of project. Design is a meaningful engineering representation of a thing that is to be built.

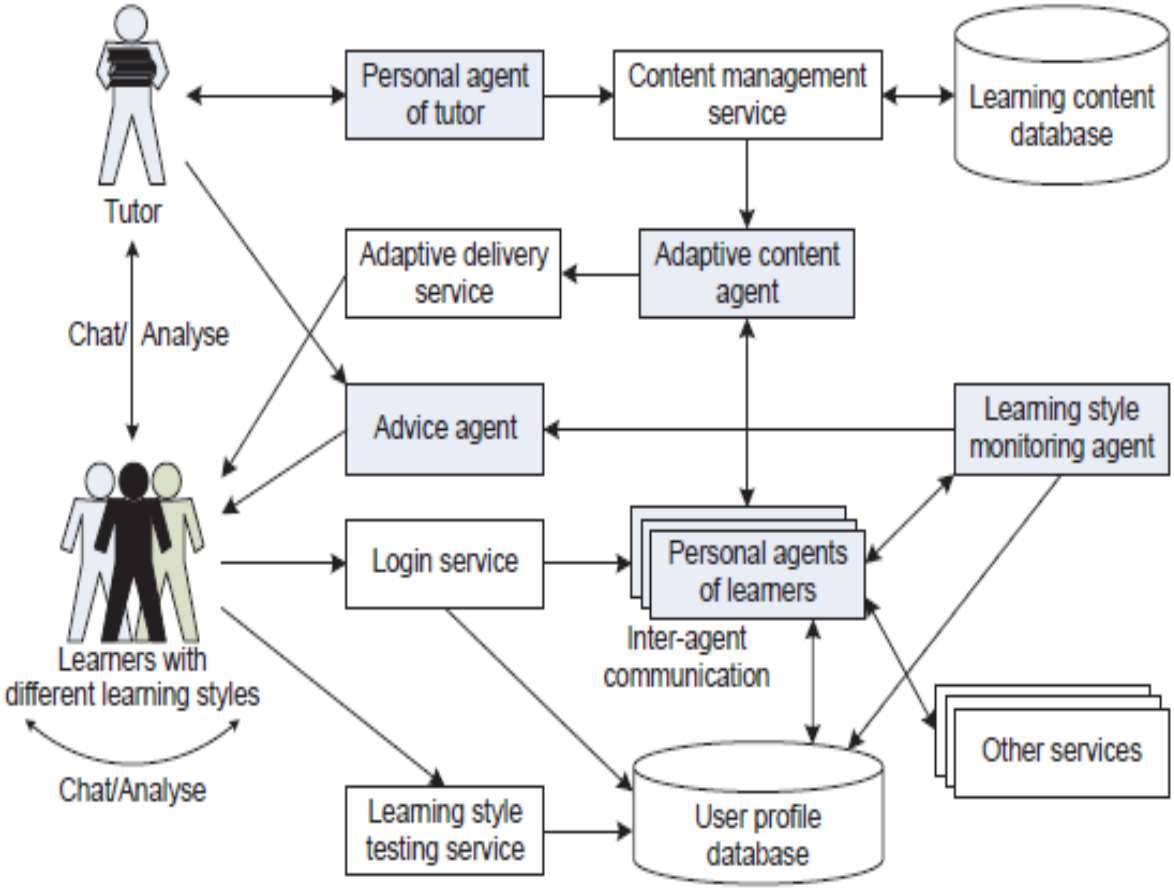


Figure 4.1: System Architecture

Software design is a process through which the requirements are translated into representation of the software. Design is the place where quality is rendered in software engineering. Design is the means to accurately translate customer requirements into finished product.

4.2 DATA FLOW DIAGRAM

Data flow diagram (DFD) is a graphical representation of the "flow" of data through an information system. It differs from the flowchart as it shows the data flow instead of the control flow of the program.

A data flow diagram can also be used for the visualization of data processing. The DFD is designed to show how a system is divided into smaller portions and to highlight the flow of data between those parts.

Data Flow Diagram (DFD) is an important technique for modeling a system's high-level detail by showing how input data is transformed to output results through a sequence of functional transformations.

DFDs reveal relationships among and between the various components in a program or system. DFD consists of four major components: entities, processes, data stores and data flow.

