UML Diagrams on Online Academic Portal

April 2023

UML INTRODUCTION :
A software development method consists of a modeling language and a process. The Unified Modeling Language (UML) is called a modeling language, not a method. The modeling language is the notation that methods use to express designs. The process describes the steps taken in doing a design. The Unified Modeling Language (UML) is developed as a graphical language for visualizing, specifying, constructing, and documenting the artifacts of a software-intensive system. In the UML, the five main views of the system are

- User
- Structural
- Behavioral
- Implementation
- Environment

1 STRUCTURAL VIEW:

1.1 Class Diagram

1.2 Object Diagram

2 IMPLEMENTATION VIEW:

2.1 Component Diagram

3 BEHAVIOUR VIEW:

3.1 Activity Diagram

3.2 Sequence Diagram

3.3 Collaboration Diagram

3.4 Use Case Diagram

4 ENVIRONMENT VIEW:

4.1 Deployment Diagram

Aim : To Draw a Class Diagram for Online Academic Portal
Description :
The class diagram includes the following classes:

1. User: represents a user of the system, who can log in, log out, and has an ID, name, email, and password.
2. Student: represents a user who is enrolled in courses, and has a list of enrolled courses. Students can enroll and unenroll in courses.
3. Course: represents a course, with a code, name, instructor, and a list of enrolled students. Courses can have students added or removed from their list of enrolled students.
4. Material: represents a material (such as a document, video, or other resource) that is associated with a course. Materials have a title, description, file URL, and are uploaded by a user (either a student or an instructor).
5. Announcement: represents an announcement that is posted to a course's announcement board. Announcements have a title, description, and are posted by a user (either a student or an instructor) on a certain date.

1.1 Class Diagram

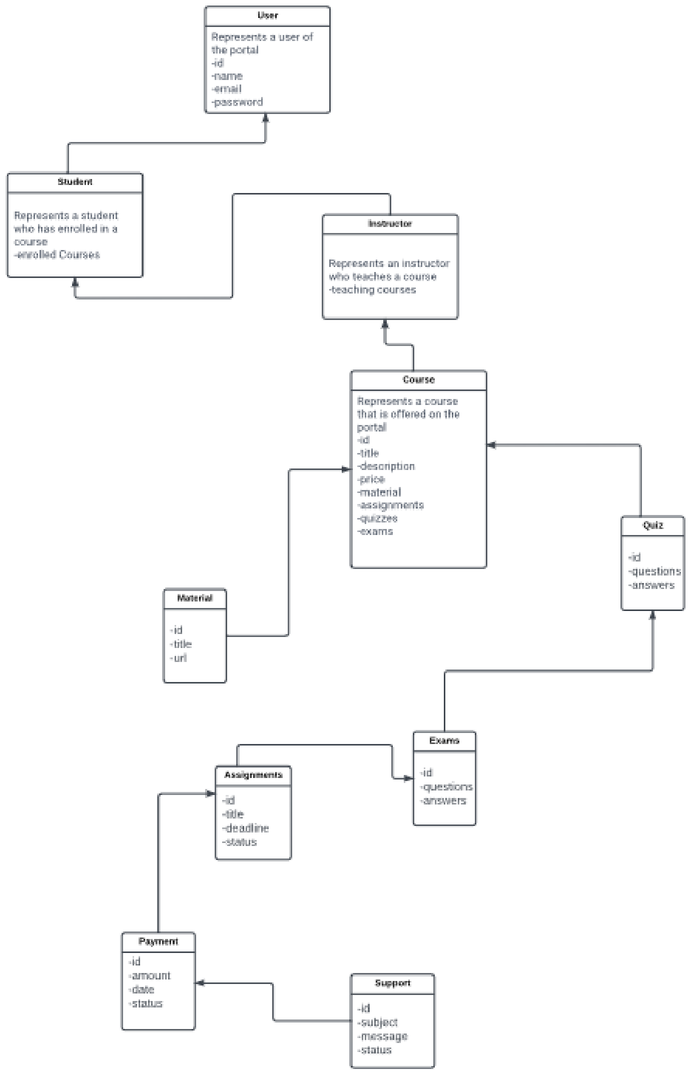


Figure 1: SCREENSHOT OF CLASS DIAGRAM

Aim : To Draw a State Diagram for Online Academic Portal
Description :
The state diagram includes the following states:

1. User: represents the user of the system, who can be either logged in or logged out.
2. Student: represents a user who is enrolled in courses, who can be either enrolled or unenrolled.
3. Course: represents a course, which can be either open (enrollment is allowed) or closed (enrollment is not allowed).

   The transitions between states are as follows:

4. A logged out user can become logged in by logging in with their username and password.
5. A logged in user can become logged out by logging out of the system.
6. An enrolled student can become unenrolled by dropping the course.
7. An unenrolled student can become enrolled by enrolling in a course.
8. An open course can become closed when the enrollment period ends, or if the instructor chooses to close the course early.
9. A closed course can become open if the instructor chooses to reopen the course.

