Karunya INSTITUTE OF TECHNOLOGY AND SCIENCES
(Declared as Deemed to be University under Sec. 3 of the UGC Act. 1956)
A CHRISTIAN MINORITY RESIDENTIAL INSTITUTION
MHRD Approved & NAAC Accredited
Karunya Nagar, Coimbatore - 641 114, Tamil Nadu, India.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SCHOOL OF ENGINEERING AND TECHNOLOGY

LABORATORY RECORD

2021-2022

Subject Code

20CS2050L

Subject Name

SOFTWARE ENGINEERING LAB

Register No.: URK20CS2001

This is the record of the work done by Mr./Ms. **RUBAN GINO SINGH.A** during the odd semester of the academic year 2021-2022 and submitted for the University Practical Examination held on **8/11/2021**

INDEX

Ex No: 1	**PROBLEM ANALYSIS AND PROJECT PLANNING**
Date: 12/07/2021	
Video Link:	https://drive.google.com/file/d/1DX0sxFwmEBmJvaBC-MfESlKX3_9xHAj_/view?usp=sharing

AIM:

The concept of this project is to provide a low-cost E-Learning Platform to access free online courses and hands-on practices for the students and working professionals. Also, provide online tutor support while studying the courses online.

PROBLEM STATEMENT:

The world is rapidly moving towards the technical side, but unfortunately, the pandemic situation made the students life virtual. On the other hand, working professionals lost their jobs due to violate market situations. In some cases, students couldn’t continue their studies and were forced to do odd jobs to support their families to meet basic needs. The education life of the affected students is in real danger.

One of the most vulnerable impacts that society faces in this condition is unpredictable. As per the data captured from The Hindu newspaper (Aug 2020), Over 24 million students left their studies from the schools and moved to a full-time job to rebuild their family situation. And this leads to being the worst fall in the literacy rate in India. And it would affect the other developing countries in the world.

Developing this E-Learning platform required a moderate level of the skilled web development team of 6 members to host publicly on the web. Because it requires both front-end and back-end development work to be done. And it will make a better look at a fully functional website, even though it is a dynamic website, which is updated daily by every teaching instructors.

BUSINESS CASE:

TITLE/ROLE:

E-Learning System to provide online teaching for the students free of cost.

THE PROJECT:

- E-Learning platform to provide free education to the students.
- Level of Implementation of this E-Learning platform has the three-layer architecture to facilitate sharing, learning modules, reusing them, and interoperability among different learning content efficiency.
- Due to covid crises in the world, a large group of students dropped our from the schools to get directly jumped into one-day jobs to support the financial situation of their families;
- If this condition goes uncontrollably, the literacy rate in India and other developing countries in the world.
- Now the only possible way to bring back students to their regular life.
- Developing an online platform will surely be a pathway to solve the problem.

HISTORY:

Covid-19 makes us feed dissatisfied with every work that we were performing in our day-to-day life. One of the most sensitive things was, covid-19 has made a tremendous impact on students' life by changing offline environment to online environment, Sudden crash in economic conditions made students lose interest in their studies and forced them to do one day jobs to year, if we want to bring back students to natural life usually takes at least 4 to 5 years to get all the things in normal condition These are some Bad history of the society, which is the key to activate this E-Learning system;

LIMITATION:

Building up this E-Learning platform faces several limitations:

1. Financially, it would require around 1,50,000 rupees to launch and receive users publicly on the web.
2. This e-learning platform required skilled, friendly teachers to post their lecture videos.
3. Some of the funds need to give online ads, television ads, social media ads, and other advertisements need to promote the platform throughout the world especially, in India.

Eventually, it would require qualified teachers to teach the students who are enrolled globally. That would be most challenging part of the whole project.

APPROACH:

- A flexible user experience is needed. If the user accesses the website, they will understand every single button, link and image to know about the video lecture on the website.
- Full-stack web developers need to work on the front-end and back-end parts of the website.
- Domain, hosting provider, and business emails are needed to build the user's trust and satisfaction.
- The website administration team needs to manage the client and admin panel of the website. Since it has a three-layer architecture, it won't be down anytime
- Promotions to get a large number of users in this e-learning platform.

BENEFITS:

Launching this E-Learning platform successfully on the web will give a lot of benefits to the learning students and the teaching instructors.

There are some ways where the students will benefit.

- The project has a three-layer architecture to facilitate sharing learning modules, reusing them, and interoperability among different learning content efficiently.
- 24/7 free coaching videos are available to focus the students constantly on their studies.
- A two-way chat application was available inside the website to clear their doubts through the instructor privately or the students.
- It would help to improve the Literacy rate in India in one year.

Some ways where the instructors will benefit:

- The instructors who joined this E-Learning system will get paid once they've posted the entire Modules in a particular course.
- They'll get paid for every better course material/content.

How does the learning platform benefit?

- This platform had a specific page where an extensive level of projects developed opt for sale.
- This learning platform will get a specific income from the project page where the other people and professionals purchasing software and materials from the page.

SECTION 1. Project Description:

This project is simply an E-Learning platform that will work as a dynamic website. This platform powered by cloud computing would allow independent Learning Management System (LMS) embedded in various E-Learning standards to share their learning objects modules and content. It will work with a three-layer architecture to facilitate sharing, learning modules, reusing them, and interoperability among different learning content efficiently. The middle layer of the infrastructure contains an indexing module and a metadata transformation module to encourage the exchange of metadata among acknowledged e-learning standards.

This website allows learners to use available leaning objects without requiring them to affiliated with any other Learning Management System (LMS).

SECTION 2. Problem Constraints:

- **What is the Purpose and Need for the work?**

The purpose and the need for the work is to provide education for free of cost and some courses at low cost.

- **What questions need to answer?**

If this project would launch by acquiring all the facilities, would it be possible to pay the instructors' salary constantly? How to encourage students to get into E-Learning if they are more interested in face-to-face learning?
How to make this new tool emerging and superior to the other similar e-learning platforms?

- **What are the key issues considered?**

Some significant issues like how the instructors will get paid constantly when this website will give courses free of cost? This case would be quite a complicated situation when going to work on this website.
Strong internet access? Is it possible to provide offline access with limited content?

- **What are the Goals and Objectives of the work?**

Overall goals and objectives were to provide free and low-cost education to the students who suffered from financial conditions due to covid-19.

- **Who is the audience?**

The audiences are the students and working professionals.

- **What types of useable information and tools are available and practical?**

The first usable information is study material which the students will get from the time they clicked the enrol option.

SECTION 3. Schedule, Resource, and Budget Constraints

- **What are the Existing Resources? Are they internal or external?**

Existing resources are internal resources like Domains and Hostings to develop and launch the website publicly on the web.

- **What is the Feasible Budget?**

At least 1,50,000 INR is wanted for the completion of work including, Domain, Hosting, and social media promotions.

- **What are the time constraints that may dictate the delivery of work items?**

The total completion with users would require a minimum of three to five months.

- **What is the availability and quality of existing data?**

Not even a single data is now available to start the project.

RESULT:

It would be complicated to provide education online for the students free of cost. Therefore, the E-learning project is in the initial conceptual phase. Also, I am working on making further possibilities to provide education to students free of payment.

Ex No: 2 Date: 19/07/2021	**SOFTWARE REQUIREMENT ANALYSIS**
Video Link:	https://drive.google.com/file/d/1jIrZNuIY0HR-rrPCYm0InIFj5WognhOw/view?usp=sharing

AIM:

To determine a project with a plan requires an essential thing to be done. That is the Software Requirement Analysis, and which will be used to analyse the software needs and the requirement to be done. It will simply examine the functional and non-functional requirements of the E-Learning Website.

DESCRIPTION:

Software requirements analysis:

Requirements Analysis is perhaps the most difficult, most error-prone and most communication intensive software development. It and be successful only through an effective customer-developer partnership. It is needed to know what the users really need. There are several requirements elicitation methods. Few of them are listed below –

1. Interviews
2. Brainstorming Sessions
3. Facilitated Application Specification Technique (FAST)
4. Quality Function Deployment (QFD)
5. Use Case Approach

The success of an elicitation technique used depends on the maturity of the analyst, developers, users, and the customer involved.

FUNCTIONAL REQUIREMENTS:

In software engineering, a functional requirement defines a system or its component. It describes the functions a software must perform. A function is nothing but inputs, its behaviour, and outputs. It can be a calculation, data manipulation, business process, user interaction, or may other specific functionality which defines what function a system is likely to perform.

Functional software requirements help you to capture the intended behavior of system. This behavior may be expressed as functions, services or tasks or which system is required to perform.

- Services the system should provide.
- What the system should do or not in reaction to particular situations.
- Example: "If a patient is known to be allergic to a particular medication, then prescription of that medication shall result in a warning message being issued to the prescriber"

Some of the more typical functional requirements include:

- Business Rules
- Transaction corrections, adjustments and calcellations
- Administrative functions
- Authentication
- Authorization levels
- Audit Tracking
- External Interfaces
- Certification Requirements
- Reporting Requirements
- Historical Data
- Legal or Regulatory Requirements.

NON-FUNCITONAL REQUIREMENTS:

A non-functional requirement defines the quality attribute of a software system. They represent a set of standards used to judge the specific operation of a system. Example, how fast does the website load?

A non-functional requirement is essential to ensure the usability and effectiveness of the entire software system. Failing to meet non-functional requirements can result in systems that fail to satisfy user need.

- Constraints on the services or functions offered by the system.
- Example: “The system shall be available to all clinics during normal working hours (Mon-Fri, 0830-1730). Downtime during normal working hours shall not exceed 5 seconds in any one day”

Some typical non-functional requirements are:

- Performance – for example Response Time, Throughput, Utilization, Static Volumetric
- Scalability
- Capacity
- Availability
- Reliability
- Recoverability
- Maintainability
- Serviceability
- Security
- Regulatory
- Manageability
- Environmental
- Data Integrity
- Usability
- Interoperability

SYSTEM REQUIREMENTS:

As mentioned in the problem statement. This E-Learning platform is a website and, it will run as a domain-based website to provide 24/7 course materials to the students as well as the working professionals to view the course materials to the user must need network stability to watch and download the course material also, this website has a two-way chat application which will connect the instructors and the students without any third party applications like What’s app, Telegram, Signal or Microsoft teams; Since the E-Learning platform provides some premium courses at low cost where the students and working professionals can make UPI payments to guy the course content and the other projects to get benefited;

SRS:

1.0 Overview:

The overview of this project is just an E-Learning website to provide the high-class education to the students at a low cost. This SRS will clearly explain the following questions.

How will instructors get benefited using this website?
How will the students get benefited?
What are the specifications to view/download the course materials?
How will this website help the students, working professionals and instructors?

2.0 GENERAL DESCRIPTION:

2.1 Product Perspective

This system allows stakeholders to invest in the student's project to get the expected outcome.

This system will display a wide variety of courses to the students and other working professionals.

This system will help the students to get constant improvement in the studies in any pandemic situation.

The system will provide information about the courses, real-time projects, sharable materials, and best instructors, depending upon their experience.

2.2 Product Functions:

The system provides the following functions:

- Life-time courses at one-time cost.
- Downloadable and sharable resources where the students can easily able to download, share the materials to the instructors by clearing doubts.
- 24/7 help and support.

- Secured authentication.
- Two-way chat application.
- Spambot to preventing phishing activities.
- Remainder mails to remind the enrolled students and working professionals to attend the quizzes and the assignments at an accurate time.
- After finishing a course, each student will get a credit and a badge as an appreciation where the badge is useful for further jobs and higher studies.

2.3 User Characteristics:

The users of this system are:

- **Level of User's computer knowledge:**
  - Since the users are students so, the students need a piece of basic computer knowledge to begin a course and, the student must know about how to create a Gmail account and how to enrol in the course.
- **Level of User's Business Knowledge:**
  - The admin panel users must know how to update the website. They need a piece of web development knowledge to work with this E-Learning dynamic website.
- **Frequency of Use:**
  - Since it was an E-Learning platform and the enrolment rate of the students is the frequency and, it will expect high use of user frequency.

2.4 General Constraints:

This E-Learning system will support the students to learn the course constantly without any interruptions.

This system will not allow the students to drop from the course that enrolled by them.

2.5 Assumption and Dependencies:

This system relies on the user’s internet bandwidth, where the user needs a stable bandwidth to view or download the course materials.

The system must have a satisfying interface to provide the best user experience and, it also needs to support some offline materials.

SPECIFIC REQUIREMENTS

3.1 Functional Requirements

3.1.1 Unit Registration

Every dynamic website needs a login page to check the user’s interactions with that particular website. This website has unit registrations which include the student’s help in selecting, adding, dropping, and changing a unit.

- **SRS-002(3.1.1.2):**

The system shall allow the user to delete a unit if the user has chosen to drop that unit.

- **SRS-003(3.1.1.3):**

The system shall check if a unit has constantly been watching the daily courses/materials.

- **SRS-004(3.1.1.4):**

The system shall allow the users to download the course materials where they can reuse them later.

- **SRS-005(3.1.1.5):**

The System can automatically unenroll the inactive students from the course.

- **SRS-006(3.1.1.6):**

The system shall allow the users to delete or sync the chat application from the concerned chat to the third-party apps.

- **SRS-007(3.1.1.7):**

This system will allow the users to add the courses to the cart while the concerned materials/project while programs are on sale.

3.1.2 Retrieving and Displaying Unit Information

SRS-014(3.1.2.1):

3.2 Design Constraints

- **SRS-031(3.2.1):**

The system shall store and retrieve persistent data.

- **SRS-032(3.2.2):**

Since the E-Learning system is a website, It doesn't need any specific requirements in the system.

- **SRS-033(3.2.3):**

The system shall developed using HTML, CSS, JS for the front-end and Node JS with Express js for the back-end.

3.3 Non-Functional Requirements

- **SRS-034(3.3.1):**

This system shall respond to any retrieval in less than 5 seconds

- **SRS-035(3.3.2):**

This system shall generate an analysis report within 1 minute

- **SRS-036(3.3.3):**

This system shall allow the user to stay remotely connected.

- **SRS-041(3.3.8):**

This system will accompanied by a comprehensive help section, where the users can search and know about this website.

3.5.3 Security

The security requirements are concerned with security and privacy issues.

SRS-029:

This system has a multi-functionality database for the instructors and the students with different servers to protect the login details from unauthorized use of the system.

SRS-030:

This system shall allow the password manager functions to the users to add, remove, and modify the user id and passwords. The login credentials have an implementation of two-factor authentication, which provides two-layer security to the users.

RESULT:

The revised Software Requirements Specification is made for the E-Learning Platform to get the stakeholders' approval.

Ex No: 3 Date: 26/07/2001	**USE CASE DIAGRAM**
Video Link:	https://drive.google.com/file/d/1OUtXoaES5ugBDqq5kQ3c5RZt9Y0Fglcn/view?usp=sharing

AIM:

Use Case Diagram for the project: E-Learning System.

Use case diagram is to identify the functionality provided by the system (use cases), The user who interact with the system (actors), and the association between the users and the functionality. Use cases are used in the Analysis phase of the software development to articulate the high-level requirements of the system. The primary goals of Use case diagrams include:

- Providing a high-level view of what the system does.
- Identify the users (actors) of the system
- Determining areas needing human-computer interface.

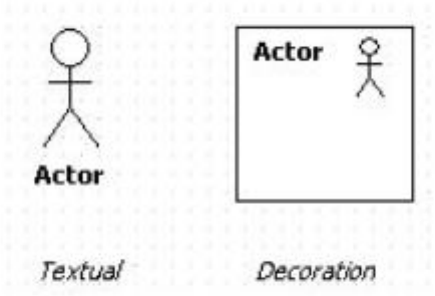
Use case extend beyond pictorial diagrams. In fact, text-based use case descriptions are often used to supplement diagrams and explore use case functionality in detail.

DESCRIPTION:

The basic components of Use case diagrams are the Actor, the Use case, and the Association.

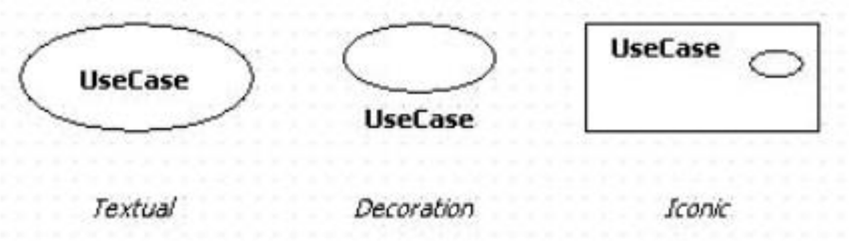
Actor:

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



Use Case:

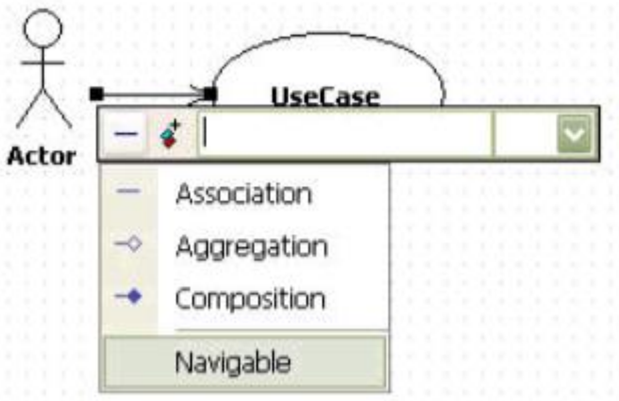
A Use case is a functionality provided by the system, typically described as verb + object (e.g.: Register Car, Delete User). Use cases are depicted with an ellipse. The name of the use case is written within the ellipse.



Association:

Associations are used to link Actors with Use Cases and indicate that an Actor participates in the Use case in some form. A line connecting the Actor and Use case depicts associations.

The following image shows how these three basic elements work together to form a use case diagram.



ALGORITHM:

Step 1: Define the actors like students, Instructor and Admin panel organizer for the E-Learning System.