4.2.1 DFD LEVEL 0

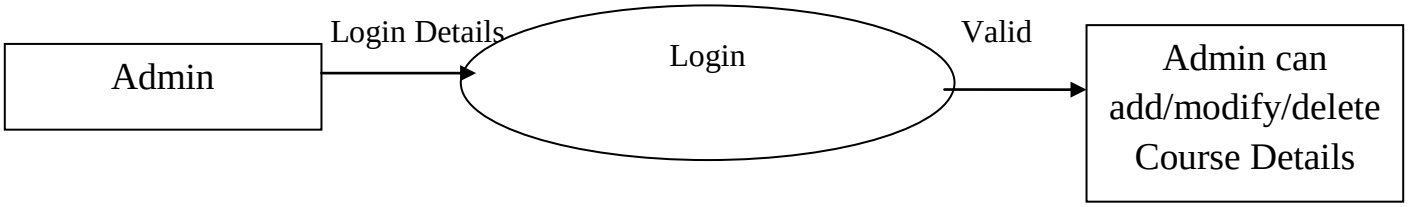


Figure 4.2: Level 0 DFD

4.2.2 DFD LEVEL 1

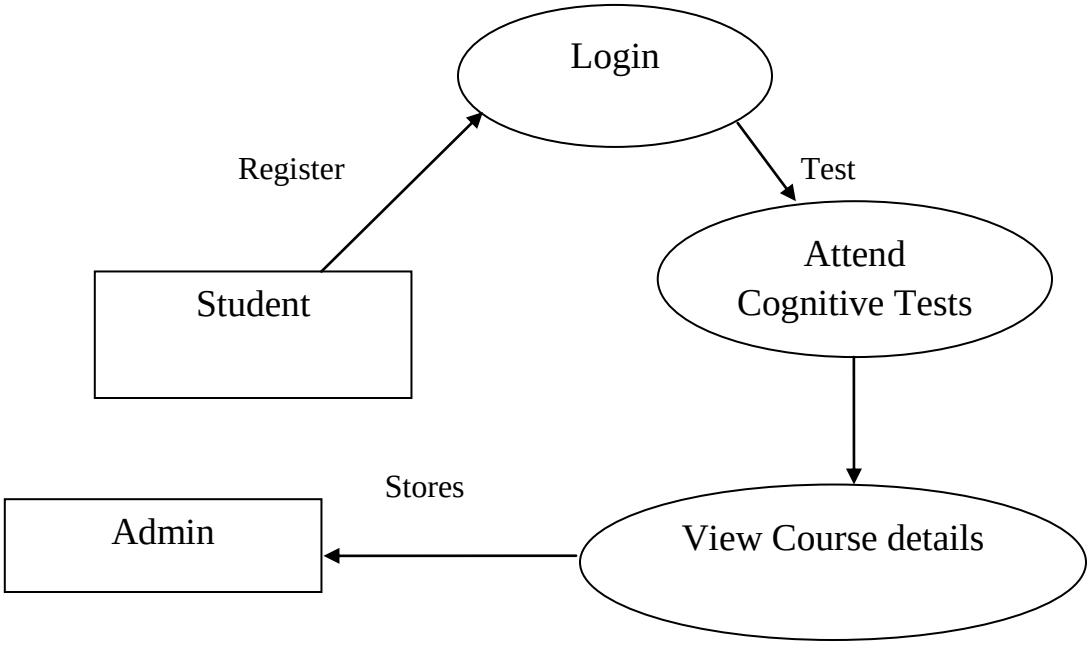


Figure 4.3: Level 1 DFD

4.2.3 DFD LEVEL 2

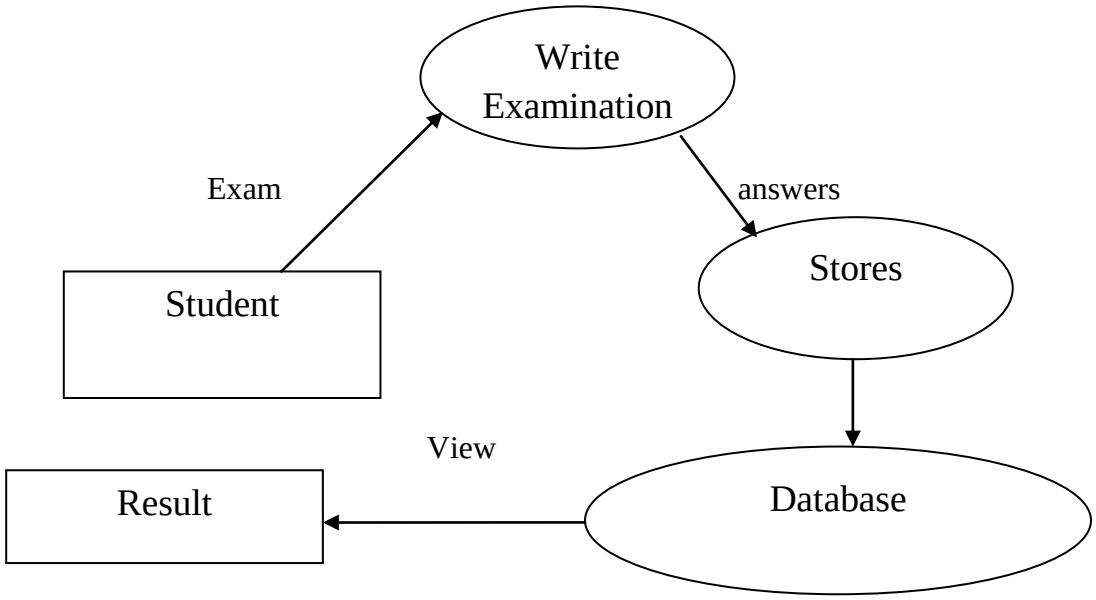


Figure 4.4: Level 2 DFD

4.2.4 OVERALL DFD

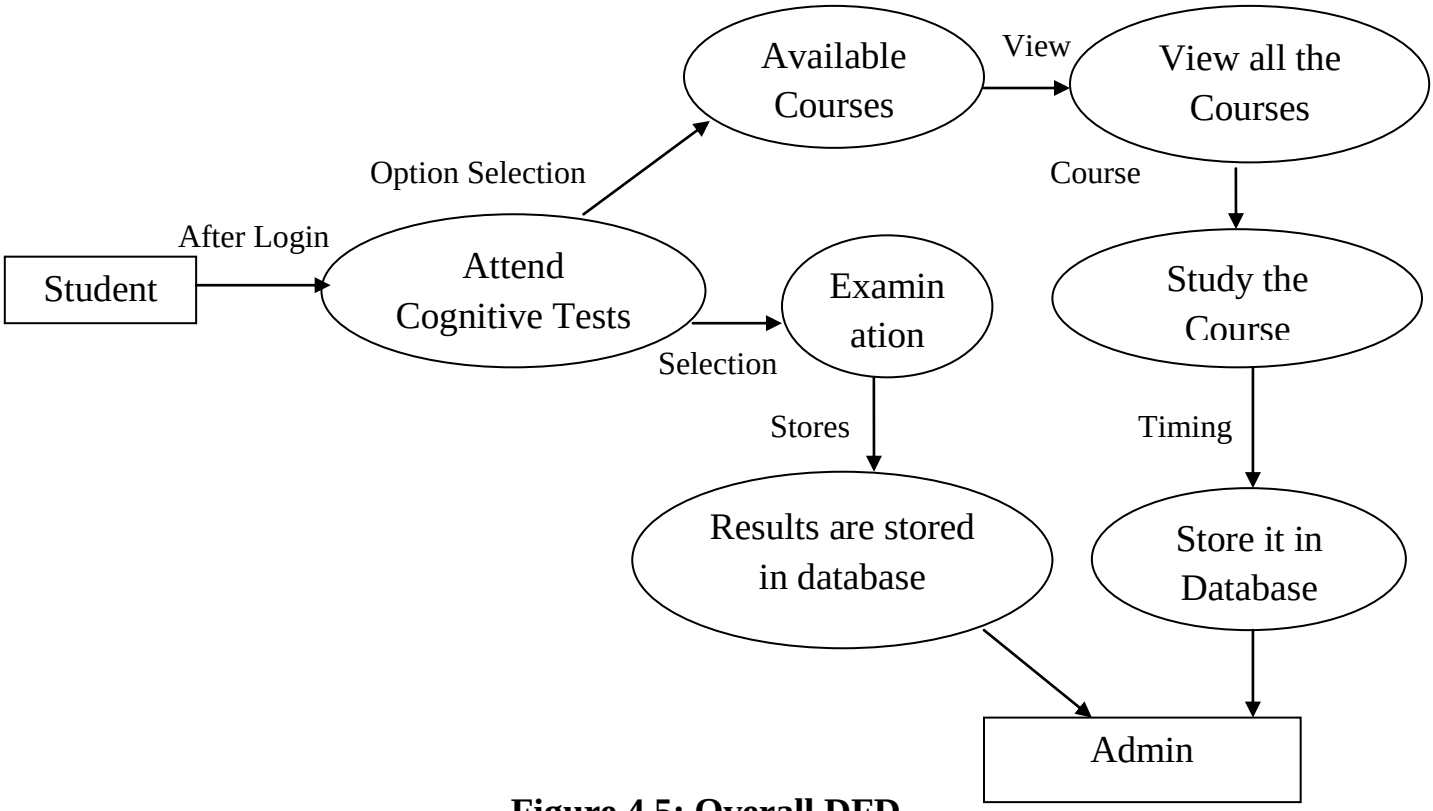


Figure 4.5: Overall DFD

4.3 UML DIAGRAMS

4.3.1 USE CASE DIAGRAM

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted. Use case diagrams are formally included in two modeling languages

defined by the OMG: the Unified Modeling Language (UML) and the Systems Modeling Language (SysML). Use case diagrams are behavior diagrams used to describe a set of actions (use cases) that some system or systems (subject) should or can perform in collaboration with one or more external users of the system (actors). Each use case should provide some observable and valuable result to the actors or other stakeholders of the system.

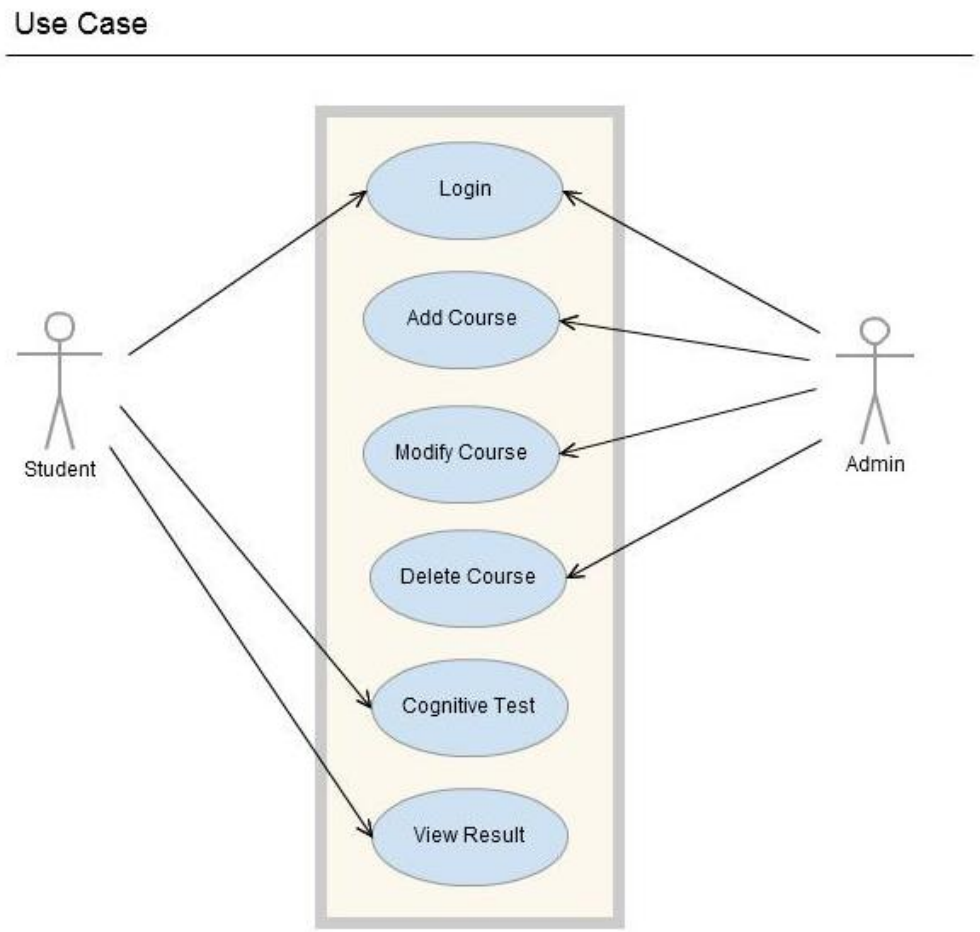


Figure 4.6: Use case diagram

4.3.2 CLASS DIAGRAM

A class diagram in the 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.

Private visibility hides information from anything outside the class partition. Public visibility allows all other classes to view the marked information. Protected visibility allows child classes to access information they inherited from a parent class.

In software engineering, 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, operations (or methods), and the relationships among the classes.

A class diagram is an illustration of the relationships and source code dependencies among classes in the Unified Modeling Language (UML).

In a class diagram, the classes are arranged in groups that share common characteristics.

A class diagram resembles a flowchart in which classes are portrayed as boxes, each box having three rectangles inside. The top rectangle contains the name of the class; the middle rectangle contains the attributes of the class; the lower rectangle contains the methods, also called operations, of the class. Lines, which may have arrows at one or both ends, connect the boxes. These lines define the relationships, also called associations, between the classes.

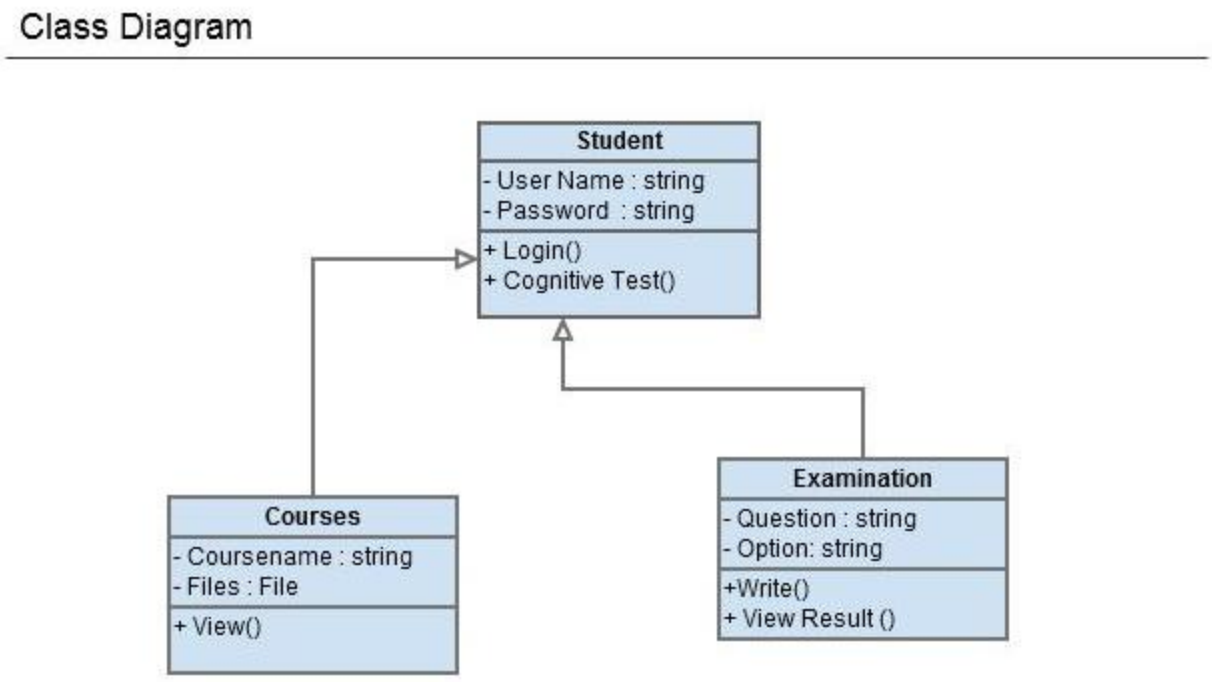


Figure 4.7: Class Diagram

4.3.3 SEQUENCE DIAGRAM

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes 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. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams typically are associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

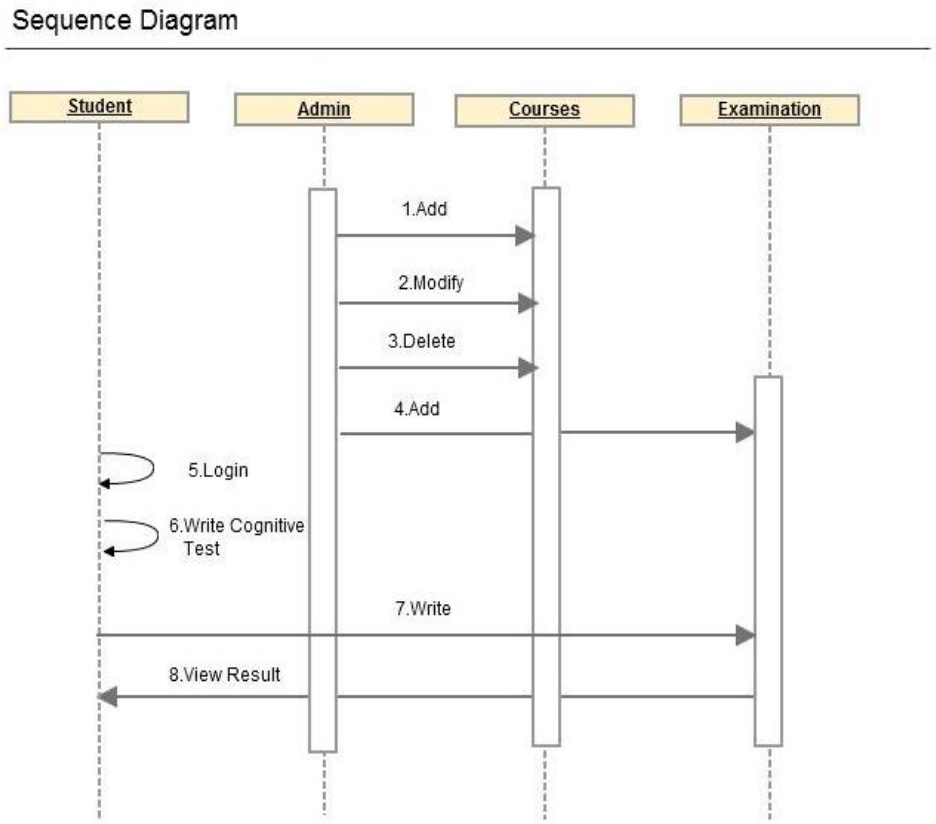


Figure 4.8: Sequence Diagram

4.3.4 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. The collaboration diagram can be a decomposition of a class, class diagram, or part of a class diagram. 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).