1.2 State Diagram

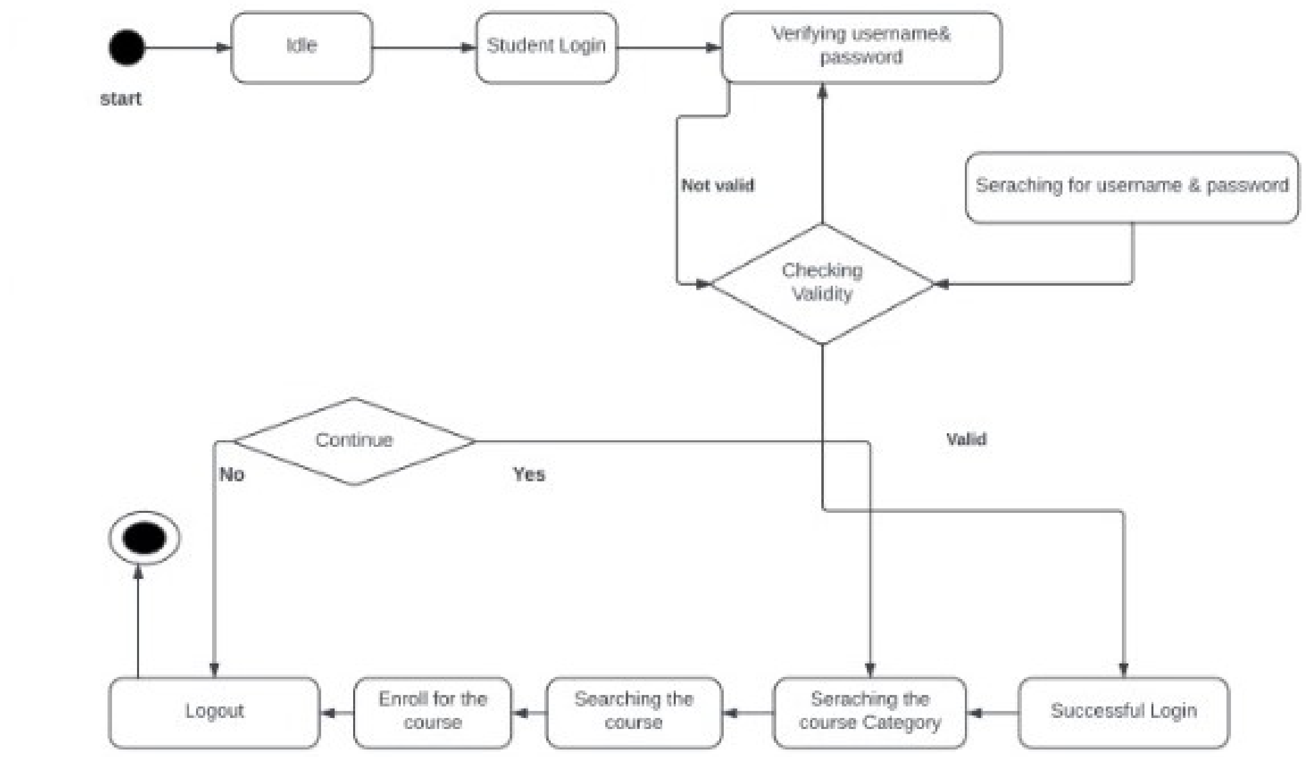


Figure 2: SCREENSHOT OF STATE DIAGRAM

Aim : To Draw a Component Diagram for Online Academic Portal
Description :
The component diagram includes the following classes:

1. User Interface: represents the interface through which users interact with the system. It includes the login screen, dashboard, course page, material page, and announcement board.
2. Server: represents the server-side components of the system. It includes the database, course manager, user manager, announcement manager, and authentication manager.
3. Controller: represents the components that handle user input and communicate with the model and server components. It includes the login controller, dashboard controller, course controller, material controller, and announcement controller.
4. Model: represents the components that store and manipulate data. It includes the course, user, material, and announcement classes.

The connections between components are as follows:

1. The User Interface communicates with the Controller components to handle user input and display information to the user.
2. The Controller components communicate with the Model components to retrieve and manipulate data.
3. The Controller components communicate with the Server components to send and receive data from the server-side components.
4. The Server components communicate with the Model components to store and retrieve data from the database.

2.1 Component Diagram

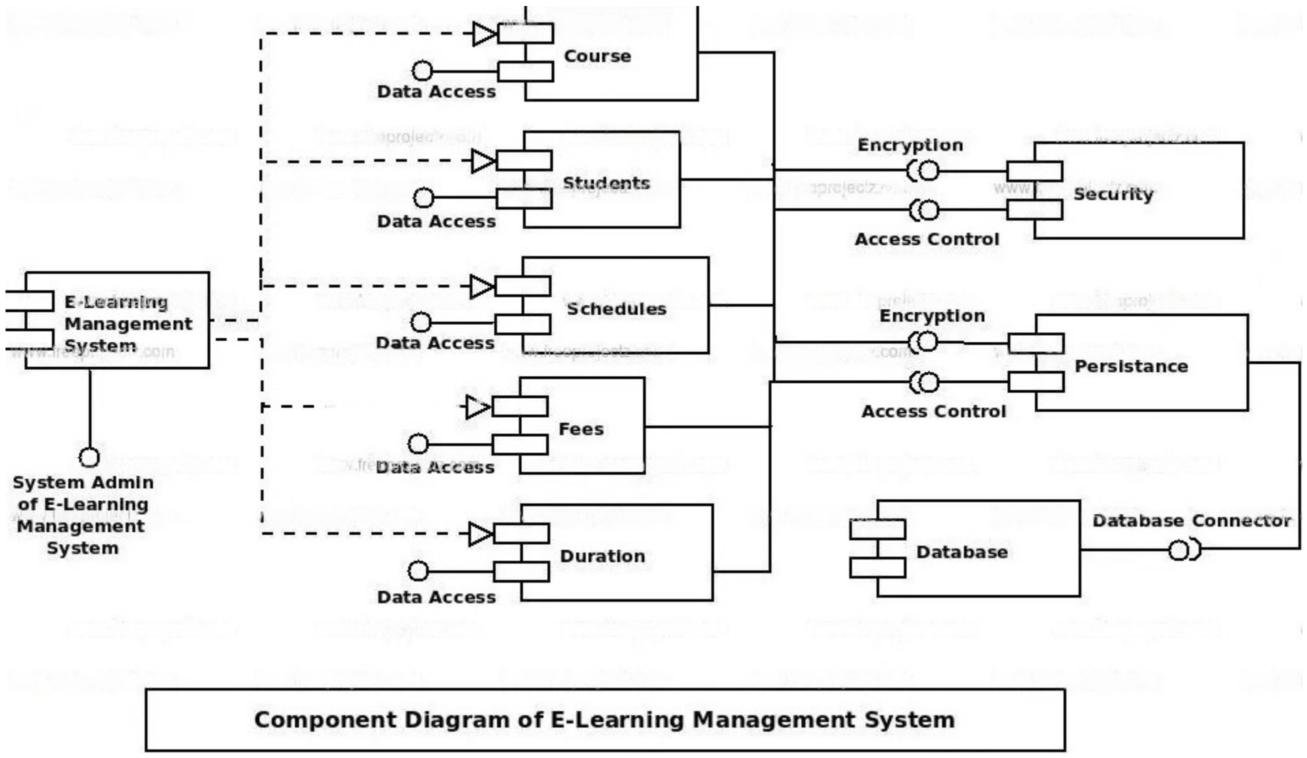


Figure 3: SCREENSHOT OF COMPONENT DIAGRAM

Aim : To Draw a Activity Diagram for Online Academic Portal
Description :
The Activity diagram includes the following classes:

1. Start Activity: represents the beginning of the enrollment process.
2. View Courses: represents the activity of browsing the available courses.
3. Select Course: represents the activity of choosing a specific course to enroll in.
4. Review Details: represents the activity of reviewing the course details, such as the syllabus, schedule, and instructor information.
5. Confirm Enroll: represents the activity of confirming the enrollment in the chosen course.
6. Finish Activity: represents the end of the enrollment process.