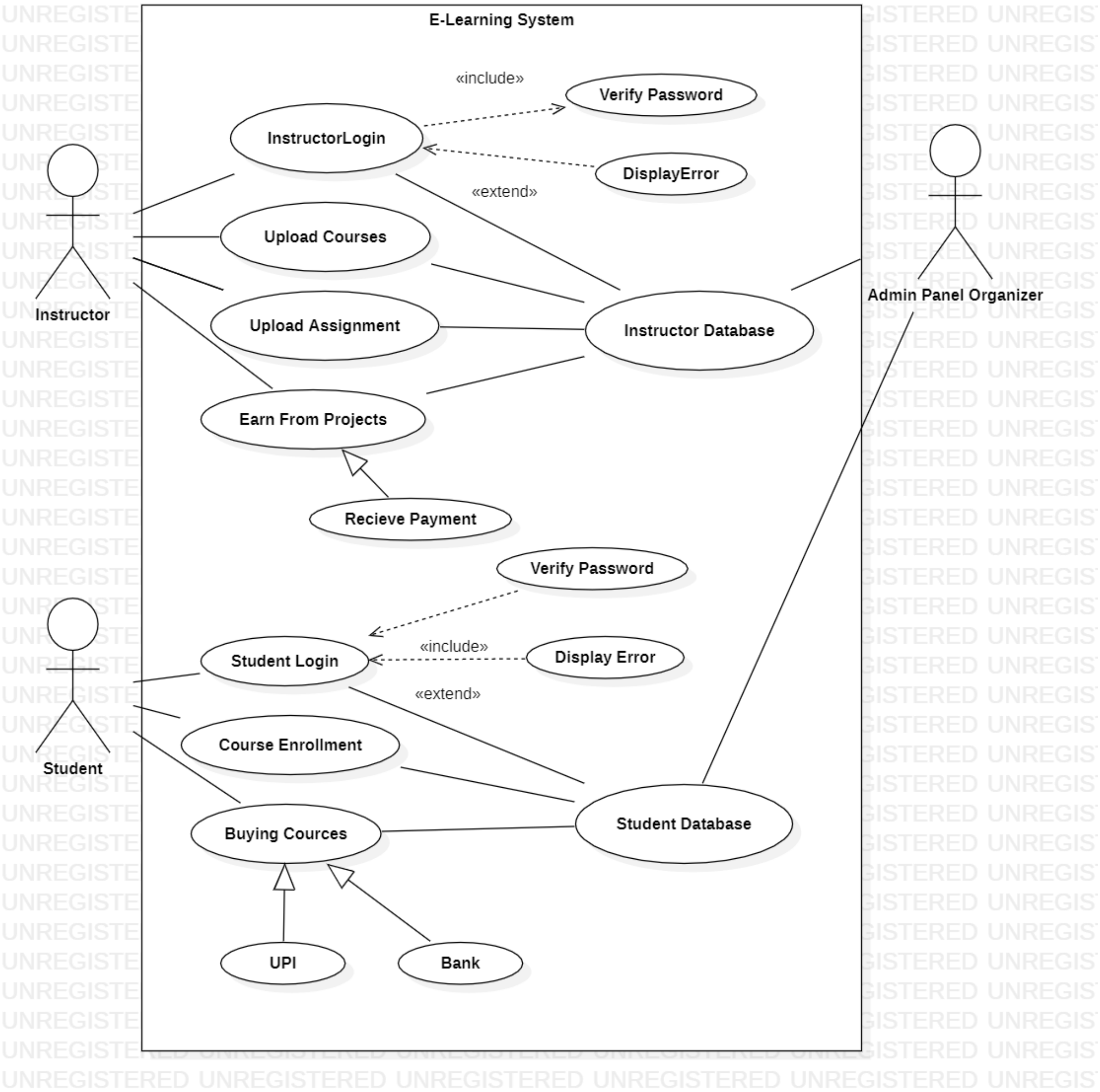
Step 2: Determine the events that are necessary for developing a system.

Step 3: Represent the communication relationship of an actor in the system by connecting the actor symbol to the Use-Case symbol with a solid path.

Step 4: Represent the ‘Uses’ and ‘Extends’ relationship if required.

Step 5: Use the necessary tools for developing the E-Learning System .

OUTPUT:



SPECIFICATION:

The above use case diagram is for the project E-Learning System. This E-Learning system contains 3 actors of Instructors, Students and Admin panel organizer. Instructor associated with Login page, course uploading page and payment receiving page. Students associated with Login page, Course Enrolment and buying courses. And the third actor is the admin panel organizer to organize all the backend stuffs.

RESULT:

The Use cases are used in the Analysis phase of software development to articulate the high-level requirements of the system are drawn successfully.

Ex No: 4	**CLASS DIAGRAM**
Date: 09/08/2021	
Video Link:	https://drive.google.com/file/d/1RvkDyjw0sDwSBSa4b1kR3ah9VdS_g8i_/view?usp=sharing

AIM:

Class diagrams identify the class structure of a system, including the properties and methods of each class. It specifies the various relationships that can exists between classes, such as an inheritance. The class diagram is one of the most widely used diagrams from the UML specification. Part of the popularity of the Class diagrams is the fact that many CASE tools, such as Rationale Rose, will provide clarity of the designing process.

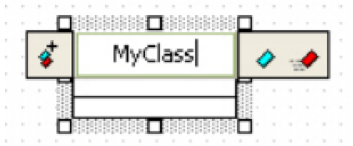
This document will explain the class diagram for the E-Learning Platform.

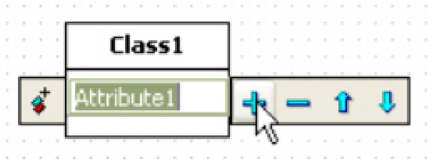
DESCRIPTION:

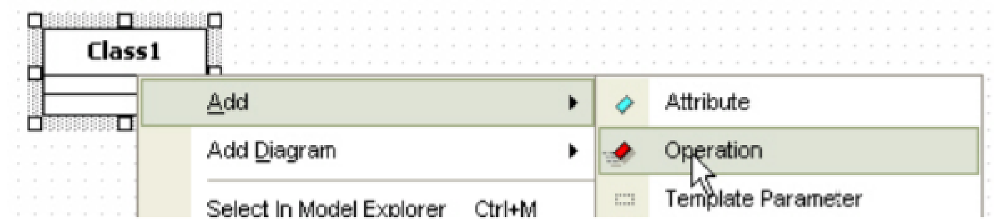
The basic components on a Class diagram are classes and the relationships between them.

Class

A class is depicted using a rectangle divided into 3 sections. The top section is the name of the class. The middle section defines the properties of the class. The bottom section lists the methods of the class.

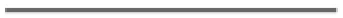






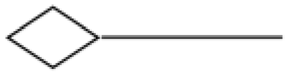
Association

An Association is a generic relationship between two classes and is modelled by a line connecting the two classes. This line cam be qualified with the type of relationship and can also feature multiplicity rules (for e.g. One-to-one, One-to-many, many-to-many) for the relationship.



Composition

If a class cannot exist by itself, and instead must be a member of another class, then that class has a Composition relationship with the containing class. A composition relationship is indicated by a line with a filled diamond.

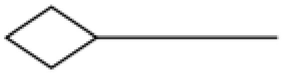


Dependency

A Dependency relationship is indicated by a dotted arrow.

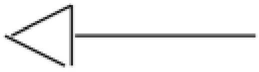
Aggregation

Aggregations indicate a whole-part relationship and are known as "has-a" relationships. An Aggregation relationship is indicated by a line with a hollow diamond.

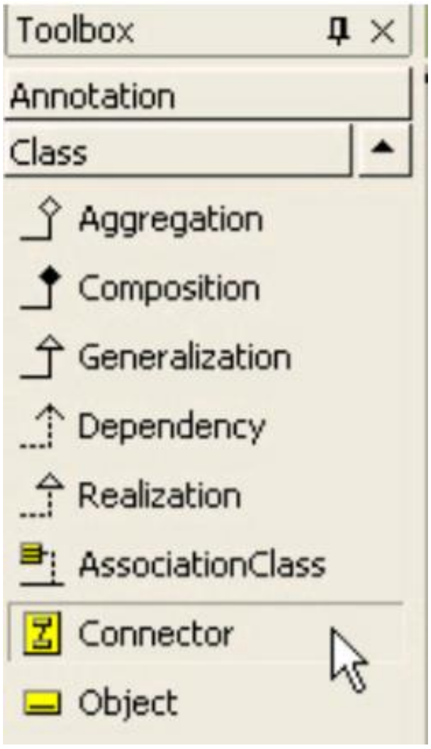


Generalization

A Generalization relationship is the equivalent of an inheritance relationship in object-oriented terms. ("is-a" relationship). A Generalization relationship is indicated by an arrow with a hollow arrow head pointing to the base of "parent" class.



Toolbox for class diagram



ALGORITHMS

Step 1: Identify the various classes that behave as main to perform various functions in the system.

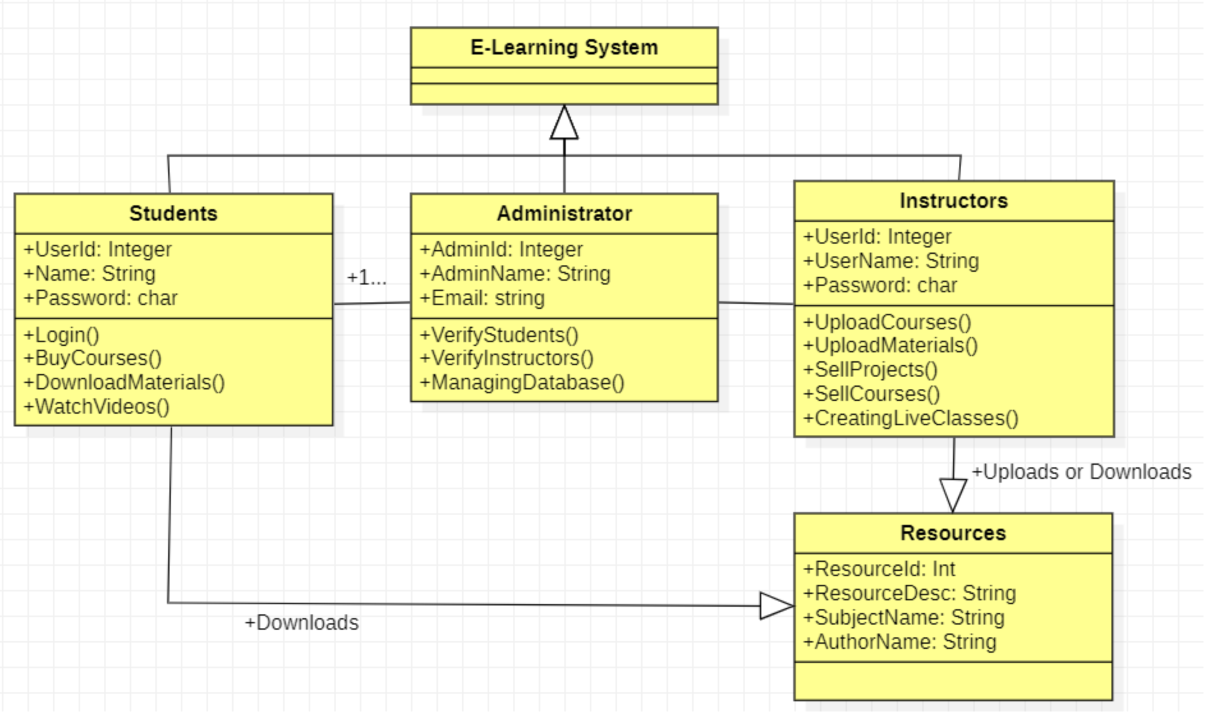
Step 2: Define all its attributes and what operations they perform.

Step 3: Identify the various relationships among the classes.

Step 4: The relationship between a class and its subclass is represented by generalization.

Step 5: Design the class diagram for the E-Learning System using the tools provided.

OUTPUT:



The Class diagram for the E-Learning system explains the website has an Administrator who will verify the accounts of the students and the instructors who are the users of this e-learning system. And the functionalities of the students are they can buy courses, download materials, and watch the videos. The functionalities of the instructors are uploading the courses, uploading materials for students, selling their own projects, selling their courses and creating live classes.

RESULT:

Class diagrams identify the class structure of a system, including the properties and methods of each class and the relationships between them are designed successfully.

Ex No: 5	ACTIVITY DIAGRAM
Date: 23/08/2021	
Video Link:	https://drive.google.com/file/d/1GOMGNUAbaAOtARRbkSICCxS-TaGnCEgp/view?usp=sharing

OBJECTIVE:

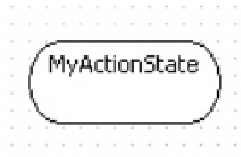
Activity diagrams are used to document workflows in a system, from the business level down to the operational level. When looking at an activity diagram, you'll notice elements from state diagrams. In fact, the Activity diagram is a variation of the state diagram where the "states" represent operations,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

This document will clearly explains the Activity diagram for the project E-Learning platform.

DESCRIPTION:

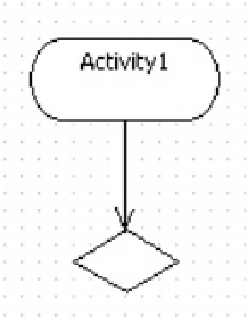
Activity states:

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



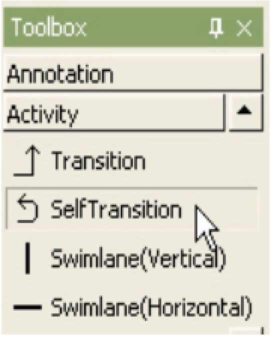
Transition:

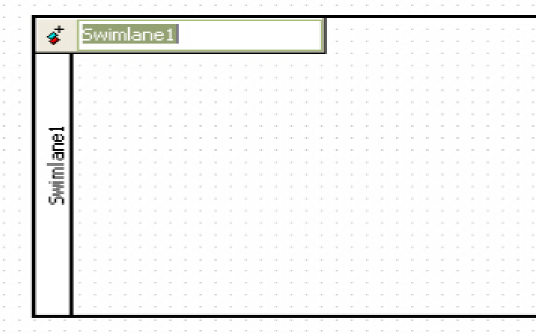
When an Activity state is completed, processing moves to another Activity state. Transitions are used to mark this movement. Transitions are modelled using arrows.



Swimlane:

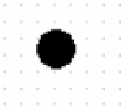
Swim lanes divide activities according to objects by objects in column format and placing activities by the object within that column. Objects are listed at the top of the column, and vertical bars separate the columns to form the swim lanes.





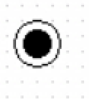
Initial State:

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



Final State:

Final States mark the end of the modelled workflow. There can multiple Final States on a diagram, and these states are modelled using a solid circle surrounded by another circle.



Synchronization Bar:

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

ALGORITHM:

Step 1: Identify the operations and transformations that are triggered by the completion of an operation, which is referred as activities.

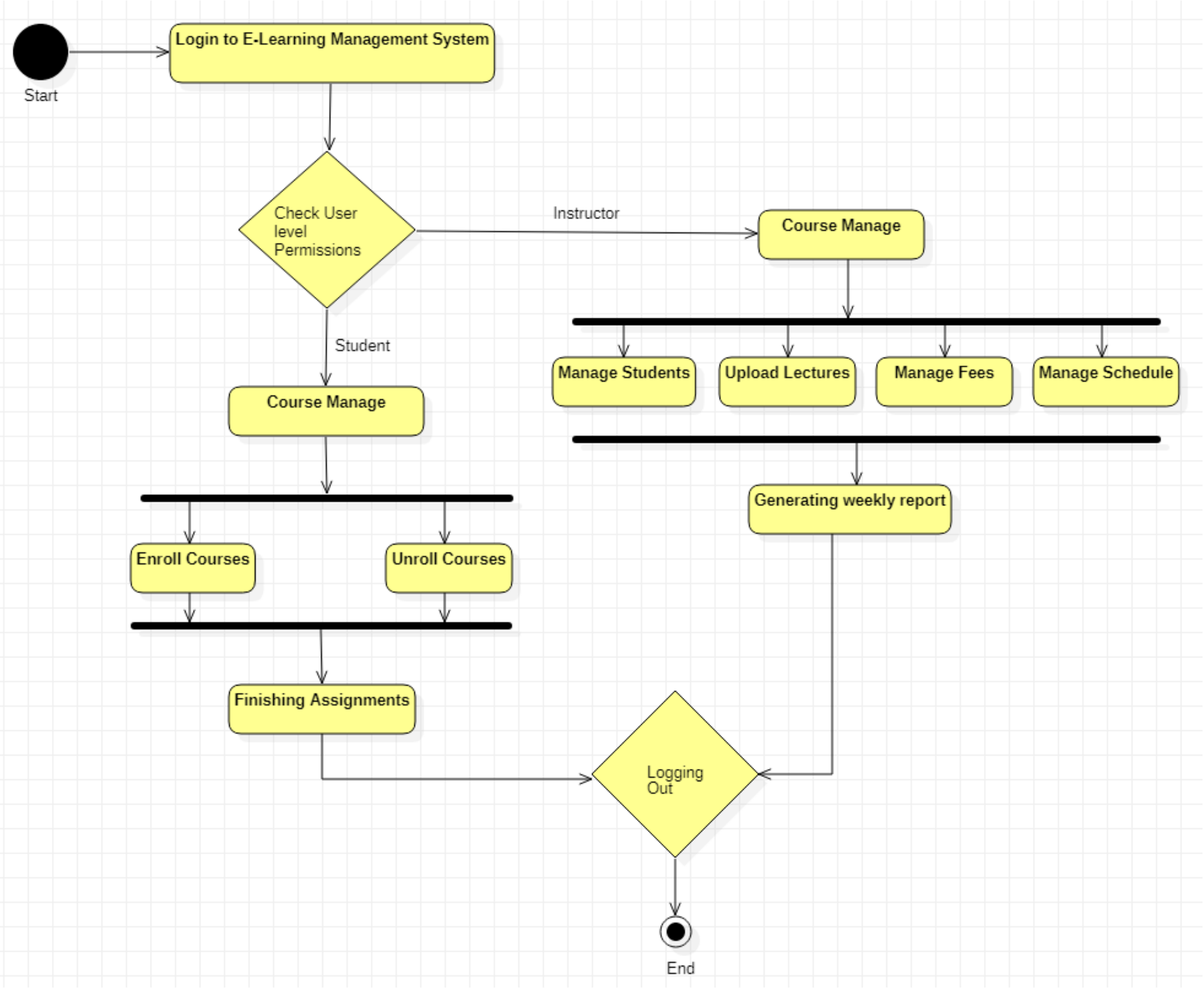
Step 2: Represent those activities by a rounded rectangle.

Step 3: Link the activities by automatic transactions, represented by arrows.

Step 4: If needed a decision which is represented by a diamond with many transitions coming out of it.

Step 5: Design the Activity diagram using the information obtained from the above steps using the tools provided.

OUTPUT:



EXPLAINATION:

The E-Learning System contains two activities for the two different user of this website. The activities are, If the student is logged into the website then he/she can have the permission/access for the enrolment/unenroll from the courses that posted by the Instructor.