A collaboration diagram resembles a flowchart that portrays the roles, functionality and behavior of individual objects as well as the overall operation of the system in real time. Objects are shown as rectangles with naming labels inside. These labels are preceded by colons and may be underlined. The relationships between the objects are shown as lines connecting the rectangles. The messages between objects are shown as arrows connecting the relevant rectangles along with labels that define the message sequencing.

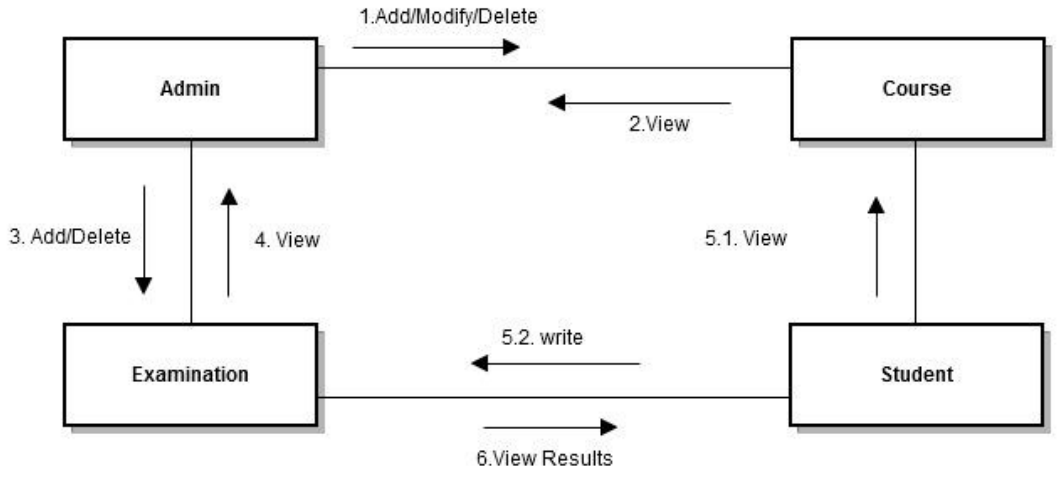


Figure 4.9: Collaboration Diagram

4.3.5 ACTIVITY DIAGRAM

Activity diagram are graphical representation of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modelling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

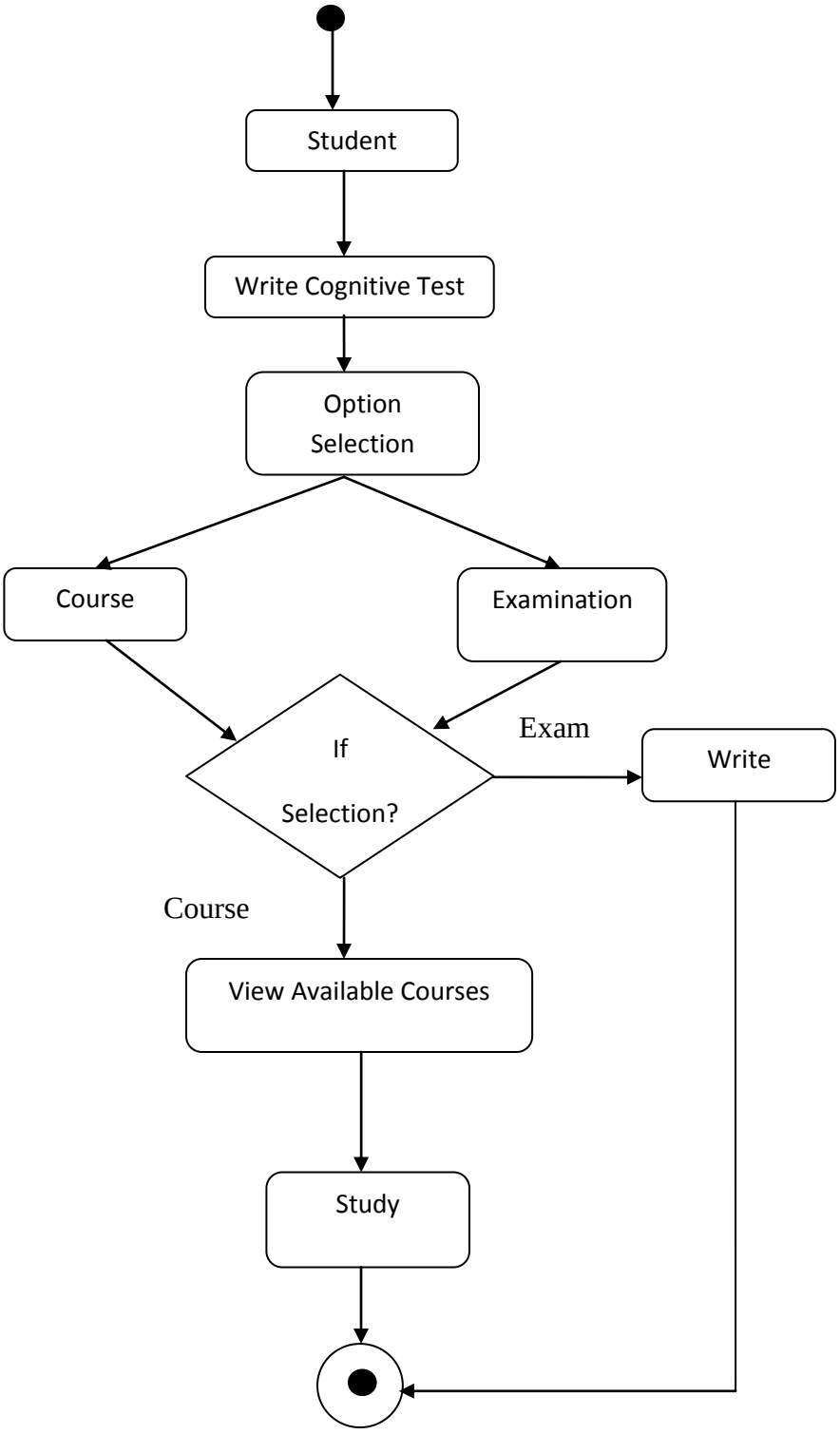


Figure 4.10: Activity Diagram

4.3.6 E-R DIAGRAM

In software engineering, an entity-relationship model (ERM) is an abstract and conceptual representation of data. Entity-relationship modeling is a database modeling method, used to produce a type of conceptual schema or semantic data model of a system, often a relational database, and its requirements in a top-down fashion. Diagrams created by this process are called entity-relationship diagrams, ER diagrams, or ERDs. An entity-relationship (ER) diagram is a specialized graphic that illustrates the relationships between entities in a database. ER diagrams often use symbols to represent three different types of information. Boxes are commonly used to represent entities. Diamonds are normally used to represent relationships and ovals are used to represent attributes.

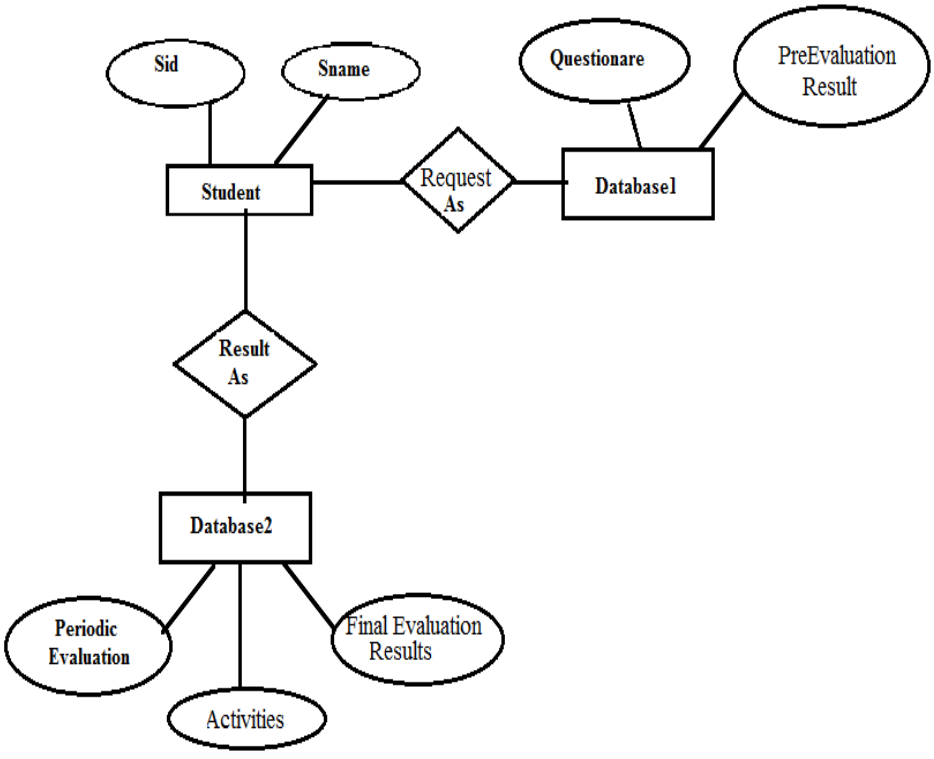


Figure 4.11: E-R Diagram

4.3.7 COMPONENT DIAGRAM

In the Unified Modeling Language, a component diagram depicts how components are wired together to form larger components and or software systems. They are used to illustrate the structure of arbitrarily complex systems.

The Component Diagram helps to model the physical aspect of an Object-Oriented software system. It illustrates the architectures of the software components and the dependencies between them. Those software components including run-time components, executable components also the source code components. A component diagram has a higher level of abstraction than a Class Diagram - usually a component is implemented by one or more classes (or objects) at runtime. They are building blocks so a component can eventually encompass a large portion of a system.