The transitions between activities are as follows

1. The Start Activity leads to the View Courses activity.
2. The View Courses activity can lead to the Select Course activity, or it can be canceled to return to the Start Activity.
3. The Select Course activity leads to the Review Details activity.
4. The Review Details activity can lead to the Confirm Enroll activity, or it can be canceled to return to the Select Course activity.
5. The Confirm Enroll activity leads to the Finish Activity, which marks the completion of the enrollment process.

3.1 Activity Diagram

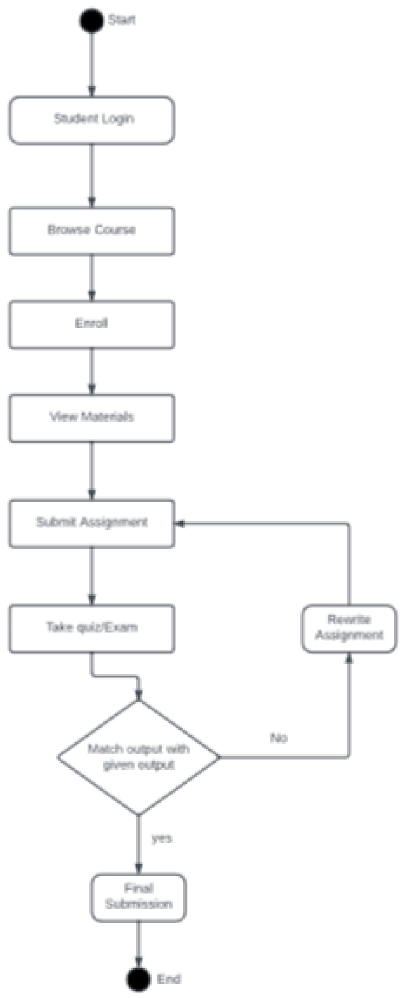


Figure 4: SCREENSHOT OF ACTIVITY DIAGRAM

Aim : To Draw a Sequence Diagram for Online Academic Portal

Description :
The Sequence diagram includes the following classes:

1. The User sends a request to start a search for a course.
2. The Server receives the request and starts searching for courses.
3. The Server filters the results based on any search criteria.
4. The Server sends the search results back to the User.
5. The User views the search results and selects a course to view.
6. The User sends a request to view the course details.
7. The Server receives the request and retrieves the course information.
8. The Server sends the course information back to the User.
9. The User views the course details.

3.2 Sequence Diagram

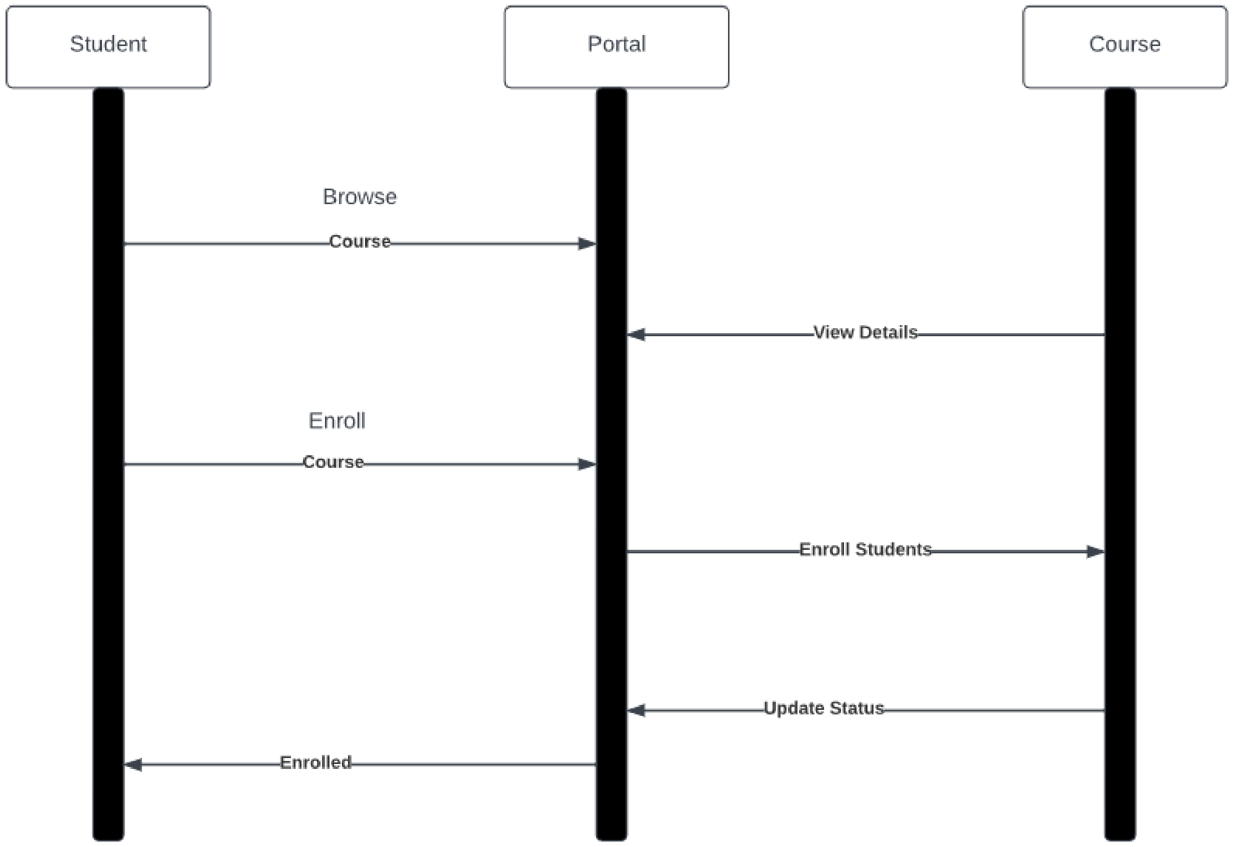


Figure 5: SCREENSHOT OF SEQUENCE DIAGRAM

Aim :
To Draw a Use case Diagram for Online Academic Portal
Description :
The Use Case diagram includes the following classes:

1. Search Course: allows the user to search for courses based on keywords or specific criteria.
2. View Course Info: allows the user to view detailed information about a specific course.
3. Enroll in Course: allows the user to enroll in a course they are interested in.
4. Cancel Course Enrollment: allows the user to cancel their enrollment in a course.
5. View Profile: allows the user to view their personal profile, which may include their name, contact information, and course history.
6. Edit Profile: allows the user to edit their personal profile information.
7. Logout: allows the user to log out of the online portal.