And the other user Instructor Can have the activities of Manage the Courses, manage the students, upload/download the lectures, manage the fees, manage the Schedules, and generating the weekly report for his/her analytics.

RESULT:

The activity diagrams are used in the Analysis phase of software development to articulate the high-level requirements of the system are drawn successfully.

Ex No: 6	**SEQUENCE DIAGRAM**
Date: 06/09/2021	
Video Link:	https://drive.google.com/file/d/1eNZh-0NhvqesbaoG6SoK4B1Xn2XWw5Lt/view?usp=sharing

OBJECTIVE

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

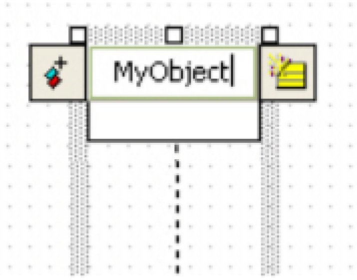
Here, this document will clearly explain the Sequence Diagram for the Project E-Learning System.

DESCRIPTION

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

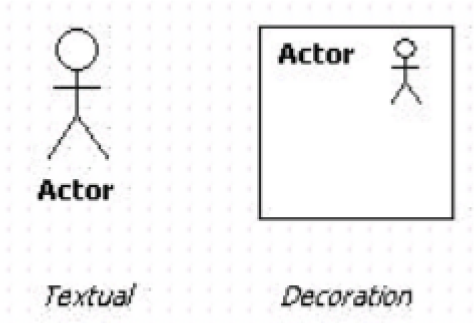
Object

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



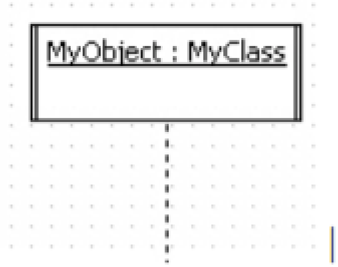
Actor

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



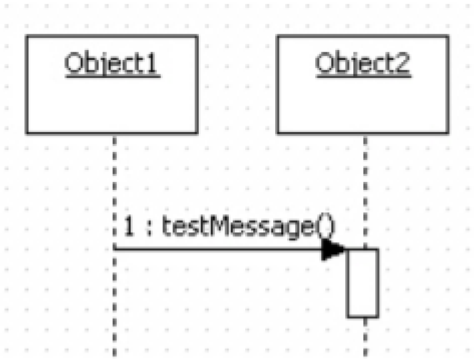
Lifeline

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



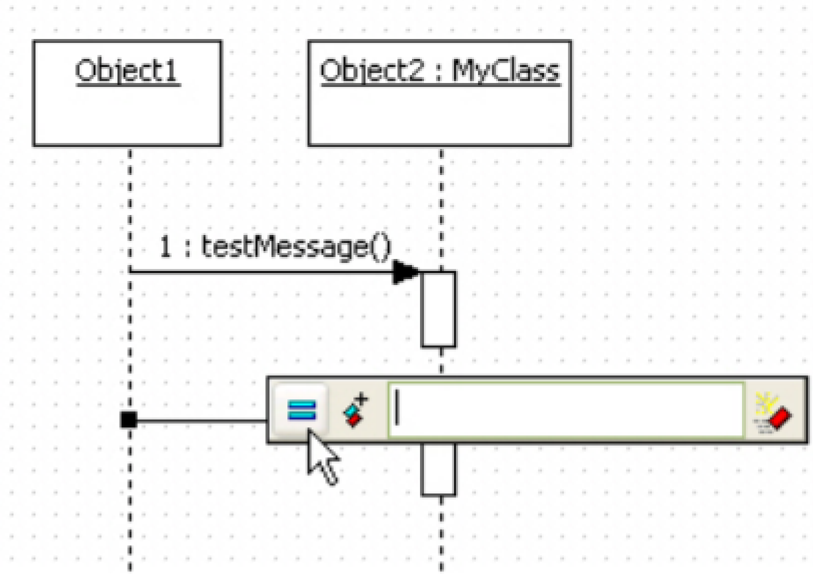
Activation

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



Message

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



ALGORITHM

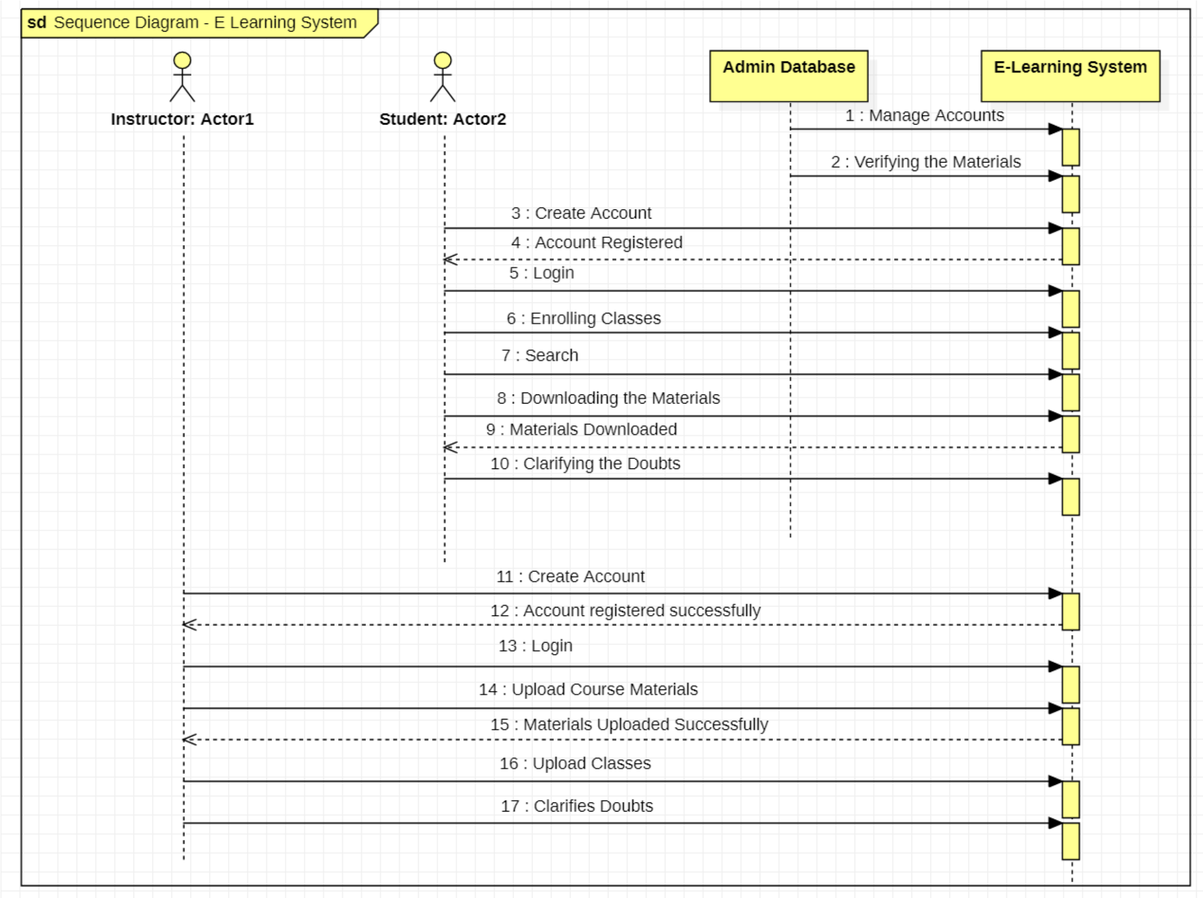
Step 1: Identify the objects in the diagram according to the system to be drawn.

Step 2: Identify the sequence of transfer of messages between the objects.

Step 3: Determine how the messages are passed between two objects and how other objects respond.

Step 4: Create the sequence diagram based on the information from step 1 and step 2 with the tools provided.

OUTPUT:



This Sequence diagram for the project e-learning system will clearly explained that this project has two actors of Student and an Instructor. Here the student can able to create account, login account and enrolling into the account and the Instructor can able to do the same login to account, upload the course materials, upload the classes and clarifies the doubts for the students. These are all the sequence of this E-Learning System.

Result:

The Sequence diagrams are used in the analysis phase of the software development to articulate the high-level requirements of the system are drawn successfully.

Ex. No: 7a	**COMPONENT DIAGRAM**
Date: 13/09/2021	
Video Link:	https://drive.google.com/file/d/1cw0HSe5donE6tKaa8UpZXOSPjLcXjKh-/view?usp=sharing

OBJECTIVE

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

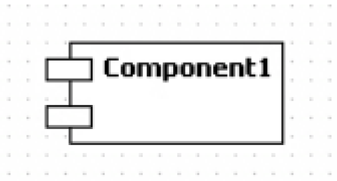
This document will explain the Component diagram for the website E-Learning System.

DESCRIPTION

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

Component

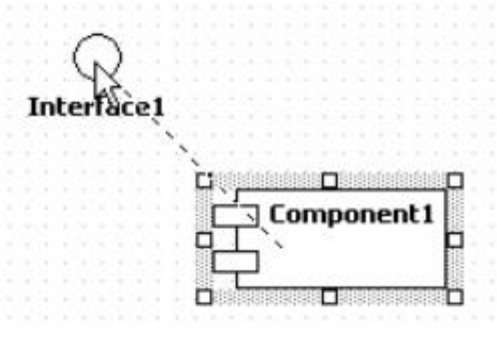
A component represents a software entity in a system. Examples include source code files, programs, documents and resource files. A component is represented using a rectangular box, with two rectangles protruding from the left side



Dependency

A dependency is used to model the relationship between two components. The notation

for a dependency relationship is a dotted arrow, pointing from a component to the component it depends on.



ALGORITHM

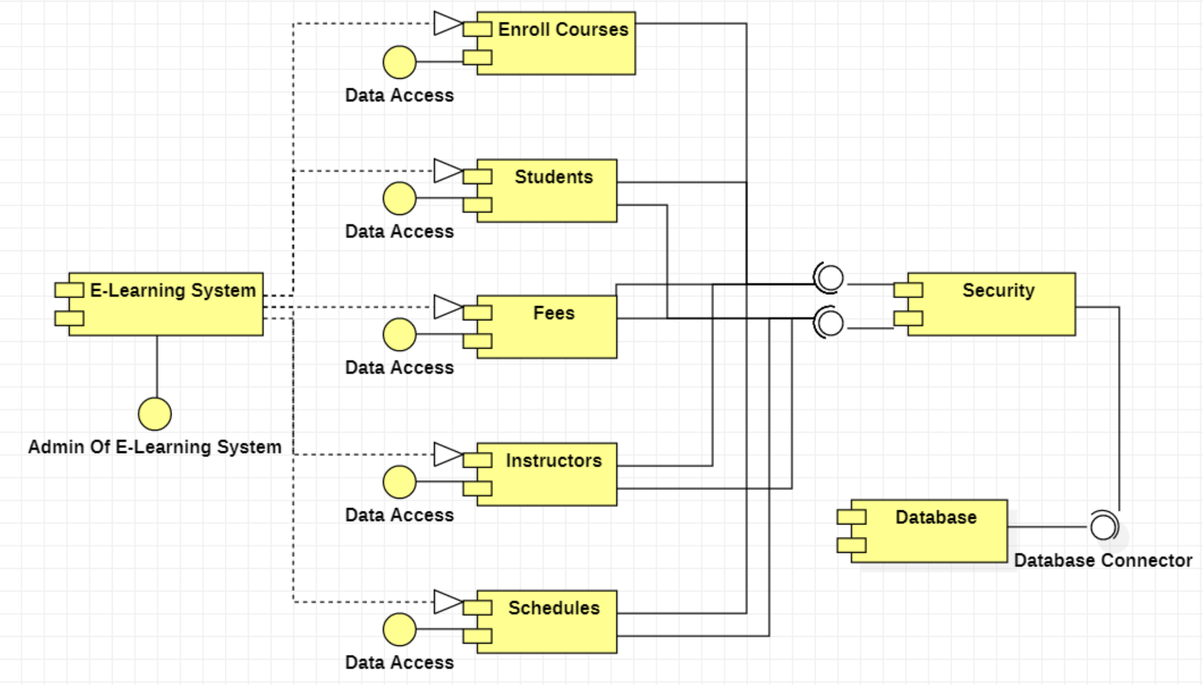
Step 1: Identify the software entities used in the system.

Step 2: The dependency relationship is drawn according to the system set up.

Step 3: The link is drawn between the components.

Step 4: Terminate the process after completion.

OUTPUT:



The components of this E-Learning System has a Enroll courses, students, fees, instructors, security, and Database control.

RESULT:

The component diagrams are used in the implementation phase of software development to articulate the high-level requirements of the system are drawn successfully

Ex. No: 7b	DEPLOYMENT DIAGRAM
Date: 13/09/2021	
Video Link:	https://drive.google.com/file/d/1WStiqqQpf7yDi-bIMKtQ8PhneyaFYIIm/view?usp=sharing

OBJECTIVE

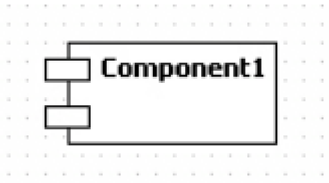
Deployment diagrams are another model in the implementation diagram category. The Deployment diagram models the hardware used in implementing a system and the association between those hardware components. Components can also be shown on a Deployment diagram to show the location of their deployment. Deployment diagrams can also be used early in the design phase to document the physical architecture of a system.

This Document is explains the Deployment diagram for the project E-Learning System.

DESCRIPTION

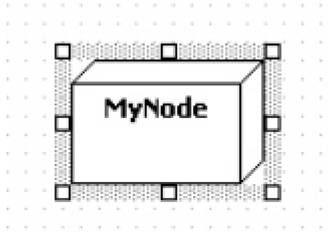
Component

A component represents a software entity in a system. Examples include source code files, programs, documents, and resource files. On a deployment diagram, components are placed within nodes to identify their deployed location. A component is represented using a rectangular box, with two rectangles protruding from the left



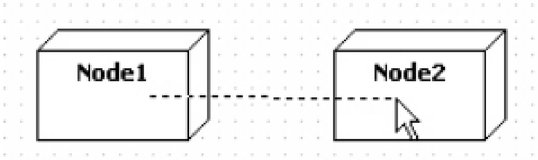
Node

A node represents a piece of hardware in the system. This entity is represented by a three-dimension



Association

An association, drawn as a solid line between two nodes, indicates a line of communication between the hardware elements



ALGORITHM

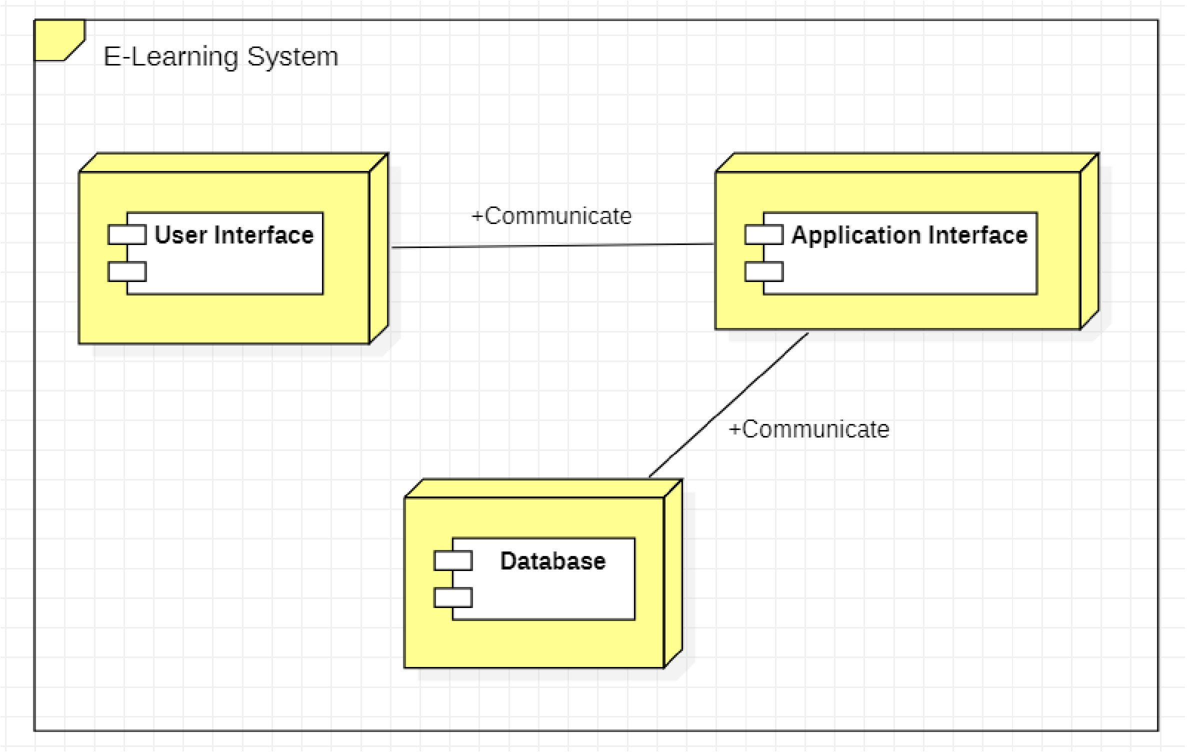
Step 1: Identify the processors and devices used in the Deployment Diagram.

Step 2: The processors are connected according to the office set up.

Step 3: The association link is drawn with the messages.

Step 4: Terminate the process after completion.

OUTPUT



The Deployment diagram for the E-Learning system has a 3 node called user Interface, Application Interface and Database. Where the 3 nodes communicate to perform the operations that are in this E-Learning System.

RESULT:

The deployment diagrams are used in the implementation phase of software development to articulate the high-level requirements of the system are drawn successfully

Ex. No: 8	**ESTIMATION OF PROJECT SIZE USING FUNCTION POINT ANALYSIS & COCOMO MODEL**
Date: 27/09/2021	
Video Link:	https://drive.google.com/file/d/1TmRDZgnvQTdGU1oxYWlq5R8TtY2eXoVz/view?usp=sharing

OBJECTIVE

The objective is to calculate the estimation of project size using function analysis & COCOMO mode.

This document will clearly explain the Function point analysis and COCOMO model for the project E-Learning platform.

METHODOLOGY

A project would be fall in one of the scenarios.

1. Much relevant project data is available for the current project but not much information about previous projects.
2. Previous project data ae available for the project but not much information about the current project.
3. Project data are available for the current project as well as that of the previous projects.
4. Some project data are available for the current projects.
5. No project data are available for both current as well as previous projects.