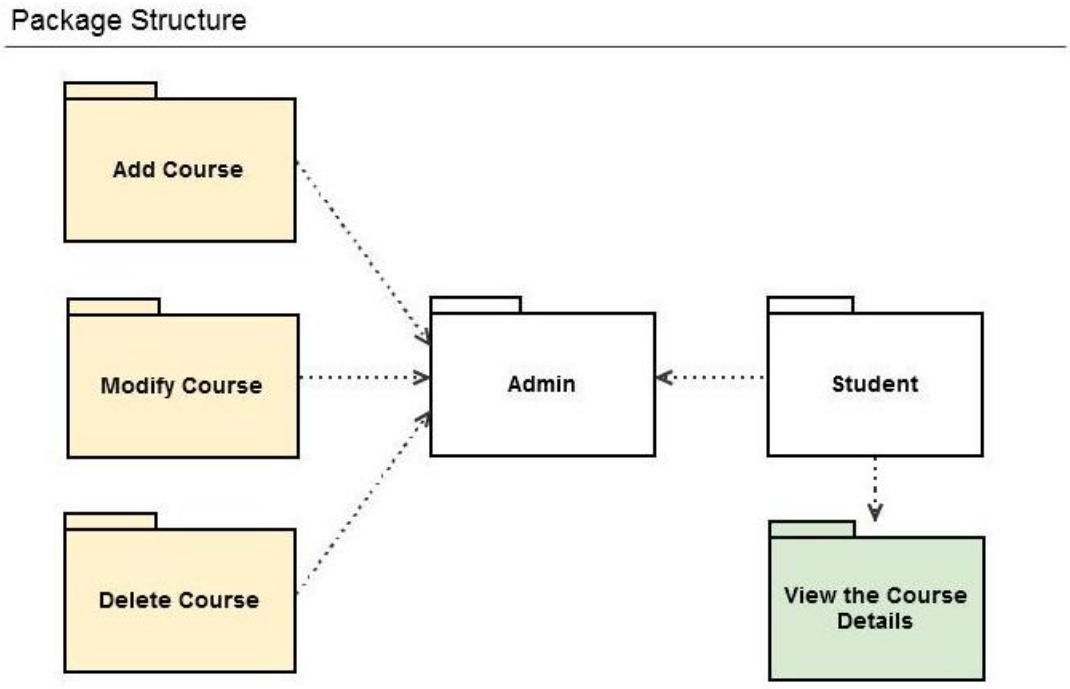


Figure 4.12: Package Diagram

4.3.8 DEPLOYMENT DIAGRAM

A deployment diagram the Unified Modeling Language models the physical deployment of artifacts on nodes. 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. There are two types of Nodes; Device Node and Execution Environment Node. Device nodes are physical computing resources with processing memory and services to execute software, such as typical computers or mobile phones. An execution environment node (EEN) is a software computing resource that runs within an outer node and which itself provides a service to host and execute other executable software elements.

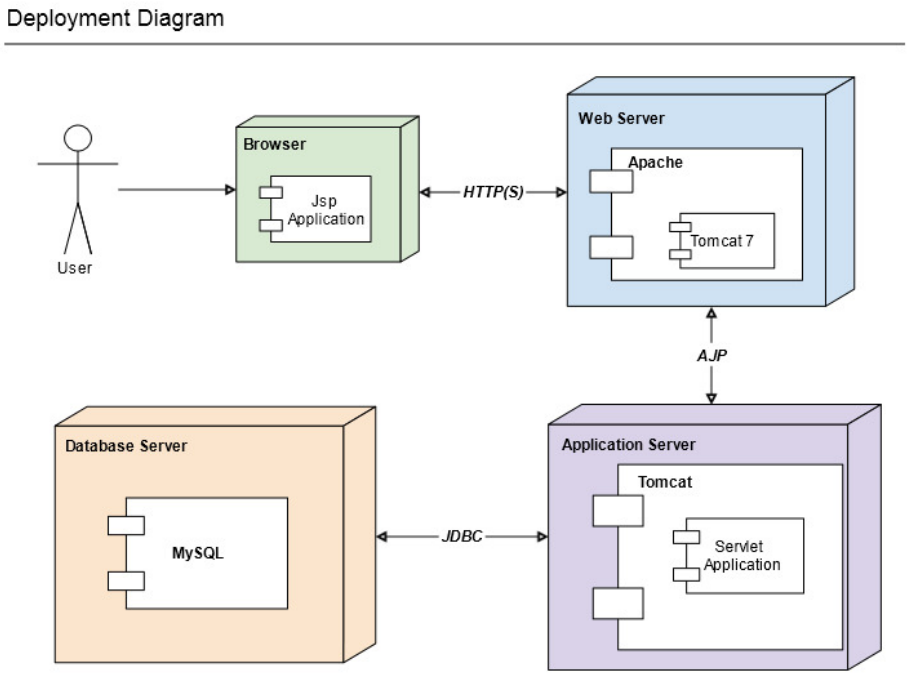


Figure 4.13: Deployment Diagram

CHAPTER 5

TESTING AND RESULTS

5.1 IMPLEMENTATION

The implementation process of our proposed project is explained in the following six steps. It is seen that the process is mainly focused over the aim to establish a concrete environment for both the tutor and the student to work efficiently. The following process will ensure that the student's level of understanding and processing and perceptive knowledge are analyzed carefully.

STEP 1: Content Management

A Course Management System (CMS) typically includes a variety of online tools and environment, such as: An area for faculty posting of class materials such as course syllabus and handouts. These are the pages that are associated with the logic pages, and contain content specific to that page.

- Organizing the topics according to that category
- Organizing the subtopics according to the subcategory
- Providing appropriate links to specified category
- A mechanism to export and import data from and to XML files
- Generating maps for specific category
- Providing a quiz after completion of specified course

This helps to guarantee that the system works efficiently and that the links between the pages will be consistent and that the expected data could be easily achieved.

STEP 2: Initial Evaluation

Initial evaluation is done before the students enter into the learning process. A questionnaire is prepared which consists of different items that reflect the cognitive skills of students. These items are taken from different learning styles models. Pre-test is conducted for students using this questionnaire. This pre – test helps to find the initial cognitive level of students. The students have to first register their details to join in LMS. Then they have to answer these items using Likert scale. The Likert scale used here is 5 – point. These values are stored in the database.

STEP 3: Record Activities

During the learning process, the activities of the students such as submitting assignments, attending classes regularly, taking tests, test marks, their navigations in course, spending extra time for learning, asking questions in class, showing interest in learning other courses, etc., are recorded automatically by the system.

STEP 4: Periodic Evaluation

With the help of the recorded activities, periodic evaluation is done automatically by the system. If the students perform the activities in the same way as they have answered in the questionnaire, their scores will increase. Otherwise their scores will decrease. These updated scores are stored in another database. For example, if a student gives three rates for the item "I will perform well in all the subjects", this rate will increase if he scores good marks in all the subjects. Otherwise this rate will decrease.

STEP 5: Periodic Comparison

The initially evaluated scores are compared with the periodically evaluated scores to find the students development in cognitive skills. If the updated score is higher than the initial score, then the student is said to be improved in his cognitive skills.

STEP 6: Final Evaluation and Result

The final evaluation is done at the end of the course to find students final cognition level. No post-test is conducted to do the final evaluation. At last, the final result about their cognitive skill is given.

A learning style model was developed by Richard Felder and Linda Silverman called as Felder-Soloman's Index of Learning Styles (ILS). Their model permits classify students in four categories; Sensory/Intuitive, Visual/Verbal, Active/Reflective, and Sequential/Global. The dimensions Sensory/Intuitive and Visual/Verbal refer to the mechanisms of perceiving information. The dimensions Active/Reflective and Sequential/Global are concerned with processing and transforming information in understanding.

Expected time spent on each learning object, $Time_{expected\_stay}$, is determined by the tutor who is responsible for the course. This time can be qualitatively identified as the maximum time used by a moderate student to finish studying the learning object. It can also be quantitatively determined in case the learning object is a video or an audio file. The time that a learner actually stays on each learning object, $Time_{spent}$, is recorded by the system. These pieces of time are also the ones calculated for each learning style labelled for the learning objects. For instance, if $Time_{expected\_stay}$ of a Reflective Sensing learning object is 15 second, then

$Time_{expected\_stay}$ assigned for Reflective, as well as for sensing is 15 second. After a period P, which is passed as a system parameter (for example, four weeks), sums of $Time_{spent}$ for each of all eight learning style elements of the learner is calculated. Then we find out eight respective ratios of time (RT):

$$RT_{LS\_element} = \frac{\sum Time_{spent}}{\sum Time_{expected\_stay}}$$

To calculate the number_of_visits' ratios, $RV_{LS\_element}$, number of learning objects visited and total of learning objects with respect to each learning style element are counted. Those data are then used in the following formula:

$$RV_{LS\_element} = \frac{\sum LOs_{visited}}{\sum LOs}$$

Obviously, with a right attitude of teachers and learners, the obtained values of RTs and RVs are between 0 and 1. The average of RT and RV for each of eight learning style element, R_{avg}, is then used to estimate learners' learning styles using the following simple rule:

R_{avg}	**LS Preference**
0 – 0.3	Weak
0.3 – 0.7	Moderate
0.7 – 1	Strong

Table 5.1: R_{avg}- LS Preference

With reference to this value the results to the cognitive test are calculated. The system works on the following algorithm and pseudocode.

ALGORITHM

Algorithm 1 Algorithm of CSC (Conceptual Subspace Clustering)

input: context (O, A, R)
input: $m =$ the minimum density of clustering
output: hierarchical conceptual clusters
generate $ODC =$ ordered dense context $(O, A^{\triangleleft}, R)_m$
$D = ||O||$ //D is the density of cluter
if $O' \neq \emptyset$ **then**
  (O, O') is a conceptual cluster
  $ODC = (O, (A - O')^{\triangleleft}, R)_m$
end if
while $D >= m$ **do**
  for all ordered dense context **do**
    find trunk attributes
    for all trunk attribute a_i **do**
      $(\{a_i\}', \{a_i\})$ is a conceptual cluster //generate sub-concept in ODC
      generate projective context of $(\{a_i\}', \{a_i\})$
      $ODC =$ new ordered dense context
    end for
  end for
  $D = ||O||$ // in ODC
end while

PSEUDO CODE

```
private void showResult(PrintWriter out,HttpSession hs)throws Exception
{
System.out.println("***");
Vector v=(Vector)hs.getValue("data");
System.out.println("Uname :"+uname);
System.out.println("Size1 :"+vsize);
vsize=(Integer)v.size()-1;
System.out.println("Size :"+vsize);
```

```
for(sl=0;sl<vsize;sl++)
{
Questions q=(Questions)v.elementAt(sl);
System.out.println("compare :"+q.ar+"-"+q.ans);
if(q.ar.equals(q.ans))
{
marks+=2;
++a;
System.out.println("Marks:"+marks);
}
}
}
```

5.2 SYSTEM TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the

Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

5.2.1 TYPES OF TESTS

The following explains the different types of the testing that is been done over the project in order to test its accuracy and efficiency. Our project is subjected to seven different types of testing that is generally seen. They are as follows:

5.2.1.1 UNIT TESTING

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

TEST STRATEGY AND APPROACH

Field testing will be performed manually and functional tests will be written in detail.