3.3 Use Case Diagram

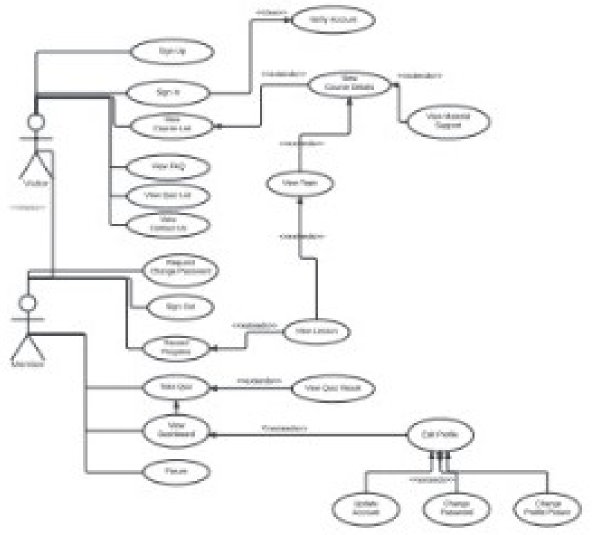


Figure 6: SCREENSHOT OF USECASE DIAGRAM

Aim : : To Draw a Deployment Diagram for Online Academic Portal
Description :
The Deployment diagram includes the following classes:

1. Web Server: represents the hardware node responsible for hosting the web server that serves the online academic portal.
2. Web App: represents the software component responsible for handling user requests and displaying information to the user.
3. Database Server: represents the hardware node responsible for hosting the database server that stores data related to the online academic portal, such as user information and course information.
4. Database: represents the software component responsible for storing and retrieving data from the database server.

4.1 Deployment Diagram

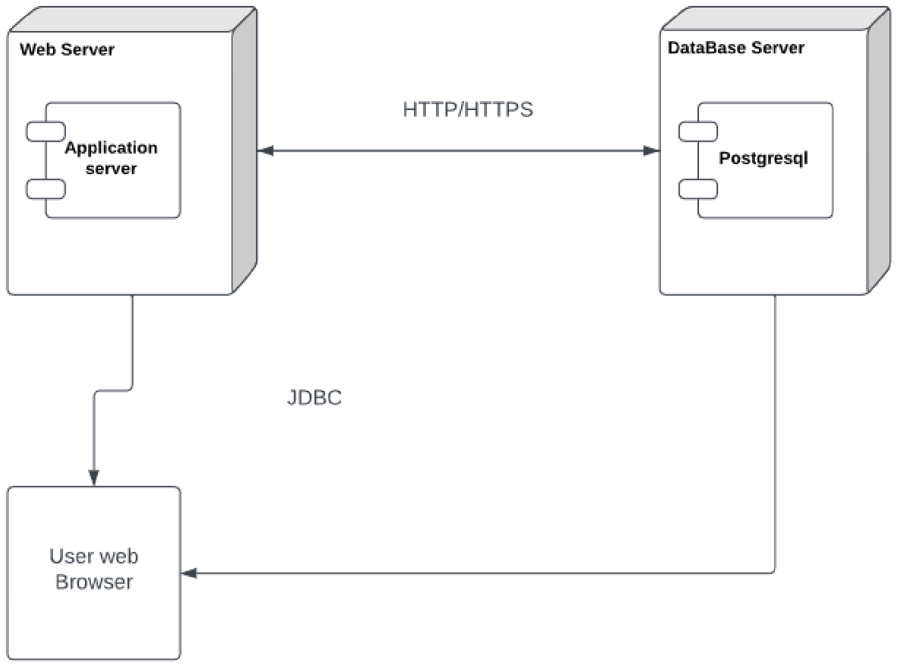


Figure 7: SCREENSHOT OF DEPLOYMENT DIAGRAM

COCOMO MODEL FOR ONLINE ACADEMIC PORTAL

AIM: Online Academic Portal using COCOMO estimation effort.

DESCRIPTION: Boehm proposed COCOMO (Constructive Cost Estima- tion Model) in 1981.COCOMO is one of the most generally used software estimation models in the world. COCOMO predicts the efforts and schedule of a software product based on the size of the software.

4.2 The necessary steps in this model are:

1. Get an initial estimate of the development effort from evaluation of thousands of delivered lines of source code(KDLOC).
2. Determine a set of 15 multiplying factors from various attributes of the project.
3. Calculate the effort estimate by multiplying the initial estimate with all the multiplying factors i.e., multiply the values in step1 and step2.The initial estimate (also called nominal estimate) is determined by an equation of the form used in the static single variable models, using KDLOC as the measure of the size. To determine the initial effort in person-months the equation used is of the type is shown below

   Ei=a*(KDLOC)b The value of the constant a and b are depends on the project type.

In COCOMO, projects are categorized into three types:

1. Organic
2. Semidetached
3. Embedded

1. Organic: A development project can be treated of the organic type, if the project deals with developing a well-understood application program, the size of the development team is reasonably small, and the team members are experienced in developing similar methods of projects. Examples of this type of projects are simple business systems, simple inventory management systems, and data processing systems.

2. Semidetached: A development project can be treated of the organic type, if the project deals with developing a well-understood application program, the size of the development team is reasonably small, and the team members are experienced in developing similar methods of projects. Examples of this type of projects are simple business systems, simple inventory management systems, and data processing systems.
3. Embedded: A development project is treated to be of an embedded type, if the software being developed is strongly coupled to complex hard- ware, or if the stringent regulations on the operational method exist.

For Example: ATM, Air Traffic control.

For three product categories, Bohem provides a different set of expression to predict effort (in a unit of person month)and development time from the size of estimation in KLOC(Kilo Line of code) efforts estimation takes into account the productivity loss due to holidays, weekly off, coffee breaks, etc.