Estimation Technique Selection Based on Project Information Availability:

	Project Details	**Estimation Technique**
01	Historical project data & current project data	Function Point Analysis
02	Current project data	COCOMO, Wide Band, Delphi
03	No data	No Technique

DESCRIPTION

(i)Function point analysis

The function point is a “unit of measurement” to express the amount of business functionality an information system (as a product) provides to a user. Function points are used to compute a functional size measurement (FSM) of a software. The cost (in dollars or hours) of a single unit is calculated from past projects.

Step-1: Calculate F where

Scale varies from 0 to 5 according to character of Complexity Adjustment Factor (CAF). Below is the scale:

0 – No Influence
1 – Incidental
2 – Moderate
3 – Average
4 – Significant
5 – Essential

Step – 2: Calculate Complexity Adjustment Factor (CAF):

Step – 3: Calculate Unadjusted Function Point (UFP) by multiplying each individual function point to corresponding values in the table.

Measurement Parameter	**Weighting factor**		
	Simple	**Average**	**Complex**
No. of user inputs	3	4	6
No. of user outputs	4	5	7
No. of user inquiries	3	4	6
No. of files	7	10	15
No. of external interfaces	5	7	10

Step – 4: Calculate Function Point (FP)

FP = UFP * CAF

Upon Calculating FP, it is used to calculate productivity and cost.

(ii) Basic COCOMO model

The COnstructive COst Estimation MOdel (COCOMO) model gets the number of estimated lines of code for the project and calculates the overall time and people required for project. The

output differs based on the different project type and the different types of projects are as follows.

(a) Organic: Relatively small, simple software projects in which small teams with good application experience work to a set of less than rigid requirements.

(b) Semi-detached: An intermediate, (in size and complexity), software project in which teams with mixed experience levels must meet a mix of rigid and less than rigid requirements.

(c) Embedded: A software project that must be developed within a set of tight hardware, software and operation constraints

The COCOMO models calculates the no of person (effort) and duration using the equations 6 &

7.

where

• KLOC is the estimated size of the software product expressed in Kilo Lines of Code

• a, b, c, d are constants for each category of software products and their values are as follows.

Project type	a	b	c	d

Organic	2.4	1.05	2.5	0.38
Semi-detached	3	1.12	2.5	0.35
Embedded	3.6	1.2	2.5	0.32

- 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 person months (PMs)

OUTPUT (Manual Calculation):

Step 1 & 2: CAF Calculation

		Weightages
1.	Does the system require reliable backup and recovery?	**2**
2.	Are data communications required?	**3**
3.	Are there distributed processing functions?	**0**
4.	Is performance critical?	**0**
5.	Will the system run in an existing, heavily utilized operational environment?	**0**
6.	Does the system require on-line data entry?	**5**
7.	Does the on-line data entry require the input transaction to be built over multiple screens or operations?	**5**
8.	Are the master files updated on-line?	**5**
9.	Are the inputs, outputs, files, or inquiries complex?	**3**
10.	Is the internal processing complex?	**3**
11.	Is the code to be designed reusable?	**5**
12.	Are the conversion and installation included in the design?	**4**
13.	Is the system designed for multiple installations as different organizations?	**0**
14.	Is the application designed to facilitate change and ease of use by the user?	**0**
	Total(F)	**35**

CAF = 0.65 + (0.01 * F)

CAF = 0.65 + (0.01 * 35)

CAF = 1

Step 3: UFP Calculation

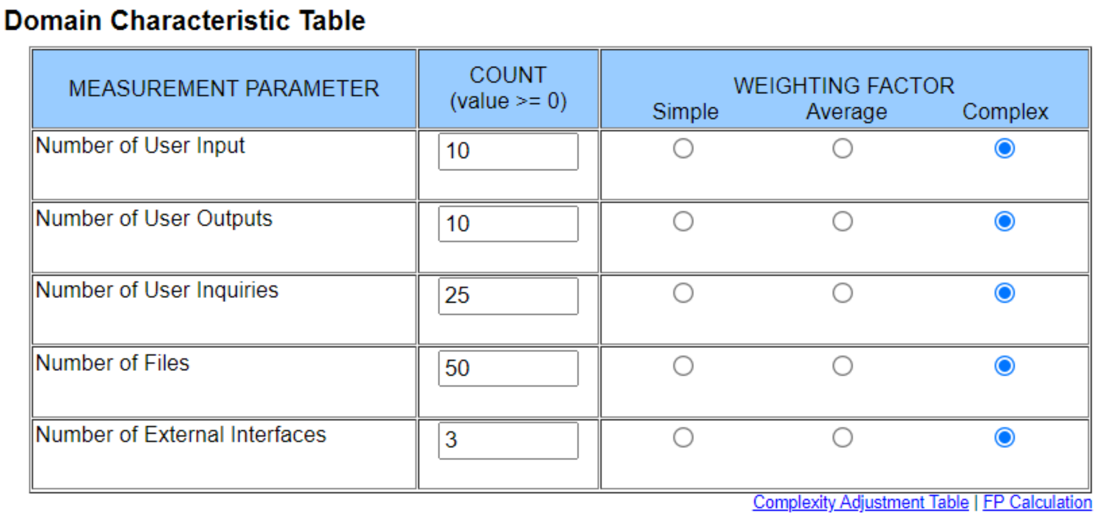
Measurement Parameter	Count	Weighing Factor (WF)	Count x WF
No. of User Input	10	6	60
No. of user output	10	7	70
No. of user inquires	25	6	150
No. of files	50	15	750
No. of external interfaces	3	10	30
		UFP	1060

Step 4: FP, Productivity & Cost calculation.

FP = UFP * CAF = 1060 * 1

FP = 1060

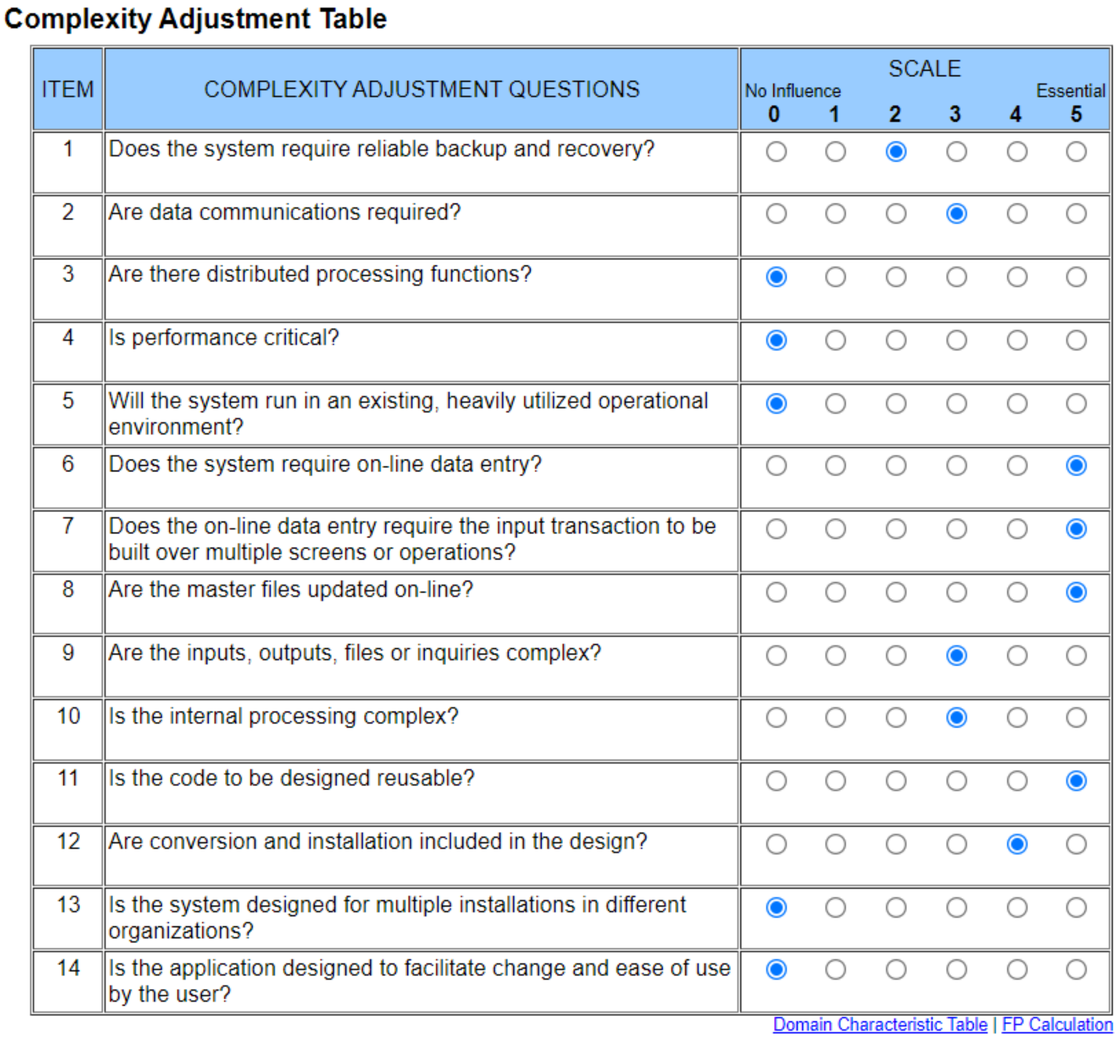
OUTPUT (Verification):



Domain Characteristic Table

MEASUREMENT PARAMETER	COUNT (value >= 0)	WEIGHTING FACTOR		
		Simple	Average	Complex
Number of User Input	10	○	○	●
Number of User Outputs	10	○	○	●
Number of User Inquiries	25	○	○	●
Number of Files	50	○	○	●
Number of External Interfaces	3	○	○	●

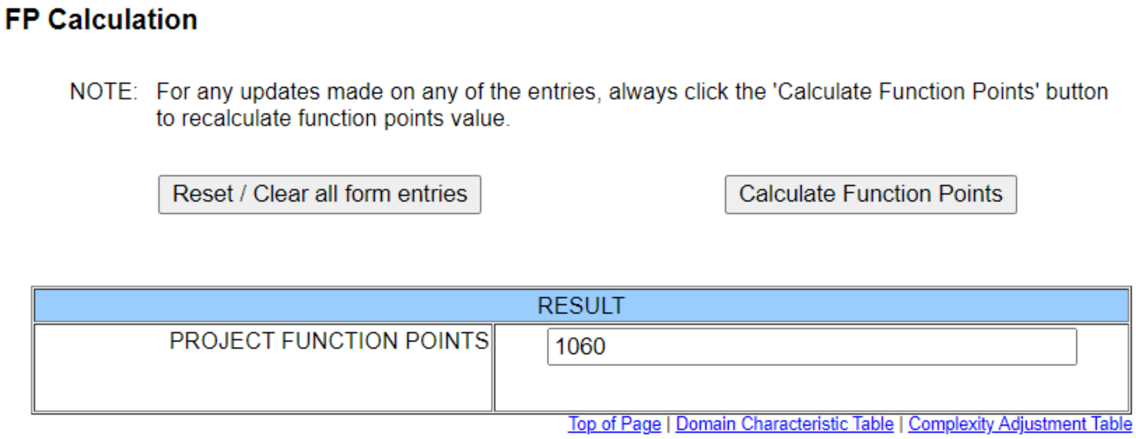
Complexity Adjustment Table | FP Calculation



Complexity Adjustment Table

ITEM	COMPLEXITY ADJUSTMENT QUESTIONS	SCALE No Influence 0	1	2	3	4	Essential 5
1	Does the system require reliable backup and recovery?	○	○	●	○	○	○
2	Are data communications required?	○	○	○	●	○	○
3	Are there distributed processing functions?	●	○	○	○	○	○
4	Is performance critical?	●	○	○	○	○	○
5	Will the system run in an existing, heavily utilized operational environment?	●	○	○	○	○	○
6	Does the system require on-line data entry?	○	○	○	○	○	●
7	Does the on-line data entry require the input transaction to be built over multiple screens or operations?	○	○	○	○	○	●
8	Are the master files updated on-line?	○	○	○	○	○	●
9	Are the inputs, outputs, files or inquiries complex?	○	○	○	●	○	○
10	Is the internal processing complex?	○	○	○	●	○	○
11	Is the code to be designed reusable?	○	○	○	○	○	●
12	Are conversion and installation included in the design?	○	○	○	○	●	○
13	Is the system designed for multiple installations in different organizations?	●	○	○	○	○	○
14	Is the application designed to facilitate change and ease of use by the user?	●	○	○	○	○	○

Domain Characteristic Table | FP Calculation



FP Calculation

NOTE: For any updates made on any of the entries, always click the 'Calculate Function Points' button to recalculate function points value.

Reset / Clear all form entries
Calculate Function Points

RESULT	
PROJECT FUNCTION POINTS	1060

Top of Page | Domain Characteristic Table | Complexity Adjustment Table

Output (Calculation-Organic mode):

KLOC = 5

Effort = a*KLOC^{b} = 2.4 * 5$^{1.05}$ = 2.4 * 5.42

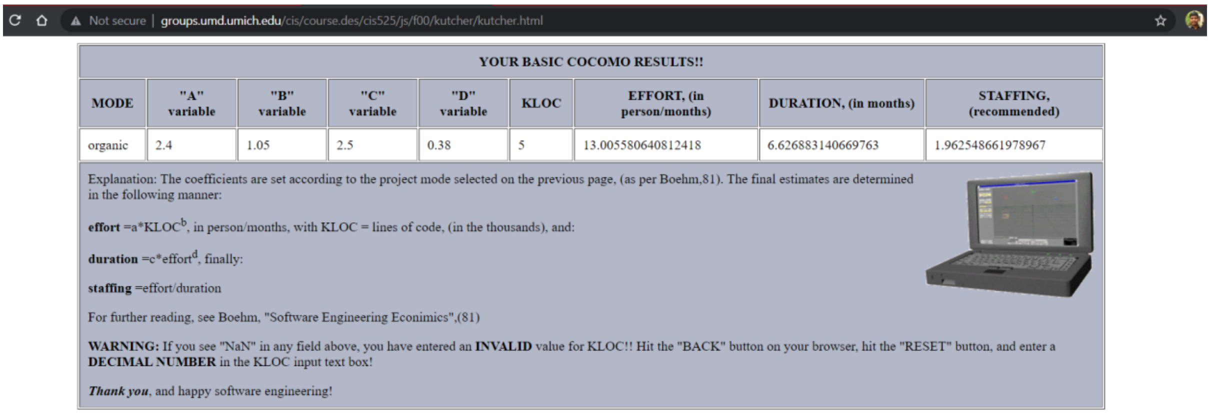
Effort = 13.01

Duration = c * effort^{d} = 2.5 * 13.01$^{0.38}$ = 2.5 * 2.65

Duration = 6.63 months

Staffing = effort/duration = 13.01/6.63

Staffing = 1.96 persons



Not secure | groups.umd.umich.edu/cis/course.des/cis525/js/f00/kutcher/kutcher.html

YOUR BASIC COCOMO RESULTS!!

MODE	"A" variable	"B" variable	"C" variable	"D" variable	KLOC	EFFORT, (in person/months)	DURATION, (in months)	STAFFING, (recommended)
organic	2.4	1.05	2.5	0.38	5	13.005580640812418	6.626883140669763	1.962548661978967

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!

Output (Calculation Semi-detached mode):

KLOC = 5

Effort = a * KLOC^{b} = 3 * 5$^{1.12}$ = 3 * 6.07

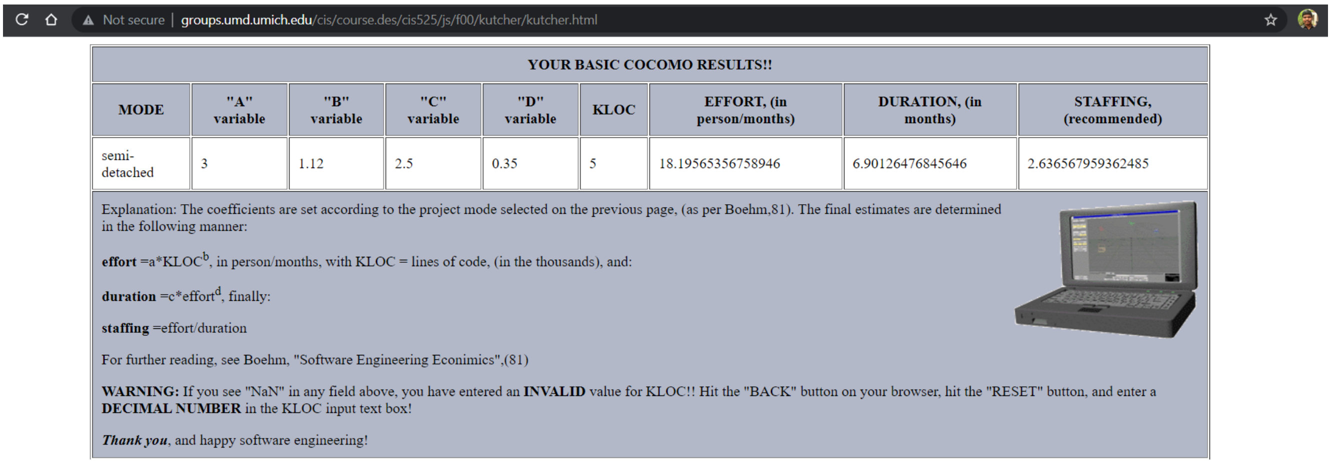
Effort = 18.21

Duration = c*Effort^{d} = 2.5 * 18.21$^{0.35}$ = 2.5 * 2.76

Duration = 6.9 months

Staffing = Effort/duration = 18.21/6.9

Staffing = 2.64 persons


Not secure | groups.umd.umich.edu/cis/course.des/cis525/js/f00/kutcher/kutcher.html

YOUR BASIC COCOMO RESULTS!!

MODE	"A" variable	"B" variable	"C" variable	"D" variable	KLOC	EFFORT, (in person/months)	DURATION, (in months)	STAFFING, (recommended)
semi-detached	3	1.12	2.5	0.35	5	18.19565356758946	6.90126476845646	2.636567959362485

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!