TEST OBJECTIVES

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.
- Only expected result should be the outcome.

FEATURES TO BE TESTED

- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page.

5.2.1.2 INTEGRATION TESTING

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields.

Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects. The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

Test Results:

All the test cases mentioned above passed successfully. No defects encountered.

5.2.1.3 FUNCTIONAL TEST

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

5.2.1.4 SYSTEM TESTING

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

5.2.1.5 WHITE BOX TESTING

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

5.2.1.6 BLACK BOX TESTING

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot "see" into it. The test provides inputs and responds to outputs without considering how the software works.

5.2.1.7 ACCEPTANCE TESTING

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user.

It also ensures that the system meets the functional requirements.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

5.2.2 TESTING

5.2.2.1 UNIT TESTING

LOGIN

No	Test Case	Expected Output	Observed Output	Result
1	Student Login with registered username and password	Cognitive Test Page should be displayed	Cognitive Test For students is displayed	Pass
2	Student Login with incorrect username and password	Display an error message	Incorrect username or password is displayed	Pass

Table 5.2: Unit Testing-Login

REGISTRATION

No	Test Case	Expected Output	Observed Output	Result
1	Student registration filling all mandatory details	Registration Successful and login page should be displayed	Registration Successful and login page is displayed	Pass
2	Students registration without filling mandatory details	Error message is displayed	Please fill all the mandatory deatails	Pass

Table 5.3: Unit Testing-Registration

5.2.2.2 INTEGRATION TESTING

No	Test Case	Expected Output	Observed Output	Result
1	Student registration successful	Should be able to View available courses and take test	View available courses and take test	Pass
2	Student registration unsuccessful	An error message is displayed	An error message is displayed and cannot view available courses	Pass

Table 5.4: Integration Testing

5.2.2.3 FUNCTIONAL TESTING

No	Test Case	Expected Output	Observed Output	Result
1	Course completed	Take examination	Take examination for corresponding course	Pass
2	Course not completed	Cannot take examination	Error message displayed	Pass

Table no 5.5: Functional Testing

5.3 TEST PLAN

The project is tested to verify its correctness and identify the bugs. The test plan includes the various test cases that acts as the set of conditions or variables that determine whether the corresponding feature in the system is working as it originally established to do so. When this test plan is executed, the errors spotted are rectified and the final testing yields following result.

It cannot confirm system functions properly under all conditions but can establish that it fails under specific conditions.

5.4 TEST ANALYSIS

In this phase of testing, the requirements for software testing are analysed and later its feasibility is determined. In the feasibility study the possibility of project development is found through suitable test cases.

5.5 RESULT

The application is tested and found to function as expected with no errors. This application provides an interface for the users to learn online and the tutors to keep track of his/her students' ability, needs and expectation.

CHAPTER 6

CONCLUSION AND FUTURE ENHANCEMENT

6.1 CONCLUSION

In this project, we presented a new literature-based method that uses tracked data of students' behavior on learning objects to estimate their learning styles. The proposed method has some advantages including time saving, automatic and dynamic learning style detection, and system architecture free. It also refers to all of four dimensions of FSLSM, as well as to their preference levels. Moreover, the good experimental result makes the method become promising and capable for wide use. We also presented architecture suitable for adaptive learning systems and an adaptive LMS with respect to FSLSM. Experiment on learning style identification and adaptation with intelligent agents in the system also had a very good result.

6.2 FUTURE ENHANCEMENT

For the future work, we will carry on extensive tests to firmly validate the proposed system and the efficiency of the method. We also study the ontological representation of learning objects to take its advantages in machine-readable and reasoning capabilities.

APPENDIX A

SAMPLE CODE

home.jsp

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html>

    <head>

        <title>E - Learning</title>

        <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">

    </head>

    <frameset rows="76,*" cols="*" frameborder="NO" border="0" framespacing="0">

    <frame src="top.jsp" name="topFrame" scrolling="NO" noresize>

    <frameset rows="*" cols="115,*" framespacing="0" frameborder="NO" border="0">

    <frame src="left.jsp" name="leftFrame" scrolling="NO" noresize>

    <frame src="right.jsp" name="mainFrame">

    </frameset>
```

```
</frameset>
<noframes>
    <body>
<object classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"
codebase="http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#version=5,0,0,0" width="100" height="23" >
<param name="movie" value="button1.swf">
<param name="quality" value="high">
<embed src="button1.swf" quality="high"
pluginspage="http://www.macromedia.com/shockwave/download/index.cgi?P1_Prod_Version=ShockwaveFlash" type="application/x-shockwave-flash"
width="100" height="23"></embed>
</object>
</body>
</noframes>
</html>
```

AdminMenu.jsp

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>

<!DOCTYPE html>

<%String uname=(String)session.getAttribute("un"); %>

<html><body bgcolor=#FFFFFF><center><h2> Welcome To Admin Page  <font color=blue > <%=uname %></font></h2>

<br><table align='center'><tr><td><a href ='AddAdmin.jsp'>Add An Adiministrator</a></td></tr>

<tr><td></td></tr><tr><td><a href ='AddCourse.jsp'>Add New Course</a></td></tr>

<tr><td></td></tr><tr><td><a href ='DelCourse.jsp'>Delete Course</a></td></tr>

<tr><td></td></tr><tr><td><a href ='ModifyCourse.jsp' >Add Course Details </td></tr>
<tr><td></td></tr><tr><td><a href ='AddExam.jsp' >Add New Exam</a></td></tr>
<tr><td></td></tr><tr><td><a href ='Upload.jsp' >Upload Documents</a></td></tr>
<tr><td></td></tr><tr><td><a href ='View.jsp' >View Pdf</a></td></tr>
<tr><td></td></tr><tr><td><a href="Logout.jsp" target="_parent">Logout</a></td></tr>

</table></center></body></html>
```

AddCourse.jsp

```
<%@page import="Connection.DB"%>

<%@page language="java" import="java.sql.*,java.util.*"%>

  <html><body>

<script type="text/javascript">

function fun2()

{

var c = document.f2.course_name;

var e= document.f2.category;

var d = document.f2.sessions;

if(C_validation(c))

{

if(Fn_validation(e))

{

if(Cn_Validation(d,5))

{

return true;

}
```

```
}

}

return false;

}

function C_validation(uid,mx)

{

var uid_len = uid.value.length;

if (uid_len == 0 )

{

alert("Course Name should not be empty ");

uid.focus();

return false;

}

return true;

}

 function Fn_validation(uid)

{

var uid_len = uid.value.length;
```

```
var regex_pat_alpspace = /^[a-zA-Z\s-]+$/;
if(regex_pat_alpspace.test(uid.value)==true)
   {
      return true
   }
else if (uid_len == 0 )
{
alert("Category Name should not be empty ");
uid.focus();
return false;
}
else if(regex_pat_alpspace.test(uid.value)==false)
   {
      alert("Name Should allow Alphabets,Space and (-) Only");
      return false;
   }
return true;
}
```

```
function Cn_Validation(uzip,mx)
{
var uid_len = uzip.value.length;
var numbers = /^[0-9]+$/;
if(uzip.value.match(numbers) && uid_len<=mx)
{
return true;
}
else if (uid_len == 0)
{
alert('Number of Sessions Should Not be Empty');
uzip.focus();
return false;
}
else if ((uzip.value.match(numbers)) && (uid_len>mx))
{
alert("Sessions Length Should be <= "+mx+"");
uzip.focus();
```

```
return false;
}
else if(uzip.value.match(numbers)!=0)
    {
alert('Sessions must have numeric characters only');
uzip.focus();
return false;
}
}
</script>
<%
 DB Db=new DB();
 Connection con=Db.con;
String method=request.getMethod();
       if(method.equals("GET"))
      {
  %>
```

```
<center><font color=blue><h2>Add a New Course </h2></center><form name="f2" method=POST action="AddCourse.jsp" onSubmit="return fun2()"><table align='center'><tr><td>Enter Course Name </td><td><input type=text  name='course_name'></td></tr><tr><td>Category</td><td><input type=text  name='category'></td></tr><tr><td>Number of Sessions</td><td><input type=text name=sessions></td></tr><tr><td><input type=submit value=Store></td><td><input type=reset value=Clear></td></tr></table></form>

<br><a href=AdminMenu.jsp>Go To Menu </a>

        <%

      }

      else if(method.equals("POST"))

   {

       String coursename = request.getParameter("course_name");

       String sessions = request.getParameter("sessions");

       String category = request.getParameter("category");

        try

        {

                ResultSet rs=Db.Select("select  max(substring(course_id,2,1)) + 1 from courses ");
```

```
            rs.next();
                        String cd=rs.getString(1);
                        if(cd==null)
                           cd="1";
            String cid = "c" +cd;
            PreparedStatement ps = con.prepareStatement("insert into courses
values(?,?,?,?) ");
            ps.setString( 1, cid);
                  ps.setString( 2, coursename);
            ps.setString( 3, category);
            ps.setString( 4, sessions);
            ps.executeUpdate();
            ps.close();
            rs.close();
%>
      <center><font color=blue><h2>Course Added Succesfully</h2><br>
            <font color=red>The Course Id Generated was :</font><%=cid%>
```


Make Sure You Provide Session Beak Up's and Data so that students can make use of Your valuable course

Go To Menu

```
<%
            }
catch(Exception e)
        {
                e.printStackTrace();
        }
        }
%>
</body></html>
```