According to Boehm, software cost estimation should be done through three stages:

1. Basic Model
2. Intermediate MOdel
3. Detalied Model

4.2.1 Basic COCOMO Model:

The basic COCOMO model provide an accurate size of the project parameters. The following expressions give the basic COCOMO estimation model:

- Effort=a1*(KLOC) a2 PM
- Tdev=b1*(efforts)b2 Months

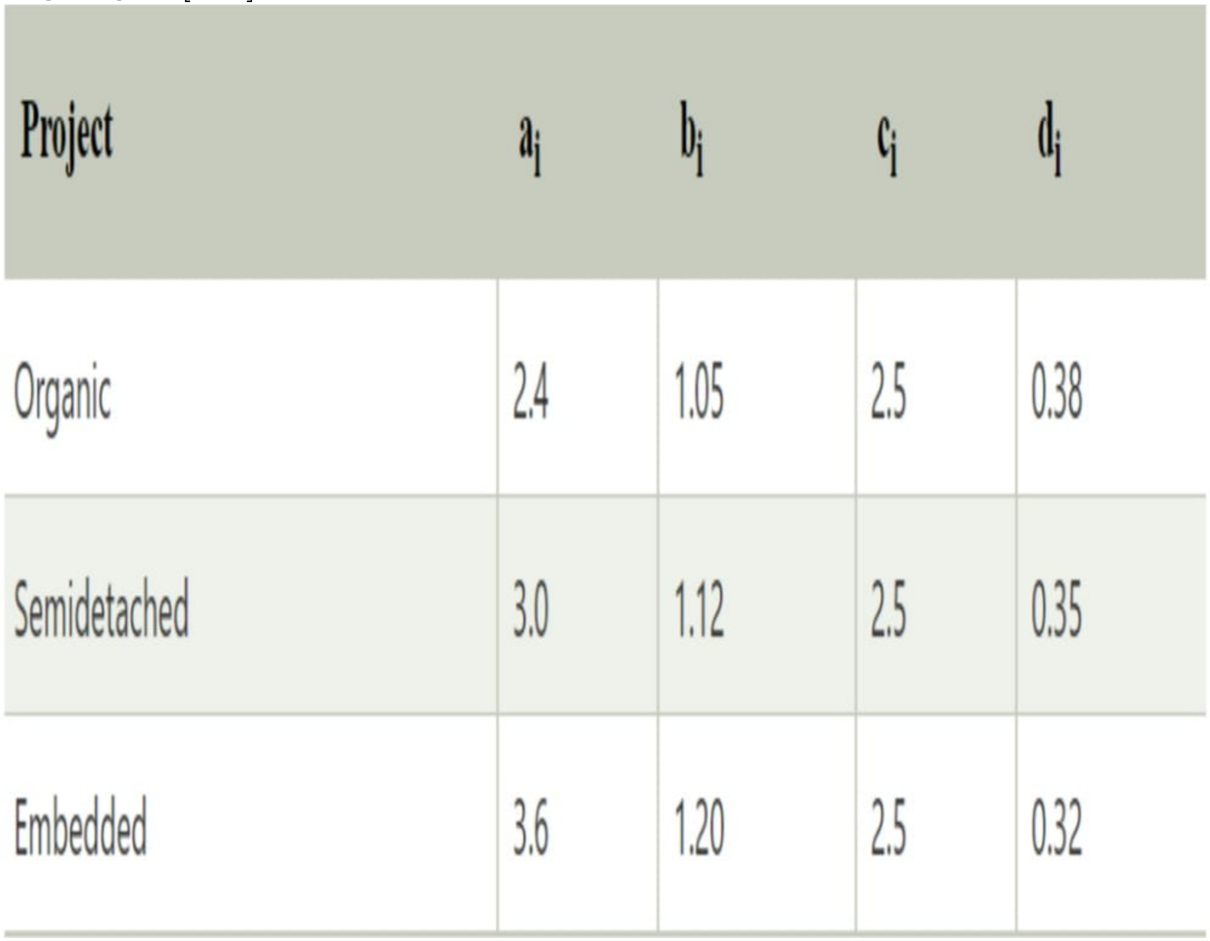
Where,KLOC is the estimated size of the software product indicate in Kilo Lines of Code,a1,a2,b1,b2 are constants for each group of software prod- ucts,Tdev is the estimated time to develop the software, expressed in months,Effort is the total effort required to develop the software product, expressed in per- son months (PMs). Estimation of development effort For the three classes of software products, the formulas for estimating the effort based on the code size are shown below:

- Organic: Effort = 2.4(KLOC) 1.05 PM
- Semi-detached: Effort = 3.0(KLOC) 1.12 PM
- Embedded: Effort = 3.6(KLOC) 1.20 PM

4.2.2 Intermediate Model:

The basic Cocomo model assumes that the effort is only a function of the number of lines of code and some constants evaluated according to the different software systems. However, in reality, no system's effort and schedule can be solely calculated on the basis of Lines of Code. For that, various other factors such as reliability, experience, Capability. These factors are known as Cost Drivers and the Intermediate Model utilizes 15 such drivers for cost estimation.

beginfigure[htp]



Project	a_i	b_i	c_i	d_i
Organic	2.4	1.05	2.5	0.38
Semidetached	3.0	1.12	2.5	0.35
Embedded	3.6	1.20	2.5	0.32