Output (Calculation Embedded mode):

KLOC = 5

Effort = a*KLOC^{b} =3.6 * 5$^{1.2}$ = 3.6 * 6.9

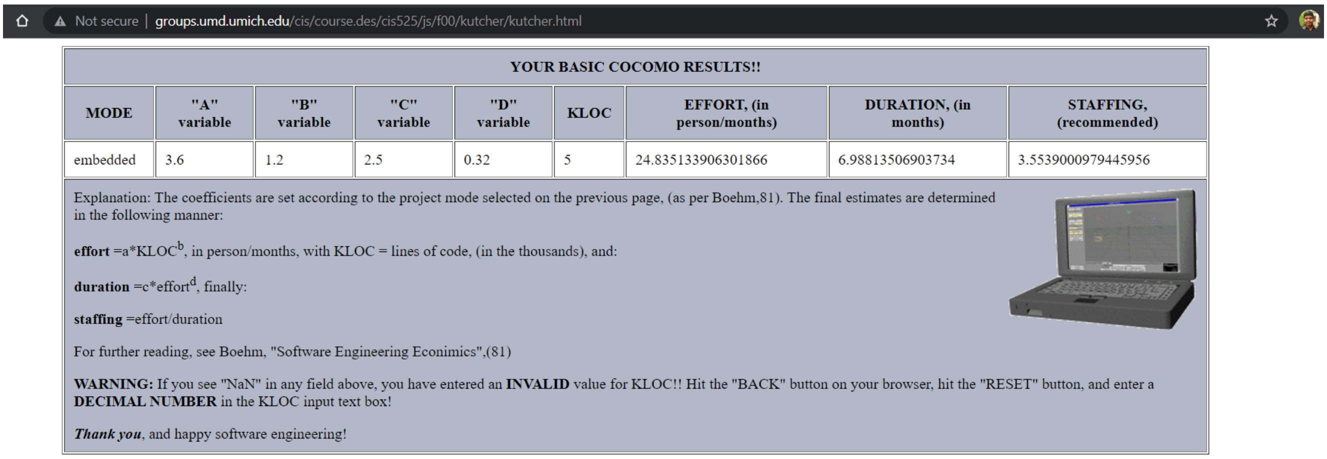
Effort = 24.84

Duration = c*effort^{d} = 2.5 * 24.84$^{0.32}$ = 2.5 * 2.8

Duration = 7 months

Staffing = Effort/duration = 24.84/7

Staffing = 3.55 persons


Not secure | groups.umd.umich.edu/cis/course.des/cis525/js/f00/kutcher/kutcher.html

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	5	24.835133906301866	6.98813506903734	3.5539000979445956

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!

According to the Domain characteristic table, Complexity adjustment table, and the COCOMO model, This E-Learning system has achieved the Functional point result of 1060.

And by using the Cocomo model this E-Learning platform achieved the organic values, semi detached values and the Embedded values successfully and got the required results.

RESULT:

The estimation of project was performed using function point analysis and COCOMO model.

Ex. No: 9	**AUTOMATION TESTING ON WEB PAGES USING SELENIUM**
Date: 14/10/2021	
Video Link:	https://drive.google.com/file/d/1PQADTm5pzTzgAwcskkI7ZJS-4OmUThoi/view?usp=sharing

OBJECTIVE

The objective of the experiment is to perform automation testing on a project web page E-Learning System using selenium framework on different browsers using Java/Python/C#

DESCRIPTION:

A webpage is to be exhaustively tested using different browsers using selenium framework. A webpage with multiple form fields (Text box, radio button, drop down buttons, buttons, hyperlink) is created and uploaded into a webserver (IIS/Apache). The content and the form fields in the webpage is individually checked. Also, the client-side validation performed using JavaScript is also to be evaluated. The evaluation is to be carried out by developing various test cases for every individual web component. To this end, an extensive software testing framework is required. Selenium is a free opensource automating and testing framework that supports C#, Java, Perl, PHP, Python and Ruby. Selenium Web driver provides the functionality of conducting various testcases.

PROCEDURE:

Server side:

- Load the designed Webpage to be tested in the server.
  (IIS Server- C:\inetpub\wwwroot)
  (Apache Server- ………………….)

Client side:

- Download all the necessary selenium packages based on the host machine requirement (windows/mac/linux, x86/x64)
- Download the corresponding browser's (Chrome/opera/firefox) web driver to integrate the IDE & browser.
- Using the IDE (Eclipse/Jupyter) create a new Java/Python program.
- Import all the required selenium packages in the java program.
- Using setProperty(), provide the name and location of the browser's webdriver.
- And a WebDriver class instances of the browser is created and it is used to access the webpage.
- The method navigate().to("URL") launches the website in the test browser.
- Every component in a webpage is accessed by the method findElement() and specific entities using the following
  o findElement(By.id(&quot;identifier value&quot;))
  o findElement(By.name(&quot;name&quot;))nd
  o findElement(By.linkText("Link text"))
  o findElement(By.className("class name"))
- Values are passed on using the method sendKeys("value").
- Inner HTML values are obtained using the method getText().
- Values from forms are obtained using the method getAttribute(&quot;value&quot;).
- Browser events such as click, clear, back, forward, refresh, and close are performed using the methods click(), clear(), back(), forward(), refresh(), quit() respectively.
- Also, moving between windows and frames are performed using the methods switchTo().window(&quot;windowName&quot;) and switchTo().frame(&quot;frameName&quot;);
- Javascript method is invoked by the method executeScript() of the class JavascriptExecuter

SOURCE CODE (Chrome Browser):

```
package selenium;

import java.util.concurrent.TimeUnit;

import org.openqa.selenium.By;
import org.openqa.selenium.JavascriptExecutor;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.chrome.ChromeOptions;
import org.openqa.selenium.support.ui.FluentWait;
import org.openqa.selenium.support.ui.Select;



public class chrome {

	public static void main(String[] args) throws InterruptedException {
		//Opening the chrome borwser
		System.setProperty("webdriver.chrome.driver", "D:\\Software-
Engineering-Lab\\Exp-9-Selenium\\chromedriver_win32\\chromedriver.exe");
		WebDriver webDriver = new ChromeDriver();

		//Maximize the Window
		webDriver.manage().window().maximize();

		//Navigate to the URL
		webDriver.get("https://www.rubangino.in/EXP-9/index.html");

		System.out.println("\nWeb Testing Started Successfully!!");

		System.out.println("\nTEST CASES");

		//Test Case 1: Verify url getCurrentURL()
		System.out.println("\nTest Case 1: Verifying the URL");
		String desiredURL = "https://www.rubangino.in/EXP-9/index.html";
		String acutalURL = webDriver.getCurrentUrl();
		System.out.println("Desired URL: " + desiredURL);
		System.out.println("Acutal URL: " + acutalURL);

		if (acutalURL.equals(desiredURL)) {
			System.out.println("Status: URL is Correct");
			System.out.println("Summary -> Test Case 1: Passed");
		}
		else {
			System.out.println("Status: URL is Wrong");
			System.out.println("Summary -> Test Case 1: Failed");
		}
		System.out.println();


		Thread.sleep(3000);


		//Test Case 2: Verify the Page title
		System.out.println("Test Case 2: Verifying the Page TITLE");
```

```
             String actualTitle = webDriver.getTitle();
             String desiredTitle = "E-Learning System";
             System.out.println("Actual TITLE: " + actualTitle);
             System.out.println("Desired TITLE: " + desiredTitle);

             if(actualTitle.equals(desiredTitle)) {
                   System.out.println("Status: Title is Correct");
                   System.out.println("Summary -> Test Case 2: Passed");
             }
             else {
                   System.out.println("Status: Title is Wrong");
                   System.out.println("Summary -> Test Case 2: Failed");
             }
             System.out.println();


             Thread.sleep(3000);


             //Test Case 3: Testing Home page Register Button
             System.out.println("Test Case 3: Testing Register Button");
             webDriver.findElement(By.id("register-btn")).click();

             String desiredRegURL = "https://www.rubangino.in/EXP-
9/index.html#register";

             String reg_URL = webDriver.getCurrentUrl();
             System.out.println("Actual URL: " + reg_URL);
             System.out.println("Desired URL: " + desiredRegURL);

             if(reg_URL.equals(desiredURL)) {
                   System.out.println("Register button is working Fine.");
                   System.out.println("Summary -> Test Case 3: Passed");
             }
             else {
                   System.out.println("Summary -> Test Case 3: Failed");
             }
             System.out.println();


             Thread.sleep(3000);


             //Test Case 4: Testing About Navigation button
             System.out.println("Test Case 4: Testing About Navigation Button");
             webDriver.findElement(By.id("about-btn")).click();

             String desiredAboutURL = "https://www.rubangino.in/EXP-
9/index.html#about";

             String about_URL = webDriver.getCurrentUrl();
             System.out.println("Actual URL: " + about_URL);
             System.out.println("Desired URL: " + desiredAboutURL);

             if(about_URL.equals(desiredAboutURL)) {
                   System.out.println("About Button Working Successfull");
                   System.out.println("Summary -> Test Case 4: Passed");
             }
             else {
                    System.out.println("Summary -> Test Case 4: Failed");
             }
             System.out.println();


             Thread.sleep(3000);


             //Test Case 5: Testing Register Navigation button
             System.out.println("Test Case 5: Testing Register Navigation 
Button");
             webDriver.findElement(By.id("register-btn")).click();

             String desiredRegisterURL = "https://www.rubangino.in/EXP-
9/index.html#register";

             String reg_nav_URL = webDriver.getCurrentUrl();
             System.out.println("Actual URL: " + reg_nav_URL);
             System.out.println("Desired URL: " + desiredRegisterURL);

             if(reg_nav_URL.equals(desiredRegisterURL)) {
                    System.out.println("Register Nav Button Working Successfull");
                    System.out.println("Summary -> Test Case 5: Passed");
             }
             else {
                    System.out.println("Summary -> Test Case 5: Failed");
             }
             System.out.println();


             Thread.sleep(3000);


             //Test Case 6: Testing Contact Navigation button
             System.out.println("Test Case 6: Testing Contact Navigation 
Button");
             webDriver.findElement(By.id("contact-btn")).click();

             String desiredContactURL = "https://www.rubangino.in/EXP-
9/index.html#contact";

             String contactURL = webDriver.getCurrentUrl();
             System.out.println("Actual URL: " + contactURL);
             System.out.println("Desired URL: " + desiredContactURL);

             if(contactURL.equals(desiredContactURL)) {
                    System.out.println("Register Nav Button Working Successfull");
                    System.out.println("Summary -> Test Case 6: Passed");
             }
             else {
                    System.out.println("Summary -> Test Case 6: Failed");
             }
             System.out.println();


             Thread.sleep(3000);


             //Testing Registration Form
```

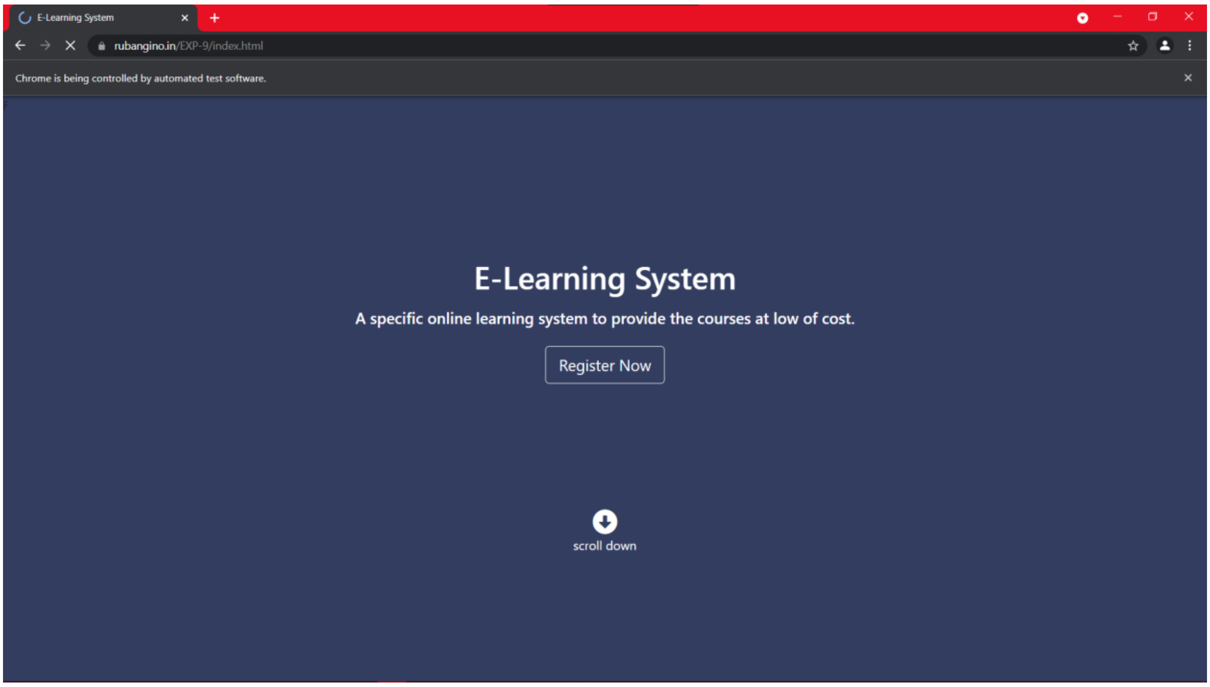
```
//Test Case 7: Testing First Name
webDriver.get("https://www.rubangino.in/EXP-9/index.html#register");

System.out.println("Test Case 7: Testing First Name");
WebElement firstName = webDriver.findElement((By.id("fname")));
firstName.sendKeys("Ruban Gino");

String string1 = firstName.getAttribute("value");
System.out.println("Input: " + string1);

if(string1.length() <= 10) {
        System.out.println("Summary -> Test Case 7: Passed");
}
else {
        System.out.println("Summary -> Test Case 7: Failed");
}
System.out.println();


Thread.sleep(3000);


//Test Case 8: Testing Last Name
System.out.println("Test Case 8: Testing Last Name");
WebElement lastName = webDriver.findElement((By.id("lname")));
lastName.sendKeys("Singh");

String string2 = lastName.getAttribute("value");
System.out.println("Input: " + string2);

if(string2.length() <= 10) {
        System.out.println("Summary -> Test Case 8: Passed");
}
else {
        System.out.println("Summary -> Test Case 8: Failed");
}
System.out.println();


Thread.sleep(3000);


//Test Case 9: Testing User Name
System.out.println("Test Case 9: Testing User Name");
WebElement userName = webDriver.findElement((By.id("username")));
userName.sendKeys("rubangino");

String string3 = userName.getAttribute("value");
System.out.println("Input: " + string3);

if(string3.length() !=0 ) {
        System.out.println("Summary -> Test Case 9: Passed");
}
else {
        System.out.println("Summary -> Test Case 9: Failed");
}
System.out.println();
```

```
            Thread.sleep(3000);


            //Test Case 10: Testing Email id
            System.out.println("Test Case 10: Testing Email Id");
            WebElement emailId = webDriver.findElement((By.id("email")));
            emailId.sendKeys("info@rubangino.in");

            String string4 = emailId.getAttribute("value");
            System.out.println("Input: " + string4);

            if(string4.contains("@") && string4.contains(".")) {
                  System.out.println("Summary -> Test Case 10: Passed");
            }
            else {
                  System.out.println("Summary -> Test Case 10: Failed");
            }
            System.out.println();


            Thread.sleep(3000);


            //Test Case 11: Testing Create Password
            System.out.println("Test Case 11: Testing Create Password Input");
            WebElement createPassword = 
webDriver.findElement((By.id("createPassword")));
            createPassword.sendKeys("E-Learning@#$System!@#$");

            String string5 = createPassword.getAttribute("value");
            System.out.println("Input: " + string5);

            if(string5.length() >= 8 && string5.length() <= 20) {
                  System.out.println("Summary -> Test Case 11: Passed");
            }
            else {
                  System.out.println("Summary -> Test Case 11: Failed");
            }
            System.out.println();


            Thread.sleep(3000);


            //Test Case 12: Testing Re-Type Password
            System.out.println("Test Case 12: Testing Re Type Password Input");
            WebElement reTypePassword = 
webDriver.findElement((By.id("reTypePassword")));
            reTypePassword.sendKeys("E-Learning@#$System!@#$");

            String string6 = createPassword.getAttribute("value");
            System.out.println("Input: " + string6);

            if(string5 == string6) {
                  System.out.println("Summary -> Test Case 12: Passed");
            }
            else {
                  System.out.println("Summary -> Test Case 12: Failed");
            }
```

```java
System.out.println();


Thread.sleep(3000);


//Test Case 13: Testing Country
System.out.println("Test Case 13: Testing Country Input");
WebElement countryInput = webDriver.findElement((By.id("country")));
countryInput.sendKeys("India");

String string7 = countryInput.getAttribute("value");
System.out.println("Input: " + string7);

if(string7.length() <= 15 && string7 != "!@#$%^&*()_-=<>?{}|][:;") {
	System.out.println("Summary -> Test Case 13: Passed");
}
else {
	System.out.println("Summary -> Test Case 13: Failed");
}
System.out.println();


Thread.sleep(3000);


//Test Case 14: Testing Profession DropDown
System.out.println("Test Case 14: Testing Drop Down");
WebElement profession = webDriver.findElement(By.id("profession"));
Select dropdown = new Select(profession);

dropdown.selectByVisibleText("Student");

String string9 = profession.getAttribute("value");
System.out.println("Input: " + string9);

String student = "Student";
String instructor = "Instructor";

if(string9.equals(student) || string9.equals(instructor)) {
	System.out.println("Summary -> Test Case 14: Passed");
}
else {
	System.out.println("Summary -> Test Case 14: Failed");
}
System.out.println();


Thread.sleep(3000);


//Test Case 15: Testing phone number
System.out.println("Test Case 15: Testing Phone Number");
WebElement phoneNumber = webDriver.findElement((By.id("phone")));
phoneNumber.sendKeys("+919344232829");

String string8 = phoneNumber.getAttribute("value");
System.out.println("Input: " + string8);
```

```
        if(string8.contains("+") && string8.length() <= 13) {
            System.out.println("Summary -> Test Case 15: Passed");
        }
        else {
            System.out.println("Summary -> Test Case 15: Failed");
        }
        System.out.println();


        Thread.sleep(3000);


        //Test Case 16: Testing Check Boxes
        System.out.println("Test Case 16: Testing Terms Check Box");
        webDriver.findElement(By.xpath("//*[@id=\"terms\"]")).click();


    if(webDriver.findElement(By.xpath("//*[@id=\"terms\"]")).isSelected()) {
            System.out.println("Summary -> Test Case 16: Passed");
        }
        else {
            System.out.println("Summary -> Test Case 16: Failed");
        }

        webDriver.findElement(By.id("submit-button")).click();

        Thread.sleep(3000);

        //Getting Alert
        System.out.println();
        String AlertText = webDriver.switchTo().alert().getText();
        webDriver.switchTo().alert().accept();
        System.out.println("Alert text is: " + AlertText);
        System.out.println("\nAlert Accepted!!");

        Thread.sleep(5000);


        //Close the browser
        webDriver.close();
        System.out.println("\nWeb Testing Ended Successfully!");
    }

}
```