ModifyCourse.jsp

```
<%@page import="Connection.DB"%>
 <%@page import="java.io.PrintWriter"%>
<%@page language="java" import="java.sql.*,java.util.*"%>
<html><body>
        <script type="text/javascript">
function fun2()
{
var c = document.f2.CID.value;
if(c=="Select")
{
alert("Please Choose Course ID");
document.f2.CID.focus();
return false;
}
}
function fun3(button,ele)
```

```
{
var c = document.f3.session_NO;
var d = document.f3.topic;
if(button.value=="Add")
   {
if(Cn_Validation(c,5))
{
if(C_validation(d,ele))
{
return true;
}
}
}
else if(button.value=="Delete")
   {
      var chks = document.getElementsByName('C');
var checkCount = 0;
for (var i = 0; i < chks.length; i++)
```

```
{
if (chks[i].checked)
{
checkCount++;
}
}
if (checkCount < 1)
{
alert("Please select at least one Checkbox.");
return false;
}
else
    {
return true;
var http = new getXMLHttpRequestObject();
   var url = "ModifyCourse.jsp";
http.open("POST", url, true);
 }
```

```
    return false;
}
 return false;
}
function Cn_Validation(uzip,mx)
{
var uid_len = uzip.value.length;
var numbers = /^[0-9]+$/;
if(uzip.value.match(numbers) && uid_len<=mx)
{
return true;
}
else if (uid_len == 0)
{
alert('Number of Sessions Should Not be Empty');
uzip.focus();
return false;
}
```

```
else if ((uzip.value.match(numbers)) && (uid_len>mx))
{
alert("Sessions Length Should be <= "+mx+"");
uzip.focus();
return false;
}
else if(uzip.value.match(numbers)!=0)
   {
alert('Sessions must have numeric characters only');
uzip.focus();
return false;
}
}
function C_validation(uid,el)
{
var uid_len = uid.value.length;
if (uid_len == 0 )
{
```

```
alert("Course Name should not be empty ");
uid.focus();
return false;
}
else
    {
var http = new getXMLHttpRequestObject();
   var url = "ModifyCourse.jsp";
     url +="?CID="+el;
http.open("POST", url, true);
 http.send(null);
return true;
    }
}
</script>
<%
 DB Db=new DB();
 response.setContentType("text/html");
```

```
 PrintWriter pw=response.getWriter();
String method=request.getMethod();
      if(method.equals("GET"))
      {
%>
<font color=blue><h2><center>Add Course Details</center> </h2><form name="f2" method=POST><br><br><table align='center'><tr><td><b>Choose Course ID:<select name='CID'> <option value="Select">Select</option>
<%
try{
        ResultSet rs = Db.Select("select course_id from courses order by 1");
        while( rs.next())
                {
                  String cid=rs.getString(1);
 %>
        <option><%=cid%> </option><br>
<%
                }
        rs.close();
```

```
%>
        </select></b></td></tr><tr><td><input type=submit  value='--GO--
'></td><tr></table></form></font> <br><a href="AdminMenu.jsp">Go To Menu
</a>
<%
        }catch(Exception e){
                e.printStackTrace();
        }
      }
       else if(method.equals("POST"))
       {
 String cid="";
          if (session.getAttribute("CID") == null)
          {
            cid=request.getParameter("CID");
            session.setAttribute("CID" , request.getParameter("CID"));
             System.out.println("cid"+cid);
          }
          else
```

```
        {
       cid = (String)session.getAttribute("CID");  System.out.println("1cid"+cid);
        session.setAttribute("CID" , cid);
        }
      %>
<font color=blue><h2><center>Add Course Details in the Course
<%=cid%></center></h2><br><br><form name="f3" method="Post"><table
border=1 align='center'><tr color='cyan'><th>Delete/Add</th><th>Session
Number</th><th>Topic</th></tr>
<%
      boolean st = false;
      try{
          String add=request.getParameter("ADD");
     Connection con=Db.con;
          if(  add!= null && ! add.equals("") &&
add.compareToIgnoreCase("Add")==0){
           cid = (String)session.getAttribute("CID");
          System.out.println("1cid"+cid);
```

```
            ResultSet rs=Db.Select("select sessions from courses where
course_id='"+cid+"'");
          if(rs.next())
          {
             System.out.println("coming");
          if(rs.getInt(1)<(Integer.parseInt(request.getParameter("session_NO"))))
          {
             st=false;
            out.println("<center><font color=red>Sorry! Session Number is
Exceed</font></center><br>");
          }
          else
            st=true;
          }
          rs.close();
          if(st==true)
          {
     PreparedStatement ps = con.prepareStatement("insert into course_details
values( ?,?,?) ");
```

```
ps.setString(1 ,cid);

ps.setString(2 ,request.getParameter("session_NO"));

ps.setString(3 ,request.getParameter("topic") );

ps.executeUpdate();

ps.close();

 }

   }

if(  add!= null && ! add.equals("") &&
add.compareToIgnoreCase("Delete")==0){

  String[] values = request.getParameterValues( "C");

   PreparedStatement  ps =con.prepareStatement("delete from course_details
where course_id=? and session_no=?");

     for(int i=0;i<values.length;i++){

        System.out.println("deleting " + values[i]);

       ps.setString( 1 , cid);

   ps.setString(2 , values[i]);

 ps.executeUpdate();

}

ps.close();
```

```
        }
          con.close();
                ResultSet rs = Db.Select("select * from course_details where
course_id='" + cid + "' order by session_no" );
          while( rs.next()){
cid=rs.getString(1);
String sess=rs.getString(2);
String topic=rs.getString(3);
out.println("<tr><td><input type=checkbox name=C value=" + sess +
"></td><td>" + sess + "</td><td><a
href='CourseContent.jsp?cid="+cid+"&cc="+sess+"','CourseContent','toolbar=0,sc
rollbars=1,width=500,height=500'); return false;>" + topic + "</a></td></tr>");
            }
        rs.close();
%>
          <tr><td><input type=submit id="ADD" value="Add" name=ADD
onclick="return fun3(this,'<%=cid%>')"></td><td><input type=text
name=session_NO></td><td><input type=text name=topic></td></tr><tr><td
cellspan=3>
```

```
            <input type=submit id="ADD" value="Delete" name=ADD
onclick="return fun3(this)"></td></tr></table></form></font>
            <br>
            <br>
              <a href="AdminMenu.jsp">Go To Menu</a></h3>
<%
      session.setAttribute("CID" , cid);  cid = (String)session.getAttribute("CID");
        System.out.println("lcid"+cid);
       }catch(Exception e){
        e.printStackTrace();
        }
  }
%>
</body></html>
```

Register.jsp

```
<%@page import="Connection.DB"%>

<%@page import="java.io.*"%>

<%@page language="java" import="java.sql.*,java.util.*"%>

<%

 response.setContentType("text/html");

PrintWriter pw=response.getWriter();

DB Db=new DB();%>

<%!String fname,lname,state,city,zip,country,email,uname,pass;%>

<%String method=request.getMethod();

PreparedStatement ps;

if(method.equals("POST"))

{

   try

    {

        boolean b=readValues(request);

        if(!b)

        {
```

```
%>

<h3><font color=blue>Required Inputs are Insufficient
</font></h3><br><form><input type=button value='Back'
onClick='history.back()'></form>

<%

}

Connection con=Db.con;

                ps=con.prepareStatement("insert into ereg values(?,?,?,?,?,?,?,?,?)");

                ps.setString(1,uname);

                ps.setString(2,pass);

                ps.setString(3,fname);

                ps.setString(4,lname);

                ps.setString(5,city);

                ps.setString(6,state);

                ps.setString(7,country);

                ps.setString(8,zip);

                ps.setString(9,email);

                int i=ps.executeUpdate();

                if(i==1)
```

```
                {
%>
<html><head><title>Registration Successful</title></head>
<body bgcolor=#FFFFFF>
<strong><h1><center><font color=blue>E-
LEARNING</font></center></h1></strong>
<br><h2>Registration Successful </h2>
<font color=blue><h2> Welcome to the New Experience of E-learning
</h2>Your User id For Further Correspondense is <font
color=red><b><%=uname%></b><br>
<br><h3>Please Use This Link to Login <a href='login.jsp'
target="_parent">Login</a></font>
</body></html>
<%
}
            con.close();
}
catch(Exception e)
{
```

```
e.printStackTrace();
}
}
%>
<%!public boolean readValues(HttpServletRequest req)throws Exception
{
fname=req.getParameter("first_name");
lname=req.getParameter("last_name");
state=req.getParameter("state");
city=req.getParameter("city");
zip=req.getParameter("zip");
country=req.getParameter("country");
email=req.getParameter("email");
pass=req.getParameter("password");
uname=req.getParameter("user");
if((fname==null) || fname.equals("") || (zip == null) || zip.equals("") || (pass == null)
|| pass.equals("")||(uname == null) || uname.equals("") )
{
```

```
return false;
}
else
{
return true;
}
}
%>
```

APPENDIX B

SCREENSHOTS

HOME PAGE

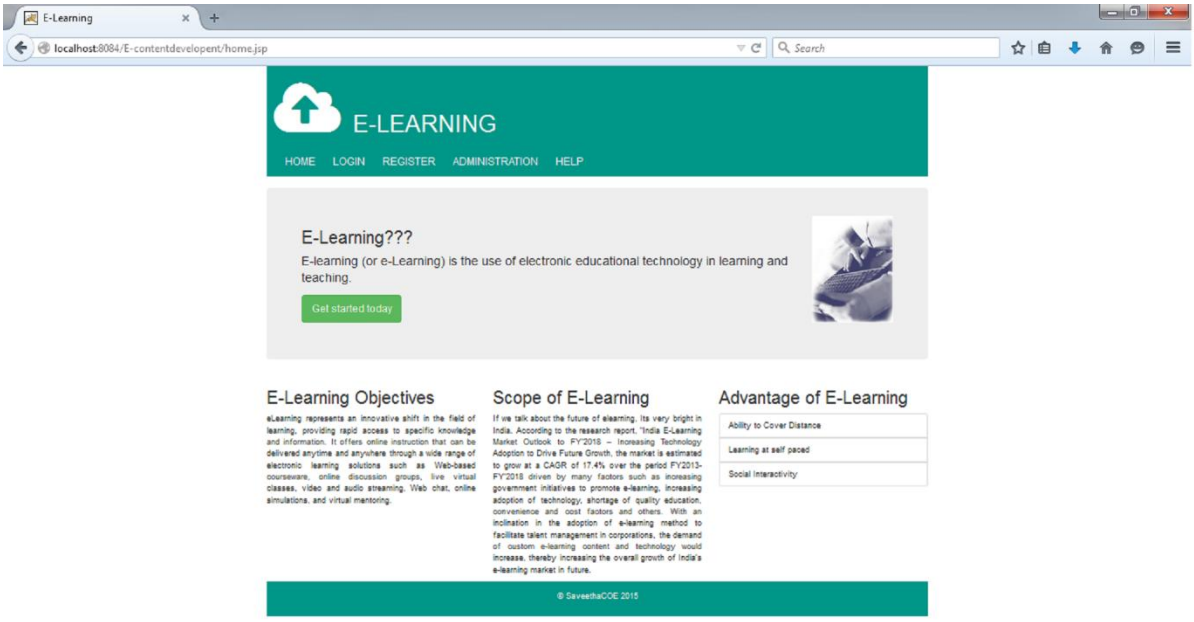


Figure B.1: Homepage

The home page displays the all available options for both students and teachers. The first step will be registration using register button.