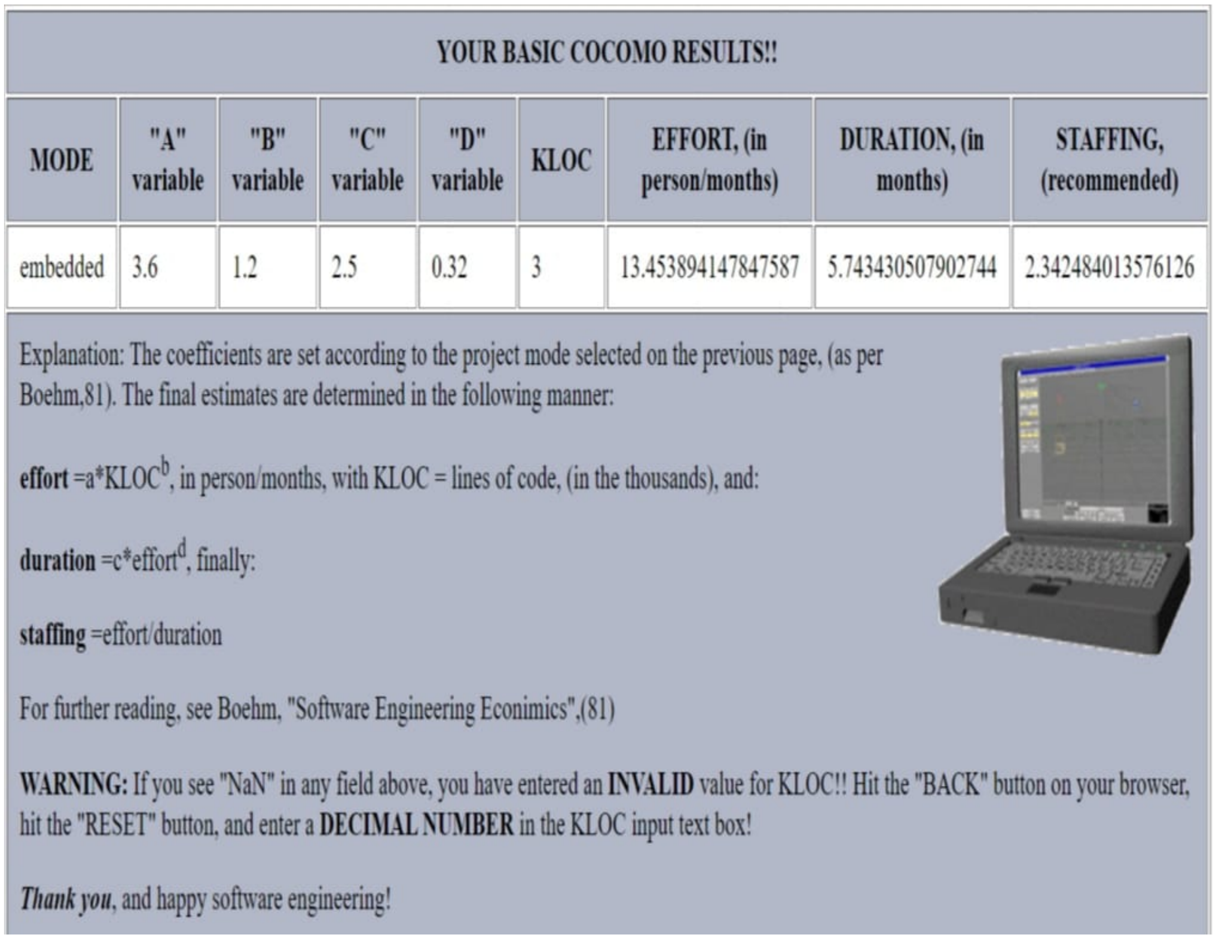
4.2.3 Detailed COCOMO Model:

Detailed COCOMO incorporates all qualities of the standard version with an assessment of the cost driver?s effect on each method of the software engi- neering process. The detailed model uses various effort multipliers for each cost driver property. In detailed cocomo, the whole software is differentiated into multiple modules, and then we apply COCOMO in various modules to estimate effort and then sum the effort.

The Six phases of detailed COCOMO are:

1. Planning and requirements
2. System structure
3. Complete structure
4. Module code and test
5. Integration and test
6. Cost Constructive model

The effort is determined as a function of program estimate, and a set of cost drivers are given according to every phase of the software lifecycle.


YOUR BASIC COCOMO RESULTS!!

MODE	"A" variable	"B" variable	"C" variable	"D" variable	KLOC	EFFORT, (in person/months)	DURATION, (in months)	STAFFING, (recommended)
embedded	3.6	1.2	2.5	0.32	3	13.453894147847587	5.743430507902744	2.342484013576126

Explanation: The coefficients are set according to the project mode selected on the previous page, (as per Boehm,81). The final estimates are determined in the following manner:

effort =a*KLOC^{b}, in person/months, with KLOC = lines of code, (in the thousands), and:

duration =c*effort^{d}, finally:

staffing =effort/duration

For further reading, see Boehm, "Software Engineering Econimics",(81)

WARNING: If you see "NaN" in any field above, you have entered an **INVALID** value for KLOC!! Hit the "BACK" button on your browser, hit the "RESET" button, and enter a **DECIMAL NUMBER** in the KLOC input text box!

Thank you, and happy software engineering!

DATA FLOW DIAGRAM FOR ONLINE ACADEMIC PORTAL

Aim : To Draw a Data Flow Diagram for Online Academic Portal

Description : A Data Flow Diagram (DFD) can be used to model an online academic portal, which is a web-based platform designed to provide educational resources and services to students, teachers, and administrators.

At the highest level, the online academic portal system can be represented as a single bubble, with arrows pointing towards and away from it, representing the flow of data between the system and external entities such as students, teachers, and administrators.

The Context Diagram of an online academic portal system would show the major external entities that interact with the system, such as students and teachers, and the major processes that the system performs, such as managing course content, conducting online assessments, and generating reports.

At the next level, a DFD can be used to break down each major process into sub-processes or modules, each with its own input and output data flows. For example, the process of managing course content can be further broken down into modules such as uploading course materials, creating quizzes and assignments, and managing student progress.

Each of these modules can be represented as bubbles in the DFD, with arrows pointing to and from them representing the flow of data between the modules and external entities. For example, a student may interact with the system by logging in and accessing course materials, while a teacher may interact with the system by creating quizzes and assignments.