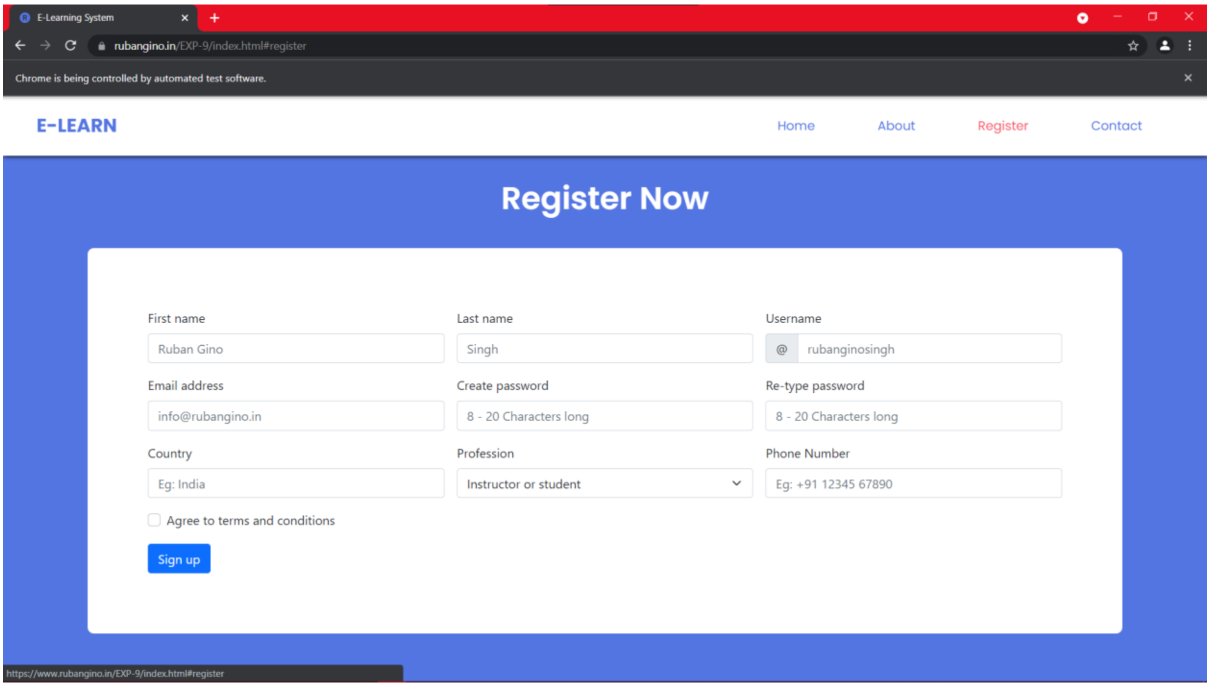
OUTPUT (Chrome Browser):



E-Learning System
rubangino.in/EXP-9/index.html
Chrome is being controlled by automated test software.

E-Learning System

A specific online learning system to provide the courses at low of cost.



E-Learning System
rubangino.in/EXP-9/index.html#register
Chrome is being controlled by automated test software.
E-LEARN
Home
About
Register
Contact

Register Now

First name: Ruban Gino

Last name: Singh

Username: @ rubanginosingh

Email address: info@rubangino.in

Create password: 8 - 20 Characters long

Re-type password: 8 - 20 Characters long

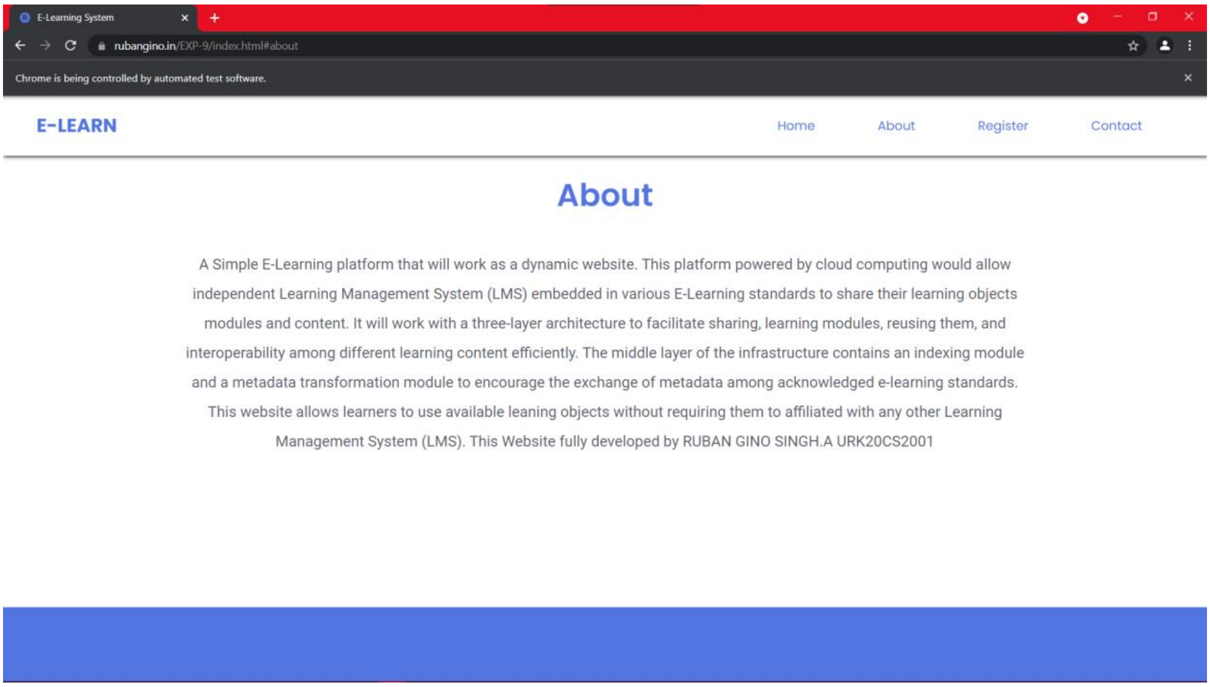
Country: Eg: India

Profession: Instructor or student

Phone Number: Eg: +91 12345 67890

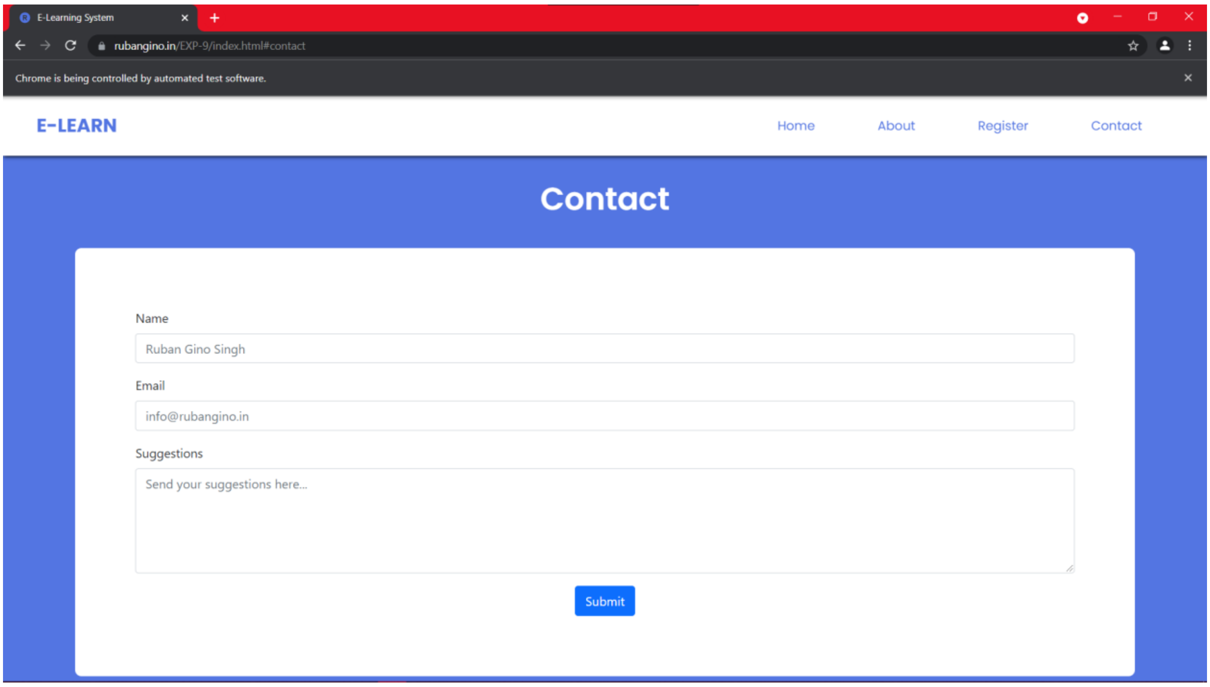
□ Agree to terms and conditions

Sign up
https://www.rubangino.in/EXP-9/index.html#register


E-Learning System
rubangino.in/EXP-9/index.html#about
Chrome is being controlled by automated test software.
E-LEARN
Home
About
Register
Contact

About

A Simple E-Learning platform that will work as a dynamic website. This platform powered by cloud computing would allow independent Learning Management System (LMS) embedded in various E-Learning standards to share their learning objects modules and content. It will work with a three-layer architecture to facilitate sharing, learning modules, reusing them, and interoperability among different learning content efficiently. The middle layer of the infrastructure contains an indexing module and a metadata transformation module to encourage the exchange of metadata among acknowledged e-learning standards. This website allows learners to use available leaning objects without requiring them to affiliated with any other Learning Management System (LMS). This Website fully developed by RUBAN GINO SINGH.A URK20CS2001

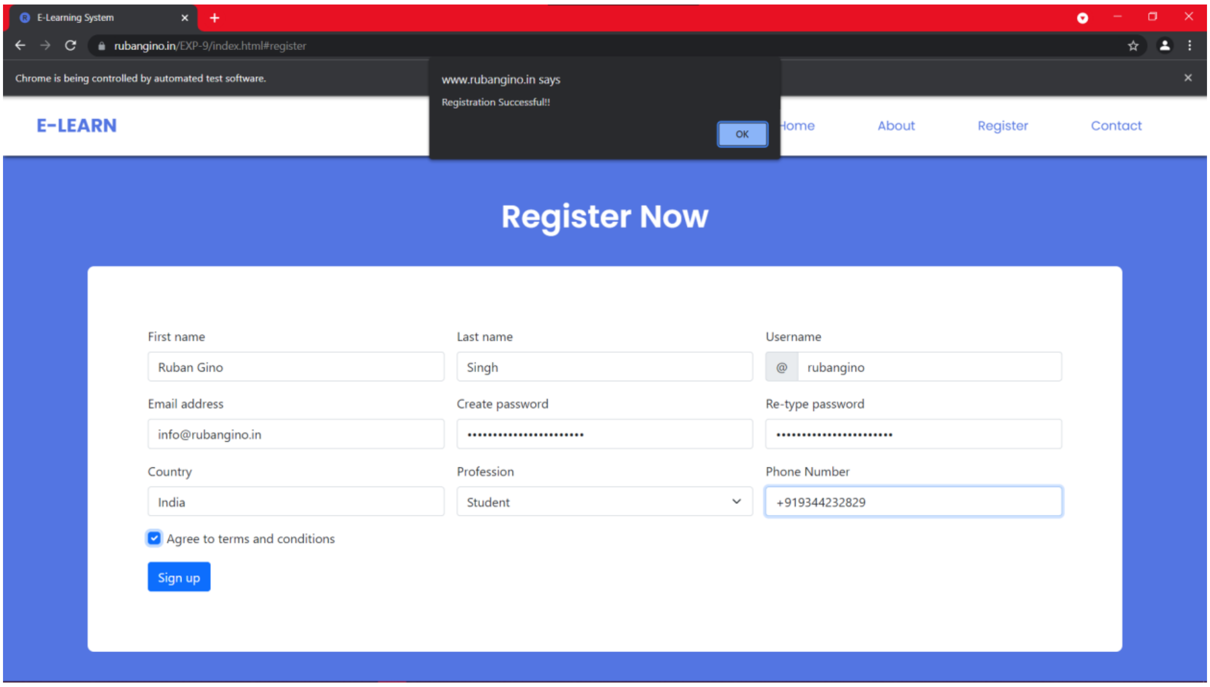

E-Learning System
rubangino.in/EXP-9/index.html#contact
Chrome is being controlled by automated test software.
E-LEARN
Home
About
Register
Contact
Submit

Contact

Name: Ruban Gino Singh

Email: info@rubangino.in

Suggestions: Send your suggestions here...



Chrome is being controlled by automated test software.

www.rubangino.in says

Registration Successful!!

OK

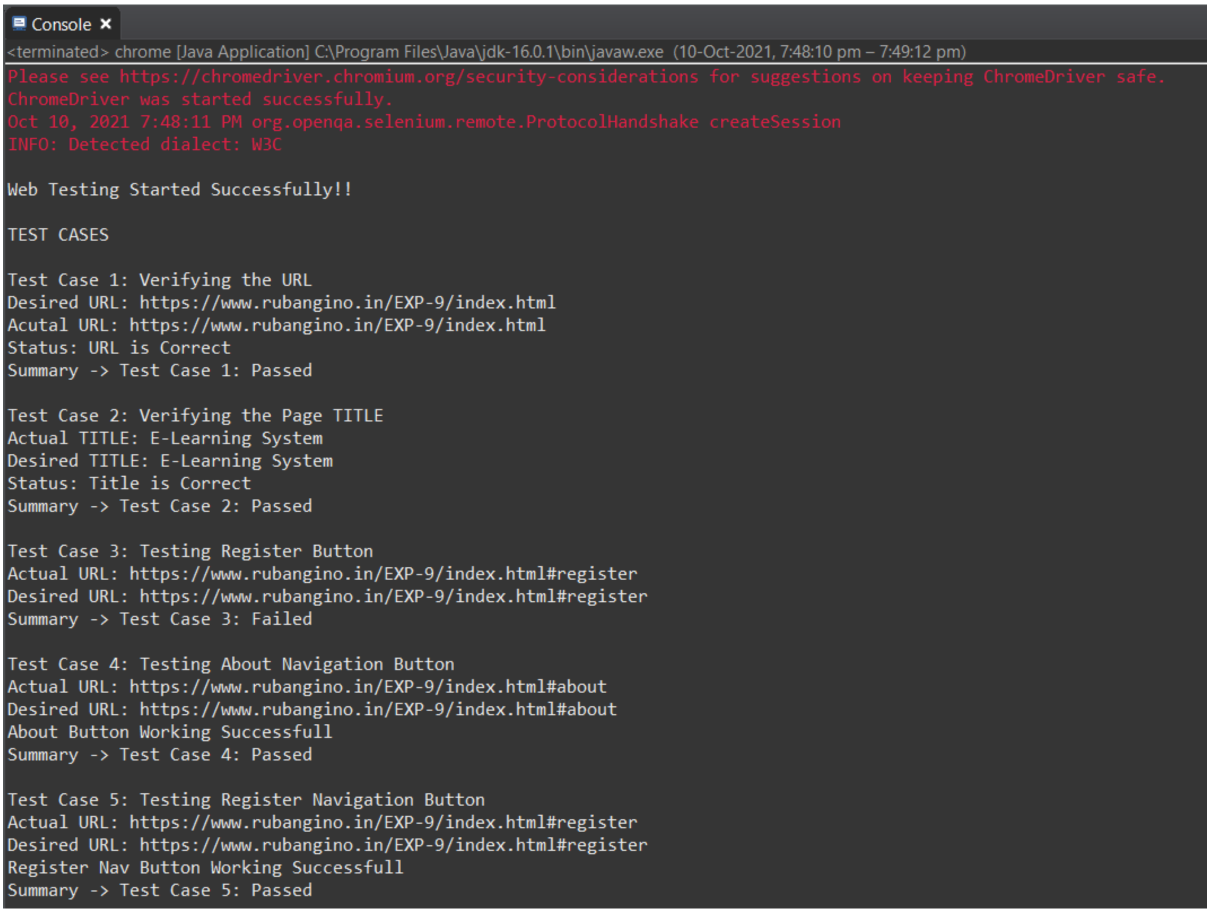
Register Now

First name	Last name	Username
Ruban Gino	Singh	@ rubangino
Email address	Create password	Re-type password
info@rubangino.in	••••••••••••••••••••••••	••••••••••••••••••••••••
Country	Profession	Phone Number
India	Student	+919344232829