REGISTRATION

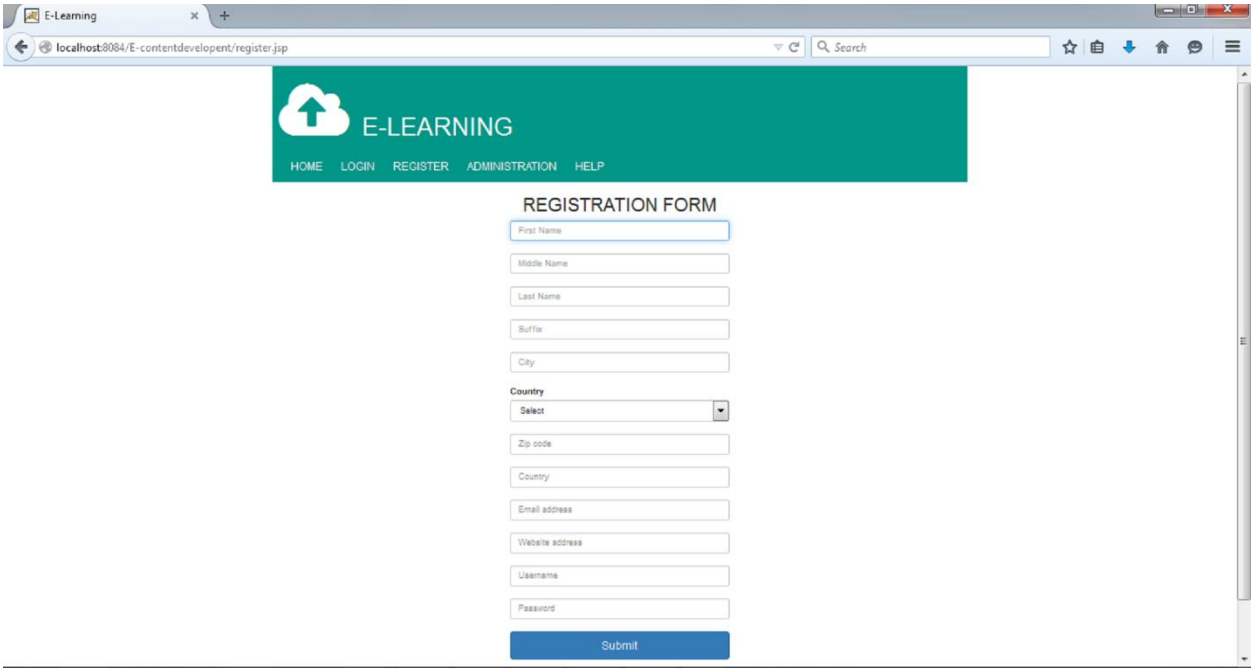


Figure B.2: Registration page

The registration form requires filling up a form with basic details such as name, address. A username and password should be registered. All the mandatory columns should be filled.

STUDENT LOGIN PAGE:

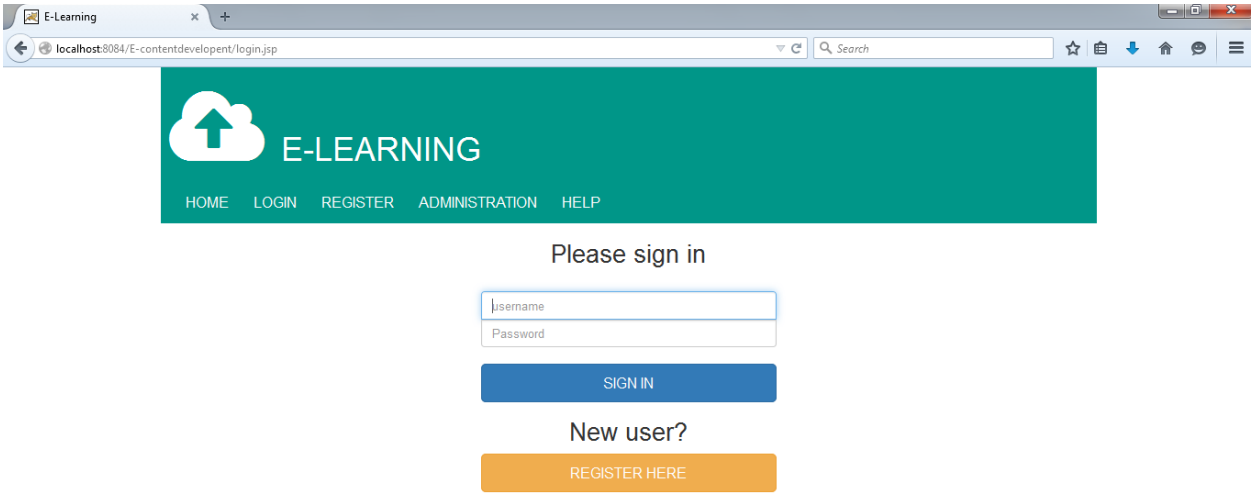


Figure B.3: Student Login Page

Student login page asks for the registered user-id or username and corresponding password using which the student will be authenticated.

COGNITIVE SKILL TEST

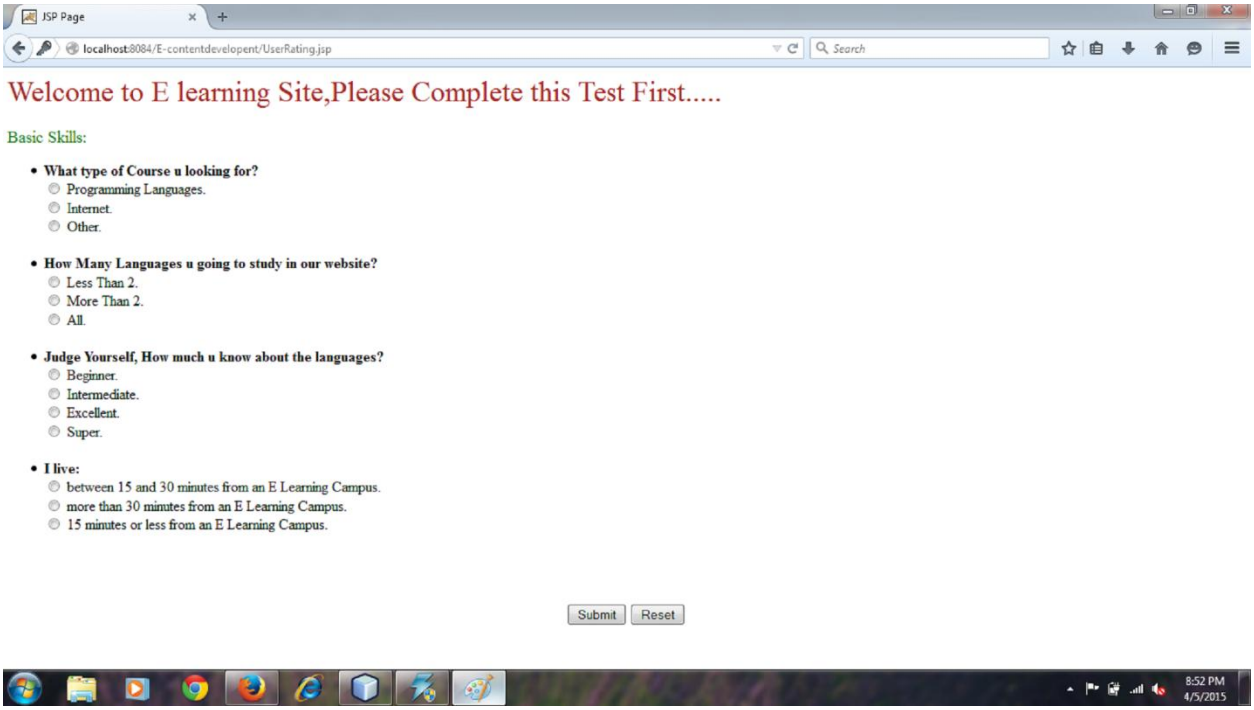


Figure B.4: Cognitive Skill Test

The cognitive skill test, tests the students performance based on their behavior. This test compares the answers of the student and their behaviour during the session and generates a score.

TEST COMPLETION

Figure B.5:Test completion

On completion of the cognitive test the student can view the available courses, learn them and then take up the examination available for that particular course.

COURSE INFORMATION

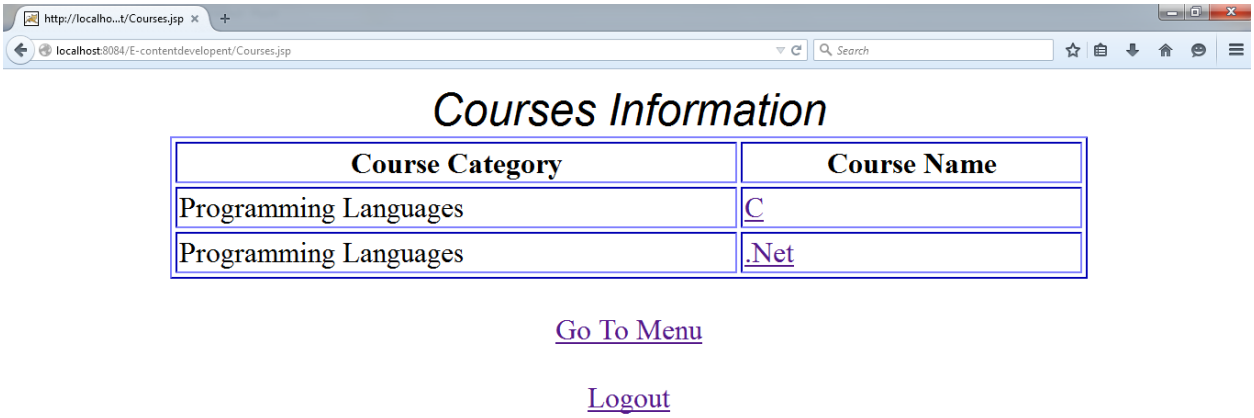


Figure B.6: Course Information

The available courses section displays all the subject courses available for that particular student. On clicking a particular course, the student can view the notes of that particular subject.

TEST RESULTS

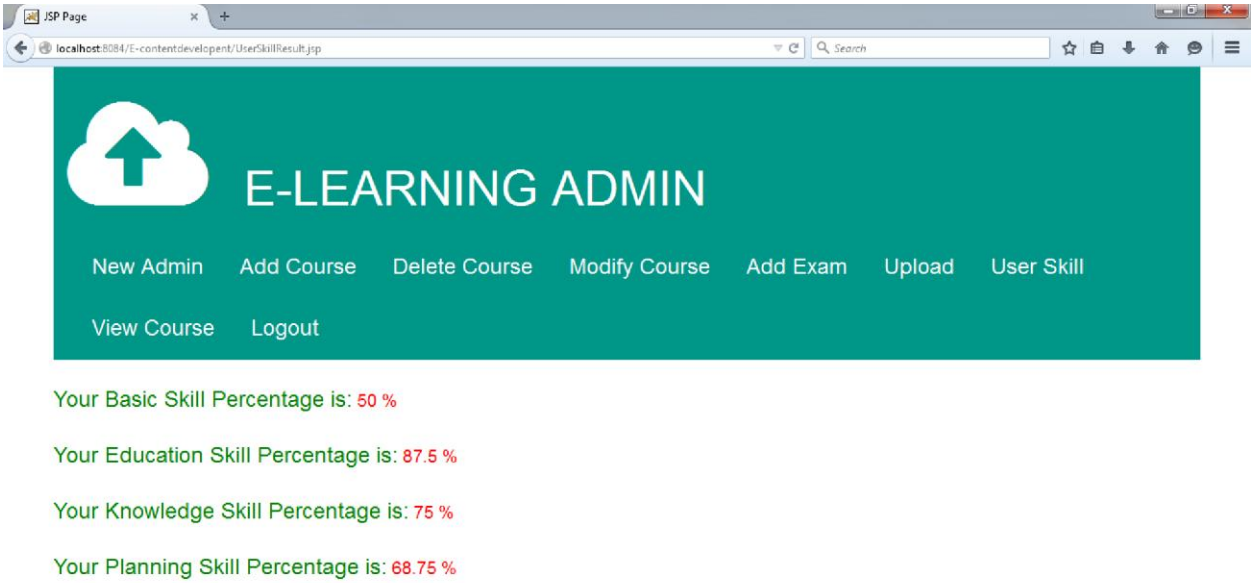


Figure B.7: Test Results

After the completion of the course and tests the admin and students can view the results of the cognitive test.