DATA FLOW DIAGRAM FOR ONLINE ACADEMIC PORTAL

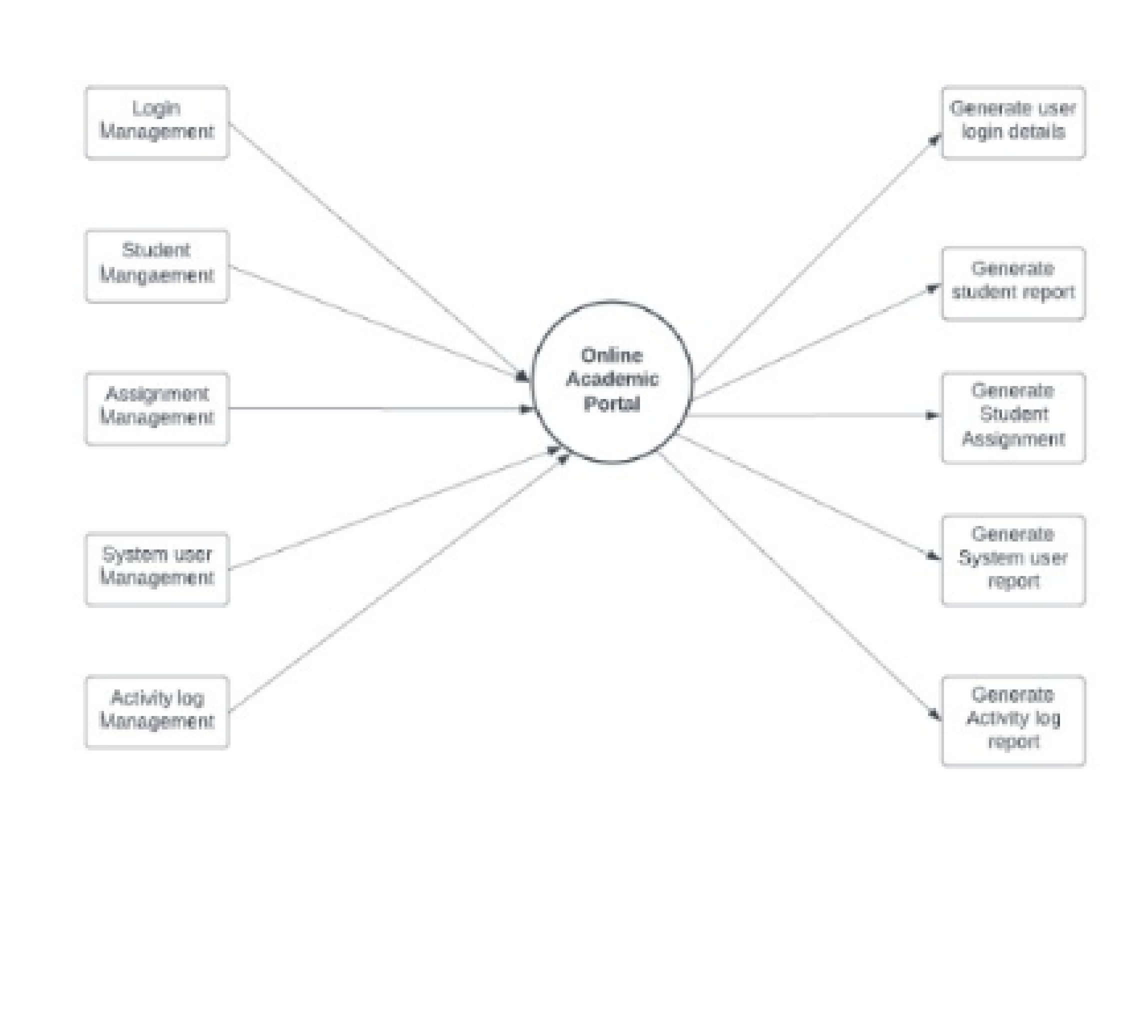


Figure 8: SCREENSHOT OF DFD DIAGRAM

CONTEXT FLOW DIAGRAM FOR ONLINE ACADEMIC PORTAL

Aim : To Draw a Context Flow Diagram for Online Academic Portal

Description : CFD stands for Context Flow Diagram, which is a type of diagram used to represent the flow of information in a system, similar to a Data Flow Diagram (DFD). However, unlike DFDs, CFDs focus on the context or environment in which a system operates, rather than the details of the system itself.

For an online academic portal, a CFD would show the major external entities that interact with the system, such as students, teachers, and administrators, as well as the major information flows between them. It would also show any external systems or data sources that the online academic portal system depends on or interfaces with, such as external databases or learning management systems.

The Context Flow Diagram of an online academic portal system would start with a single bubble representing the system itself, with arrows pointing towards and away from it representing the flow of data between the system and external entities.

CONTEXT FLOW DIAGRAM FOR ONLINE ACADEMIC PORTAL

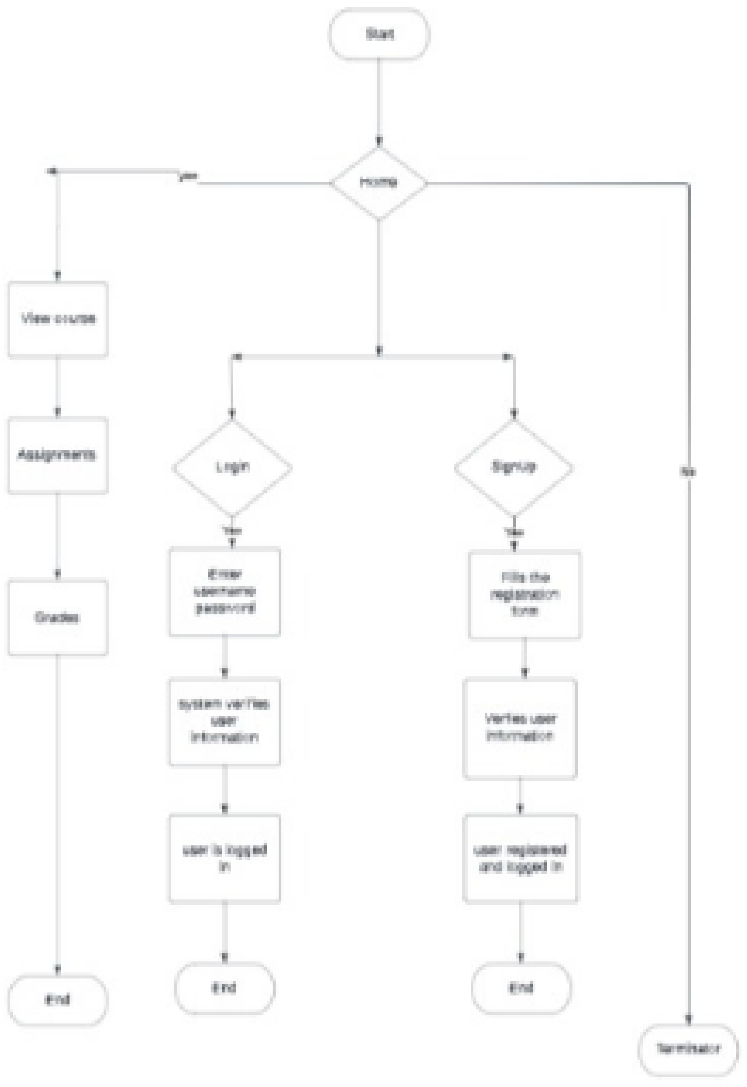


Figure 9: SCREENSHOT OF CFD DIAGRAM

ENTITY DIAGRAM FOR ONLINE ACADEMIC PORTAL

Aim : To Draw an Entity Flow Diagram for Online Academic Portal

Description : The Entity-Relationship (E-R) model is a conceptual model used to represent the relationships between entities in a system. It can be used to model an online academic portal, which is a web-based platform designed to provide educational resources and services to students, teachers, and administrators.