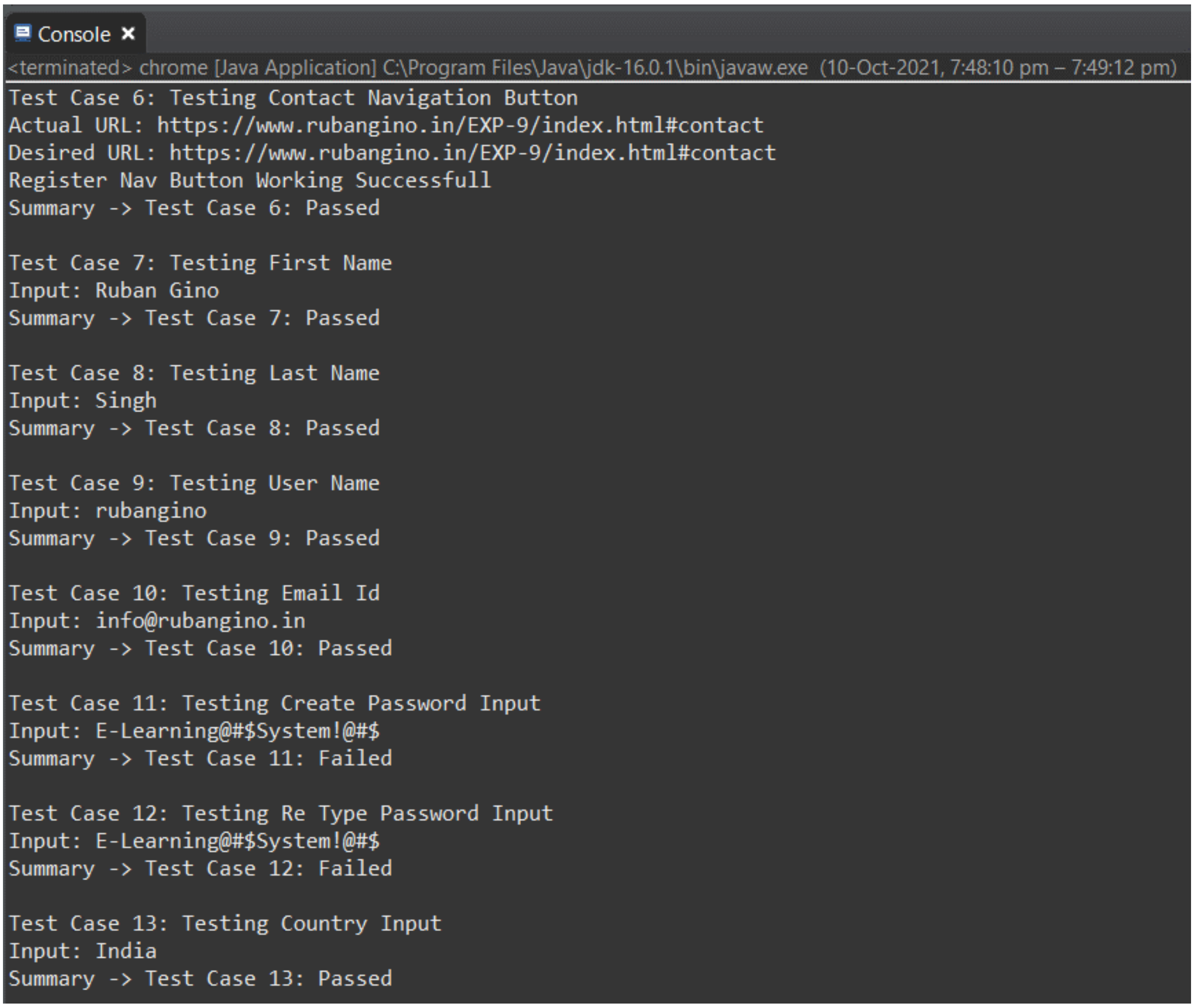
☑ Agree to terms and conditions

Sign up

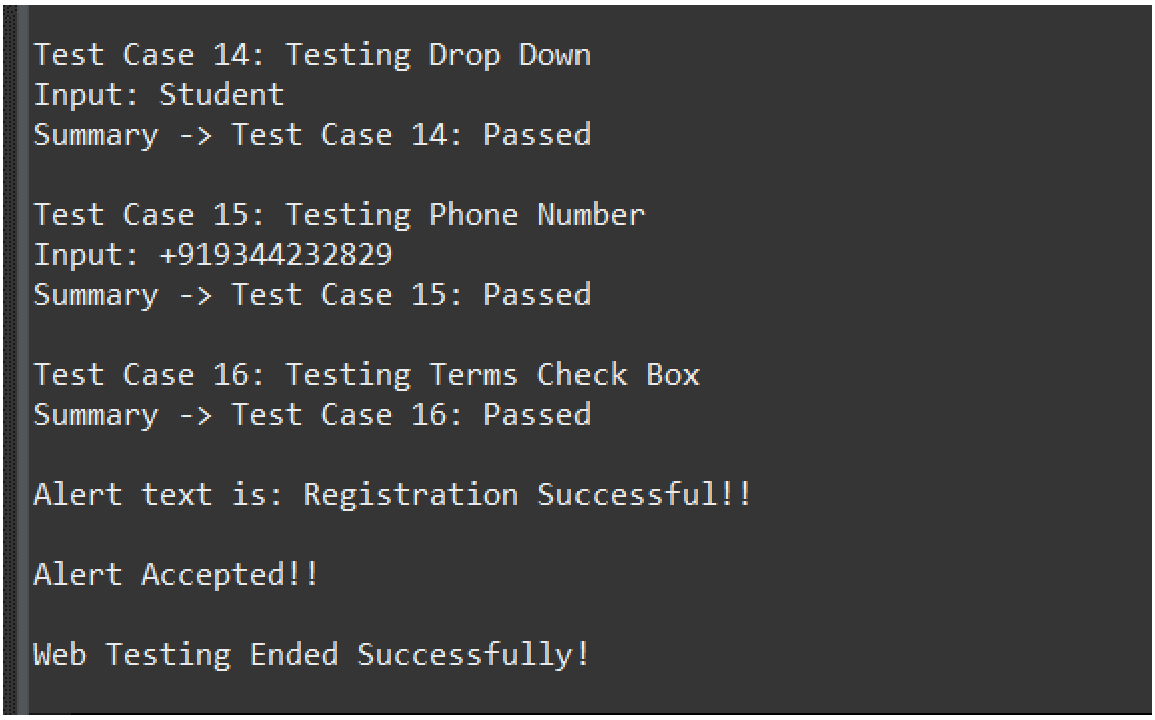
Browser: Chrome Browser



```
<terminated> chrome [Java Application] C:\Program Files\Java\jdk-16.0.1\bin\javaw.exe  (10-Oct-2021, 7:48:10 pm – 7:49:12 pm)
Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver safe.
ChromeDriver was started successfully.
Oct 10, 2021 7:48:11 PM org.openqa.selenium.remote.ProtocolHandshake createSession
INFO: Detected dialect: W3C

Web Testing Started Successfully!!

TEST CASES

Test Case 1: Verifying the URL
Desired URL: https://www.rubangino.in/EXP-9/index.html
Acutal URL: https://www.rubangino.in/EXP-9/index.html
Status: URL is Correct
Summary -> Test Case 1: Passed

Test Case 2: Verifying the Page TITLE
Actual TITLE: E-Learning System
Desired TITLE: E-Learning System
Status: Title is Correct
Summary -> Test Case 2: Passed

Test Case 3: Testing Register Button
Actual URL: https://www.rubangino.in/EXP-9/index.html#register
Desired URL: https://www.rubangino.in/EXP-9/index.html#register
Summary -> Test Case 3: Failed

Test Case 4: Testing About Navigation Button
Actual URL: https://www.rubangino.in/EXP-9/index.html#about
Desired URL: https://www.rubangino.in/EXP-9/index.html#about
About Button Working Successfull
Summary -> Test Case 4: Passed

Test Case 5: Testing Register Navigation Button
Actual URL: https://www.rubangino.in/EXP-9/index.html#register
Desired URL: https://www.rubangino.in/EXP-9/index.html#register
Register Nav Button Working Successfull
Summary -> Test Case 5: Passed
```



```
<terminated> chrome [Java Application] C:\Program Files\Java\jdk-16.0.1\bin\javaw.exe  (10-Oct-2021, 7:48:10 pm – 7:49:12 pm)
Test Case 6: Testing Contact Navigation Button
Actual URL: https://www.rubangino.in/EXP-9/index.html#contact
Desired URL: https://www.rubangino.in/EXP-9/index.html#contact
Register Nav Button Working Successfull
Summary -> Test Case 6: Passed

Test Case 7: Testing First Name
Input: Ruban Gino
Summary -> Test Case 7: Passed

Test Case 8: Testing Last Name
Input: Singh
Summary -> Test Case 8: Passed

Test Case 9: Testing User Name
Input: rubangino
Summary -> Test Case 9: Passed

Test Case 10: Testing Email Id
Input: info@rubangino.in
Summary -> Test Case 10: Passed

Test Case 11: Testing Create Password Input
Input: E-Learning@#$System!@#$
Summary -> Test Case 11: Failed

Test Case 12: Testing Re Type Password Input
Input: E-Learning@#$System!@#$
Summary -> Test Case 12: Failed

Test Case 13: Testing Country Input
Input: India
Summary -> Test Case 13: Passed
```



```
Test Case 14: Testing Drop Down
Input: Student
Summary -> Test Case 14: Passed

Test Case 15: Testing Phone Number
Input: +919344232829
Summary -> Test Case 15: Passed

Test Case 16: Testing Terms Check Box
Summary -> Test Case 16: Passed

Alert text is: Registration Successful!!

Alert Accepted!!

Web Testing Ended Successfully!
```

SOURCE CODE (Opera Browser):

```
package selenium;

import java.util.concurrent.TimeUnit;

import org.openqa.selenium.By;
import org.openqa.selenium.JavascriptExecutor;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.chrome.ChromeOptions;
import org.openqa.selenium.opera.OperaDriver;
import org.openqa.selenium.support.ui.FluentWait;
import org.openqa.selenium.support.ui.Select;


//import com.sun.org.apache.bcel.internal.generic.Select;

public class opera {

	public static void main(String[] args) throws InterruptedException {
		//Opening the Opera borwser
		System.setProperty("webdriver.opera.driver", "D:\\Software-
Engineering-Lab\\Exp-9-
Selenium\\operadriver_win64\\operadriver_win64\\operadriver.exe");
		WebDriver webDriver = new OperaDriver();

		//Maximize the Window
		webDriver.manage().window().maximize();

		//Navigate to the URL
		webDriver.get("https://www.rubangino.in/EXP-9/index.html");

		System.out.println("\nWeb Testing Started Successfully!!");

		System.out.println("\nTEST CASES");

		//Test Case 1: Verify url getCurrentURL()
		System.out.println("\nTest Case 1: Verifying the URL");
		String desiredURL = "https://www.rubangino.in/EXP-9/index.html";
		String acutalURL = webDriver.getCurrentUrl();
		System.out.println("Desired URL: " + desiredURL);
		System.out.println("Acutal URL: " + acutalURL);

		if (acutalURL.equals(desiredURL)) {
			System.out.println("Status: URL is Correct");
			System.out.println("Summary -> Test Case 1: Passed");
		}
		else {
			System.out.println("Status: URL is Wrong");
			System.out.println("Summary -> Test Case 1: Failed");
		}
		System.out.println();
```

```
            Thread.sleep(3000);


            //Test Case 2: Verify the Page title
            System.out.println("Test Case 2: Verifying the Page TITLE");
            String actualTitle = webDriver.getTitle();
            String desiredTitle = "E-Learning System";
            System.out.println("Actual TITLE: " + actualTitle);
            System.out.println("Desired TITLE: " + desiredTitle);

            if(actualTitle.equals(desiredTitle)) {
                  System.out.println("Status: Title is Correct");
                  System.out.println("Summary -> Test Case 2: Passed");
            }
            else {
                  System.out.println("Status: Title is Wrong");
                  System.out.println("Summary -> Test Case 2: Failed");
            }
            System.out.println();


            Thread.sleep(3000);


            //Test Case 3: Testing Home page Register Button
            System.out.println("Test Case 3: Testing Register Button");
            webDriver.findElement(By.id("register-btn")).click();

            String desiredRegURL = "https://www.rubangino.in/EXP-
9/index.html#register";

            String reg_URL = webDriver.getCurrentUrl();
            System.out.println("Actual URL: " + reg_URL);
            System.out.println("Desired URL: " + desiredRegURL);

            if(reg_URL.equals(desiredURL)) {
                  System.out.println("Register button is working Fine.");
                  System.out.println("Summary -> Test Case 3: Passed");
            }
            else {
                  System.out.println("Summary -> Test Case 3: Failed");
            }
            System.out.println();


            Thread.sleep(3000);


            //Test Case 4: Testing About Navigation button
            System.out.println("Test Case 4: Testing About Navigation Button");
            webDriver.findElement(By.id("about-btn")).click();

            String desiredAboutURL = "https://www.rubangino.in/EXP-
9/index.html#about";

            String about_URL = webDriver.getCurrentUrl();
            System.out.println("Actual URL: " + about_URL);
            System.out.println("Desired URL: " + desiredAboutURL);
```

```
             if(about_URL.equals(desiredAboutURL)) {
                    System.out.println("About Button Working Successfull");
                    System.out.println("Summary -> Test Case 4: Passed");
             }
             else {
                    System.out.println("Summary -> Test Case 4: Failed");
             }
             System.out.println();


             Thread.sleep(3000);


             //Test Case 5: Testing Register Navigation button
             System.out.println("Test Case 5: Testing Register Navigation
Button");
             webDriver.findElement(By.id("register-btn")).click();

             String desiredRegisterURL = "https://www.rubangino.in/EXP-
9/index.html#register";

             String reg_nav_URL = webDriver.getCurrentUrl();
             System.out.println("Actual URL:  " + reg_nav_URL);
             System.out.println("Desired URL:  " + desiredRegisterURL);

             if(reg_nav_URL.equals(desiredRegisterURL)) {
                    System.out.println("Register Nav Button Working Successfull");
                    System.out.println("Summary -> Test Case 5: Passed");
             }
             else {
                    System.out.println("Summary -> Test Case 5: Failed");
             }
             System.out.println();


             Thread.sleep(3000);


             //Test Case 6: Testing Contact Navigation button
             System.out.println("Test Case 6: Testing Contact Navigation
Button");
             webDriver.findElement(By.id("contact-btn")).click();

             String desiredContactURL = "https://www.rubangino.in/EXP-
9/index.html#contact";

             String contactURL = webDriver.getCurrentUrl();
             System.out.println("Actual URL:  " + contactURL);
             System.out.println("Desired URL:  " + desiredContactURL);

             if(contactURL.equals(desiredContactURL)) {
                    System.out.println("Register Nav Button Working Successfull");
                    System.out.println("Summary -> Test Case 6: Passed");
             }
             else {
                    System.out.println("Summary -> Test Case 6: Failed");
             }
             System.out.println();
```

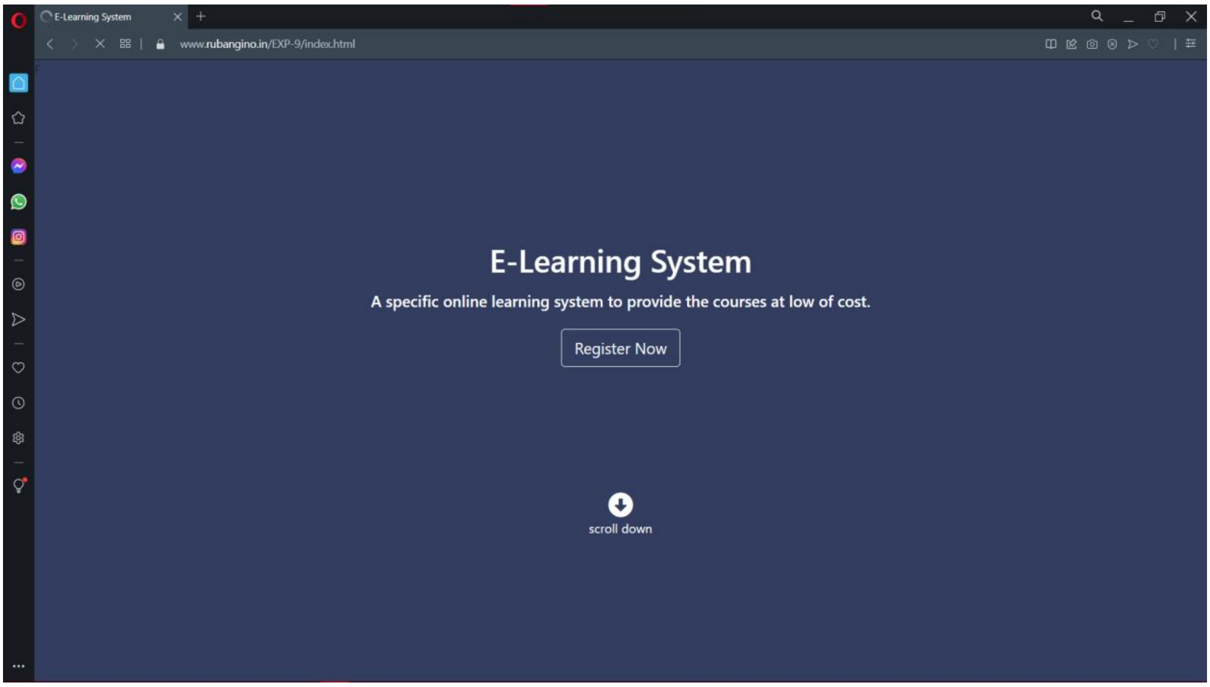
```java
Thread.sleep(3000);


//Testing Registration Form
//Test Case 7: Testing First Name
webDriver.get("https://www.rubangino.in/EXP-9/index.html#register");

System.out.println("Test Case 7: Testing First Name");
WebElement firstName = webDriver.findElement((By.id("fname")));
firstName.sendKeys("Ruban Gino");

String string1 = firstName.getAttribute("value");
System.out.println("Input: " + string1);

if(string1.length() <= 10) {
        System.out.println("Summary -> Test Case 7: Passed");
}
else {
        System.out.println("Summary -> Test Case 7: Failed");
}
System.out.println();


Thread.sleep(3000);


//Test Case 8: Testing Last Name
System.out.println("Test Case 8: Testing Last Name");
WebElement lastName = webDriver.findElement((By.id("lname")));
lastName.sendKeys("Singh");

String string2 = lastName.getAttribute("value");
System.out.println("Input: " + string2);

if(string2.length() <= 10) {
        System.out.println("Summary -> Test Case 8: Passed");
}
else {
        System.out.println("Summary -> Test Case 8: Failed");
}
System.out.println();


Thread.sleep(3000);


//Test Case 9: Testing User Name
System.out.println("Test Case 9: Testing User Name");
WebElement userName = webDriver.findElement((By.id("username")));
userName.sendKeys("rubangino");

String string3 = userName.getAttribute("value");
System.out.println("Input: " + string3);

if(string3.length() !=0 ) {
        System.out.println("Summary -> Test Case 9: Passed");
}
```

```
            else {
                  System.out.println("Summary -> Test Case 9: Failed");
            }
            System.out.println();


            Thread.sleep(3000);


            //Test Case 10: Testing Email id
            System.out.println("Test Case 10: Testing Email Id");
            WebElement emailId = webDriver.findElement((By.id("email")));
            emailId.sendKeys("info@rubangino.in");

            String string4 = emailId.getAttribute("value");
            System.out.println("Input: " + string4);

            if(string4.contains("@") && string4.contains(".")) {
                  System.out.println("Summary -> Test Case 10: Passed");
            }
            else {
                  System.out.println("Summary -> Test Case 10: Failed");
            }
            System.out.println();


            Thread.sleep(3000);


            //Test Case 11: Testing Create Password
            System.out.println("Test Case 11: Testing Create Password Input");
            WebElement createPassword = 
webDriver.findElement((By.id("createPassword")));
            createPassword.sendKeys("E-Learning@#$System!@#$");

            String string5 = createPassword.getAttribute("value");
            System.out.println("Input: " + string5);

            if(string5.length() >= 8 && string5.length() <= 20) {
                  System.out.println("Summary -> Test Case 11: Passed");
            }
            else {
                  System.out.println("Summary -> Test Case 11: Failed");
            }
            System.out.println();


            Thread.sleep(3000);


            //Test Case 12: Testing Re-Type Password
            System.out.println("Test Case 12: Testing Re Type Password Input");
            WebElement reTypePassword = 
webDriver.findElement((By.id("reTypePassword")));
            reTypePassword.sendKeys("E-Learning@#$System!@#$");

            String string6 = createPassword.getAttribute("value");
            System.out.println("Input: " + string6);
```

```
if(string5 == string6) {
       System.out.println("Summary -> Test Case 12: Passed");
}
else {
       System.out.println("Summary -> Test Case 12: Failed");
}
System.out.println();


Thread.sleep(3000);


//Test Case 13: Testing Country
System.out.println("Test Case 13: Testing Country Input");
WebElement countryInput = webDriver.findElement((By.id("country")));
countryInput.sendKeys("India");

String string7 = countryInput.getAttribute("value");
System.out.println("Input: " + string7);

if(string7.length() <= 15 && string7 != "!@#$%^&*()_-=<>?{}|][:;") {
       System.out.println("Summary -> Test Case 13: Passed");
}
else {
       System.out.println("Summary -> Test Case 13: Failed");
}
System.out.println();


Thread.sleep(3000);


//Test Case 14: Testing Profession DropDown
System.out.println("Test Case 14: Testing Drop Down");
WebElement profession = webDriver.findElement(By.id("profession"));
Select dropdown = new Select(profession);

dropdown.selectByVisibleText("Student");

String string9 = profession.getAttribute("value");
System.out.println("Input: " + string9);

String student = "Student";
String instructor = "Instructor";

if(string9.equals(student) || string9.equals(instructor)) {
       System.out.println("Summary -> Test Case 14: Passed");
}
else {
       System.out.println("Summary -> Test Case 14: Failed");
}
System.out.println();


Thread.sleep(3000);


//Test Case 15: Testing phone number
System.out.println("Test Case 15: Testing Phone Number");
```