ADMIN LOGIN PAGE:

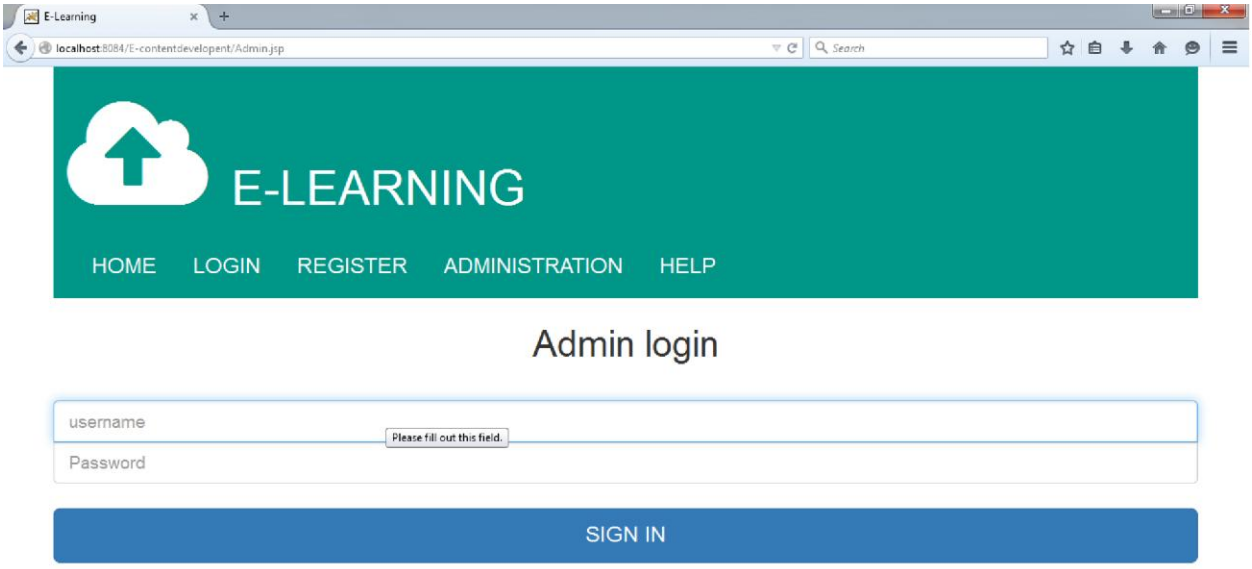


Figure B.8: Admin login page

The admin login requires a username and password to authenticate the faculty member to upload notes, to add examination and to view the test results.

UPLOAD FILE:

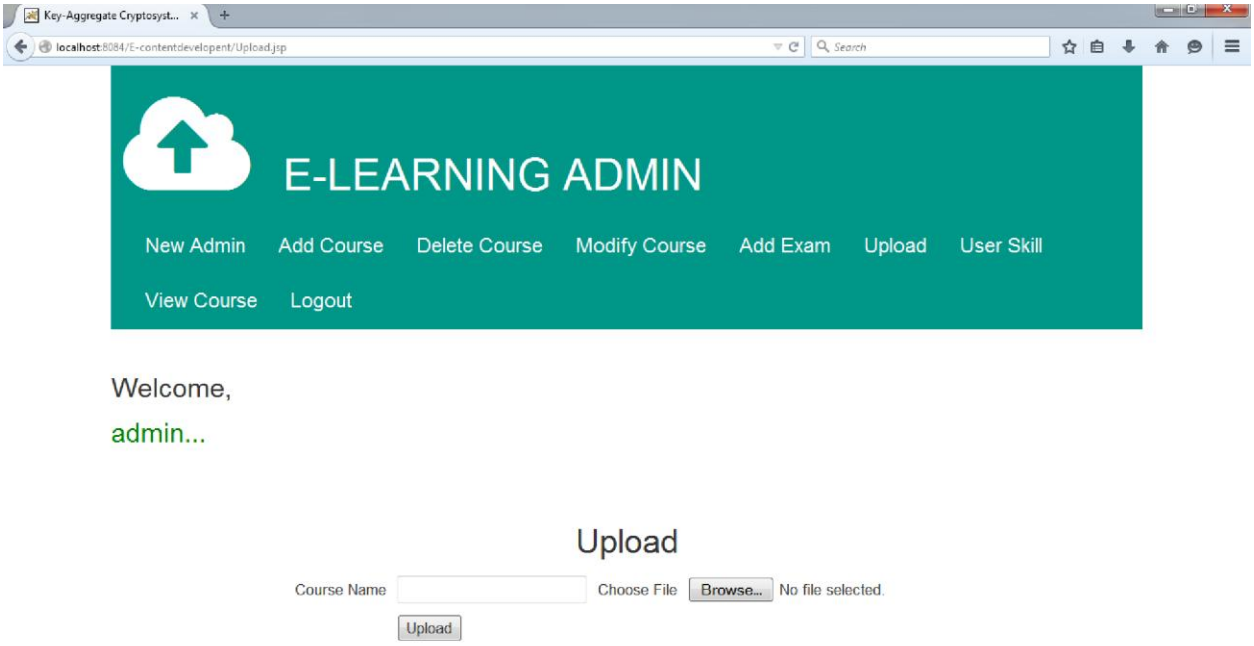


Figure B.9: Upload File

To upload documents into a particular subject the admin needs to specify the course name and upload corresponding documents which can be viewed by the students.

AVAILABLE DOCUMENTS:

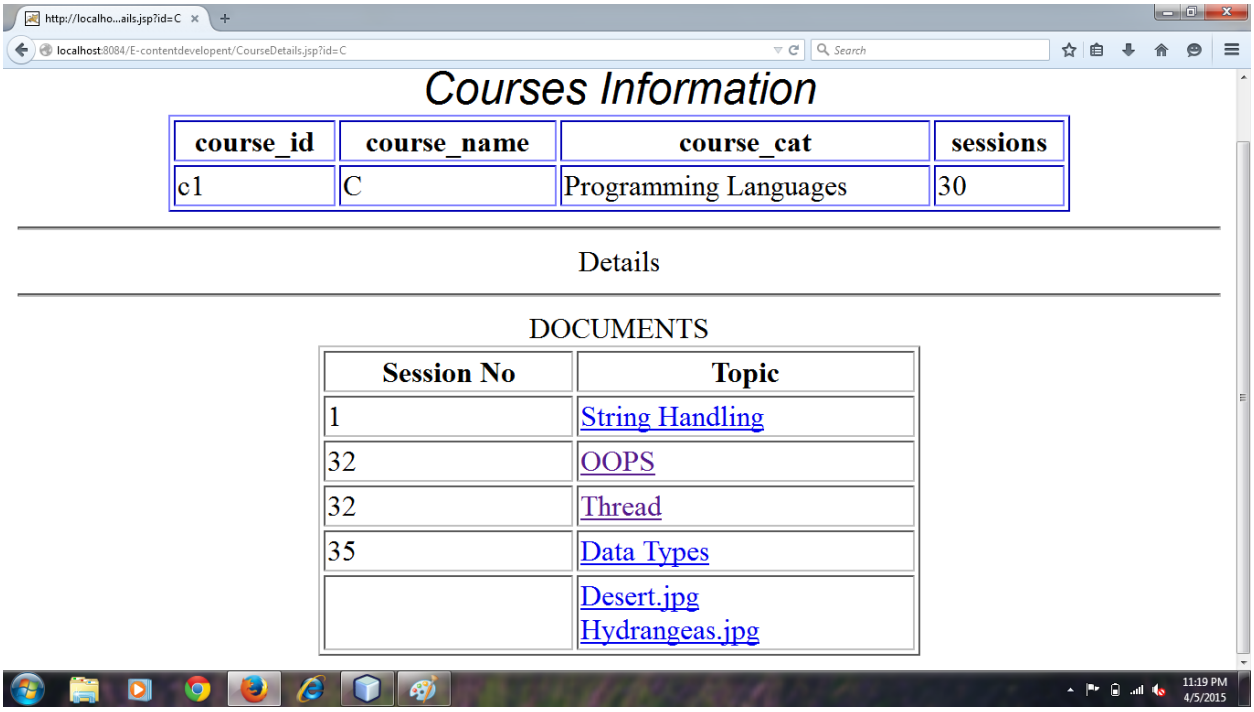


Figure B.10: Available Documents

The course info can be viewed by both the admin and the students. The students can choose the required course the view the notes.

REFERENCES

1. B.A.Soloman and R. M. Felder. 1997. Index of Learning Styles. Available: http://www.engr.ncsu.edu/learningstyles/ilsweb.html.

2. C.I. Peña, J.L. Marzo, and J.L. Rosa. 2005. Intelligent Agents to Improve Adaptivity in A Web-Based Learning Environment. StudFuzz 178, 2005, pp. 141–170.

3. D. A. Wiley. 2000. Connecting learning objects to instructional design theory: A definition, a metaphor, and a taxonomy. Available: http://wesrac.usc.edu/wired/bldg-7_file/wiley.pdf.

4. E. Popescu, P. Trigano and C. Badica. 2008. Relations between Learning Style and Learner Behavior in an Educational Hypermedia System: an Exploratory Study. In Proceedings of the 8th IEEE International Conference on Advanced Learning Technologies, (ICALT'08), pp.725- 726.

5. Hong H. & Kinshuk. 2004. Adaptation to student learning styles in web based educational systems. In Proceedings of ED-MEDIA 2004, World Conference on Educational Multimedia, Hypermedia & Telecommunications, pp. 491- 496.

6. H. J. Cha, Y.S. Kim, S.H. Park, T.B. Yoon, Y.M. Jung, and J.-H. Lee. 2006. Learning Style Diagnosis Based on User Interface Behavior for the Customization of Learning Interfaces in an Intelligent Tutoring System. In Proceedings of the 8th International Conference on Intelligent Tutoring Systems, Lecture Notes in Computer Science, Springer-Verlag Berlin Heidelberg, Vol. 4053, 2006, pp. 513-524.

7. J. W. Keefe. 1979. Learning Style: An Overview. In J.W. Keefe (ed.), Student Learning Styles: Diagnosing and Prescribing Programs, Reston, VA.: National Association of Secondary School Principals.

8. James, W. B. & Gardner, D. L. (1995). Learning styles: Implications for distance learning. ERIC Document Reproduction Service No. EJ 514 356.

9. LOM 2002. IEEE 1484.12.1-2002. Draft Standard for Learning Object Metadata v.1. Available: http://ltsc.ieee.org/wg12/files/LOM_1484_12_1_v1_Final_Draft.pdf.

10. Merriam, S. B., and Caffarella, R. S. 1991. Learning in Adulthood. San Francisco, CA: Jossey-Bass.