In an E-R model for an online academic portal, the entities would include students, teachers, courses, assignments, quizzes, and other relevant entities. Each entity would be represented by a rectangle, with the name of the entity inside the rectangle.

The relationships between the entities would be represented by lines connecting the rectangles. For example, a student entity would be connected to a course entity by a line representing the enrollment relationship, indicating that a student can enroll in a course. Similarly, a course entity would be connected to an assignment entity by a line representing the assignment relationship, indicating that a course can have multiple assignments.

ER DIAGRAM FOR ONLINE ACADEMIC PORTAL

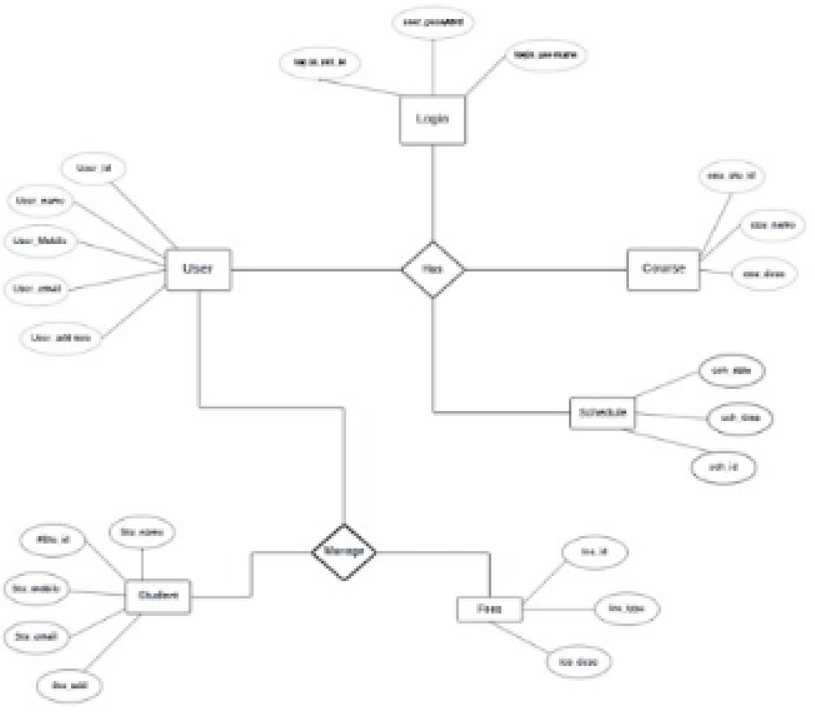


Figure 10: SCREENSHOT OF ER DIAGRAM

STRUCTURED DIAGRAM FOR ONLINE ACADEMIC PORTAL

Aim : To Draw a Structured Diagram for Online Academic Portal

Description : A structured diagram is a type of diagram that shows the hierarchical structure of a system, breaking it down into smaller and more manageable components. It can be used to model an online academic portal, which is a web-based platform designed to provide educational resources and services to students, teachers, and administrators.

A structured diagram for an online academic portal would start with the highest level of the system, which could be represented by a single box. The box would contain the name of the system and any high-level functions it performs, such as managing courses and assignments.

STRUCTURED DIAGRAM FOR ONLINE ACADEMIC PORTAL

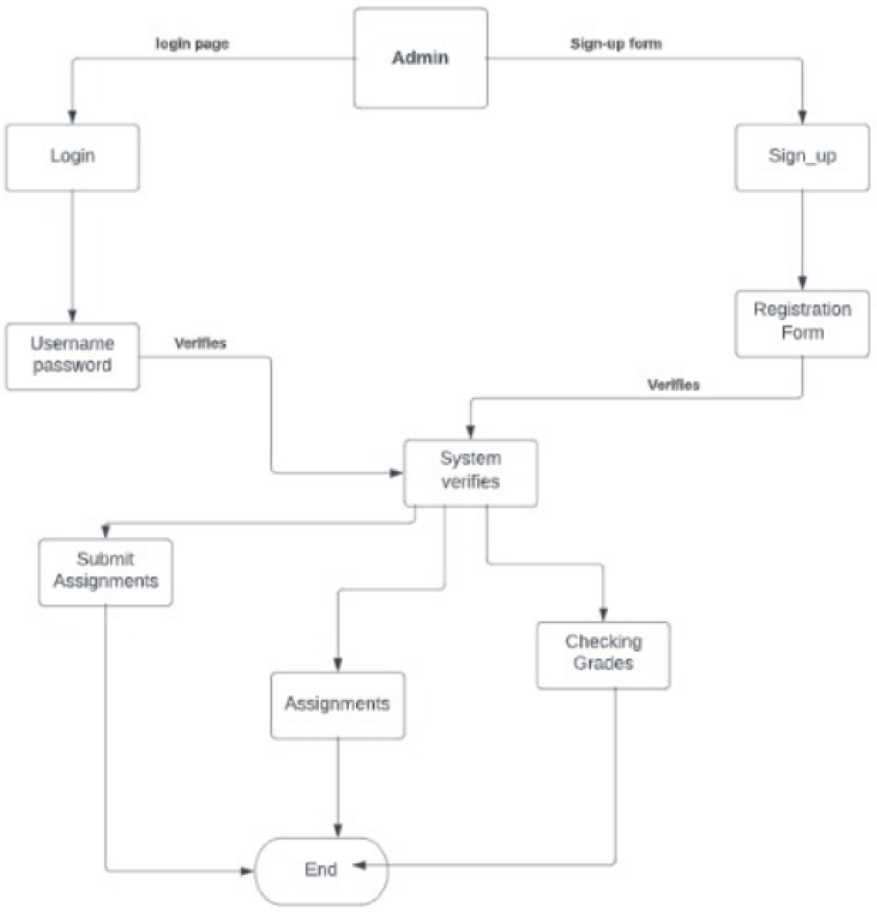


Figure 11: SCREENSHOT OF STRUCTURE DIAGRAM