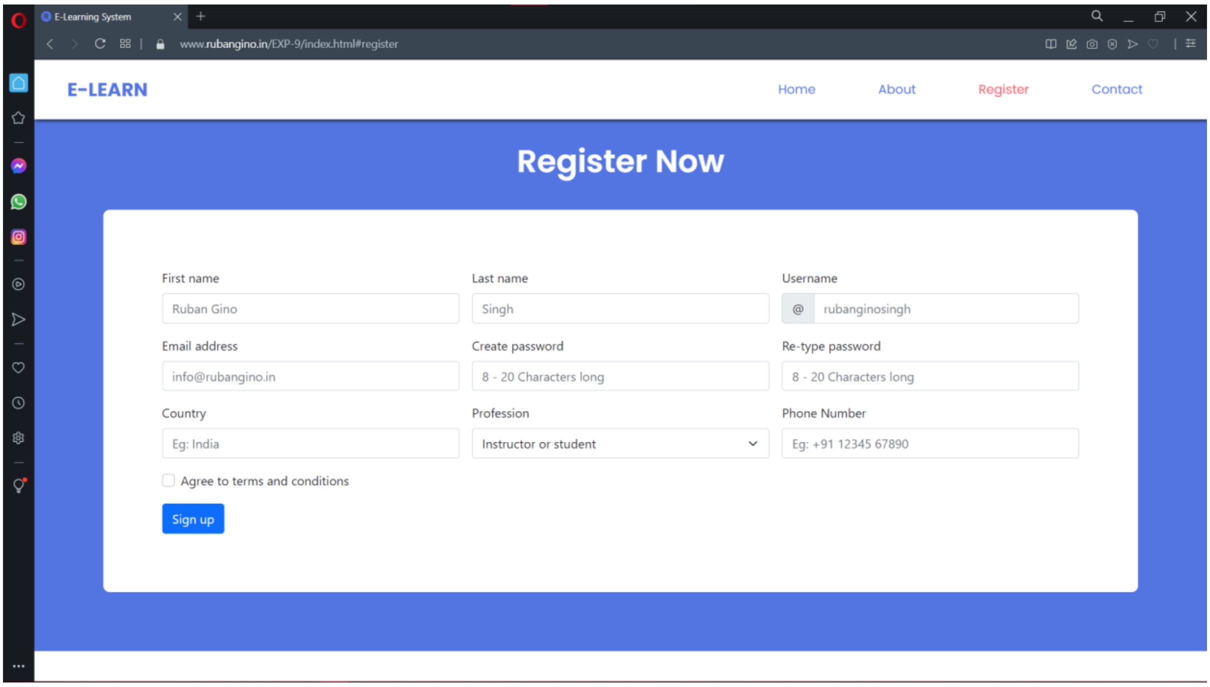
```
        WebElement phoneNumber = webDriver.findElement((By.id("phone")));
        phoneNumber.sendKeys("+919344232829");

        String string8 = phoneNumber.getAttribute("value");
        System.out.println("Input: " + string8);

        if(string8.contains("+") && string8.length() <= 13) {
            System.out.println("Summary -> Test Case 15: Passed");
        }
        else {
            System.out.println("Summary -> Test Case 15: Failed");
        }
        System.out.println();


        Thread.sleep(3000);


        //Test Case 16: Testing Check Boxes
        System.out.println("Test Case 16: Testing Terms Check Box");
        webDriver.findElement(By.xpath("//*[@id=\"terms\"]")).click();


    if(webDriver.findElement(By.xpath("//*[@id=\"terms\"]")).isSelected()) {
            System.out.println("Summary -> Test Case 16: Passed");
        }
        else {
            System.out.println("Summary -> Test Case 16: Failed");
        }

        webDriver.findElement(By.id("submit-button")).click();

        Thread.sleep(3000);

        //Getting Alert
        System.out.println();
        String AlertText = webDriver.switchTo().alert().getText();
        webDriver.switchTo().alert().accept();
        System.out.println("Alert text is: " + AlertText);
        System.out.println("\nAlert Accepted!!");

        Thread.sleep(5000);


        //Close the browser
        webDriver.close();
        System.out.println("\nWeb Testing Ended Successfully!");
    }

}
```

OUTPUT (Opera Browser):



E-Learning System

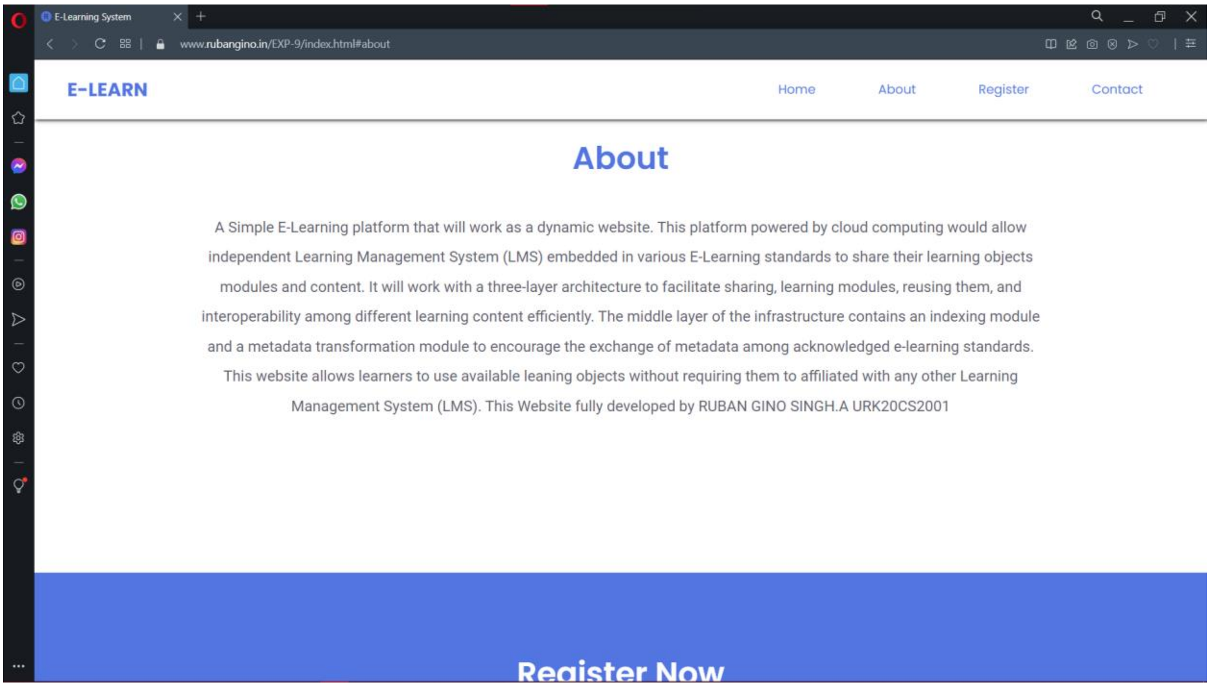
A specific online learning system to provide the courses at low of cost.



Register Now

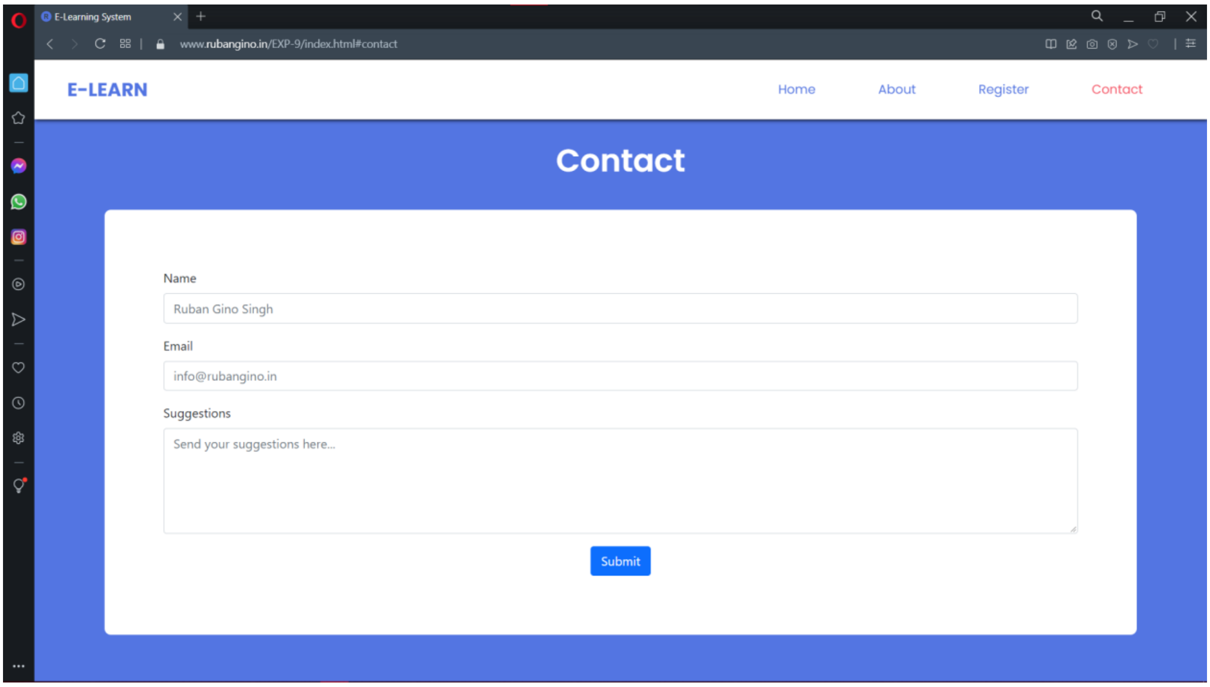
First name	Last name	Username
Ruban Gino	Singh	@ rubanginosingh
Email address	Create password	Re-type password
info@rubangino.in	8 - 20 Characters long	8 - 20 Characters long
Country	Profession	Phone Number
Eg: India	Instructor or student	Eg: +91 12345 67890

□ Agree to terms and conditions



About

A Simple E-Learning platform that will work as a dynamic website. This platform powered by cloud computing would allow independent Learning Management System (LMS) embedded in various E-Learning standards to share their learning objects modules and content. It will work with a three-layer architecture to facilitate sharing, learning modules, reusing them, and interoperability among different learning content efficiently. The middle layer of the infrastructure contains an indexing module and a metadata transformation module to encourage the exchange of metadata among acknowledged e-learning standards. This website allows learners to use available leaning objects without requiring them to affiliated with any other Learning Management System (LMS). This Website fully developed by RUBAN GINO SINGH.A URK20CS2001



Contact

Name

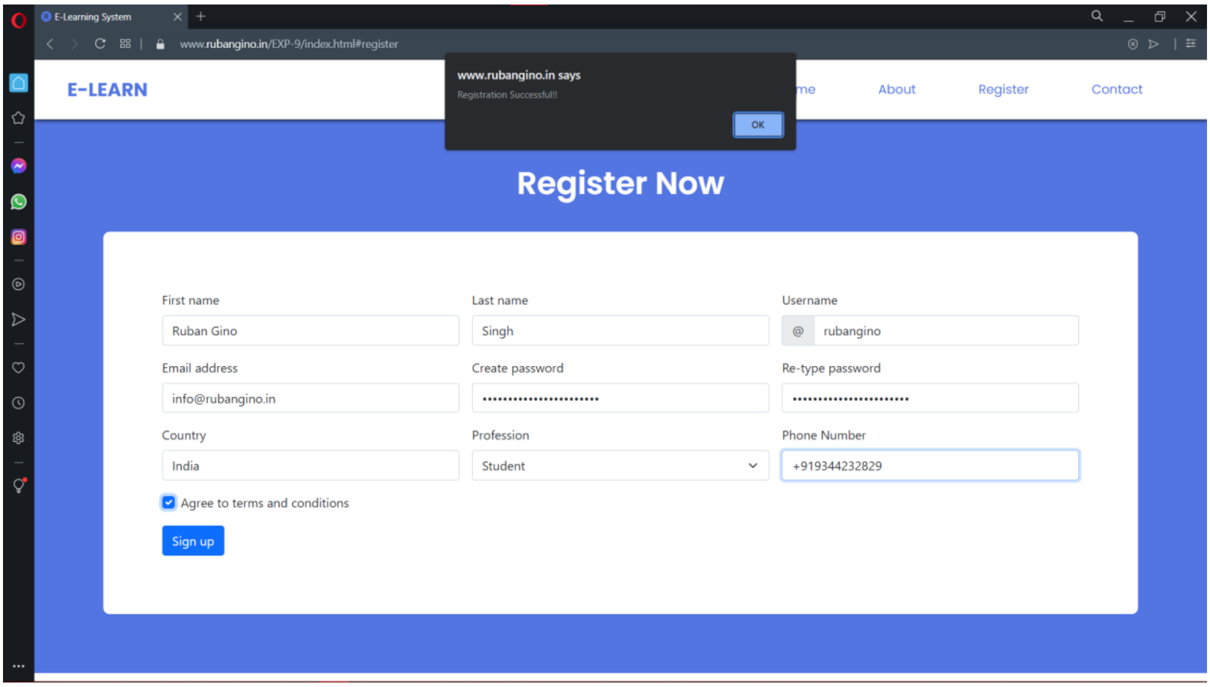
Ruban Gino Singh

Email

info@rubangino.in

Suggestions

Send your suggestions here...



www.rubangino.in says

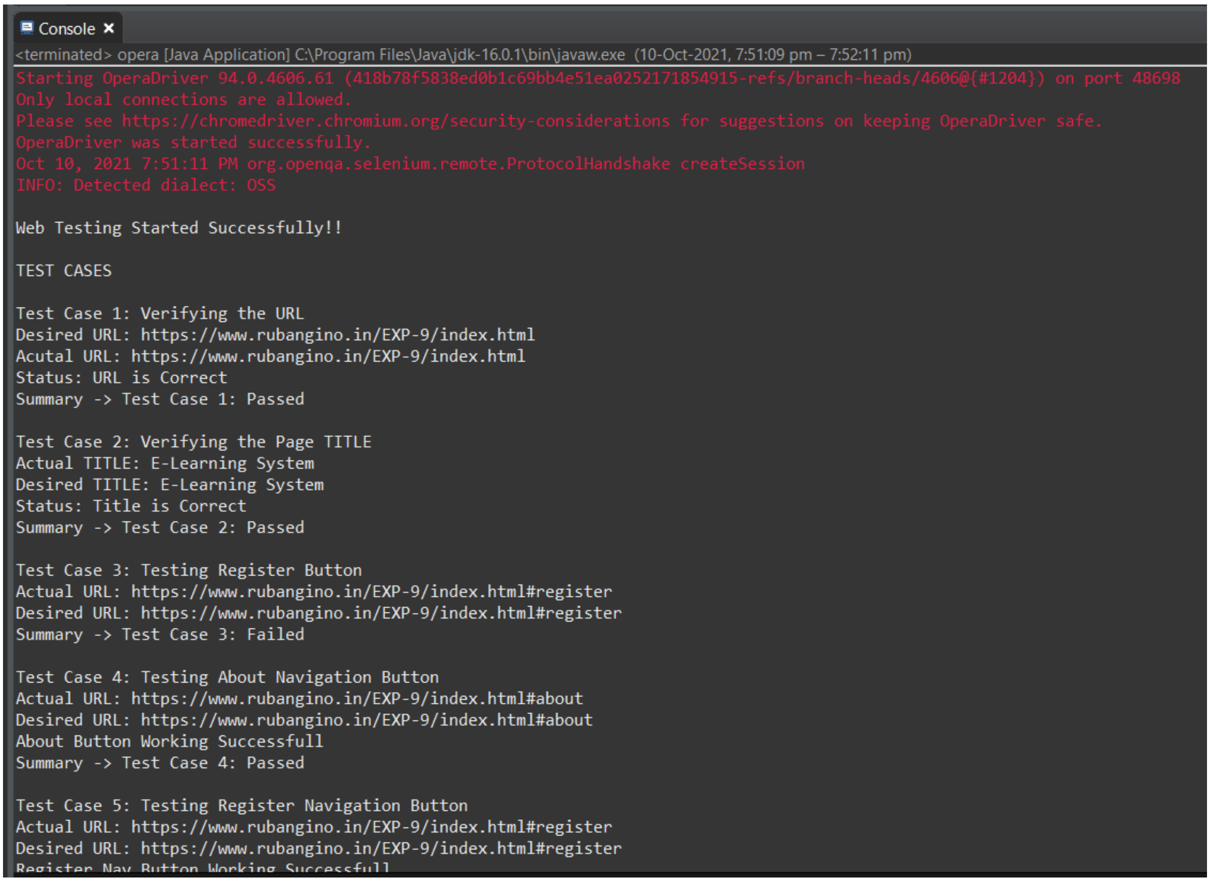
Registration Successfull!

Register Now

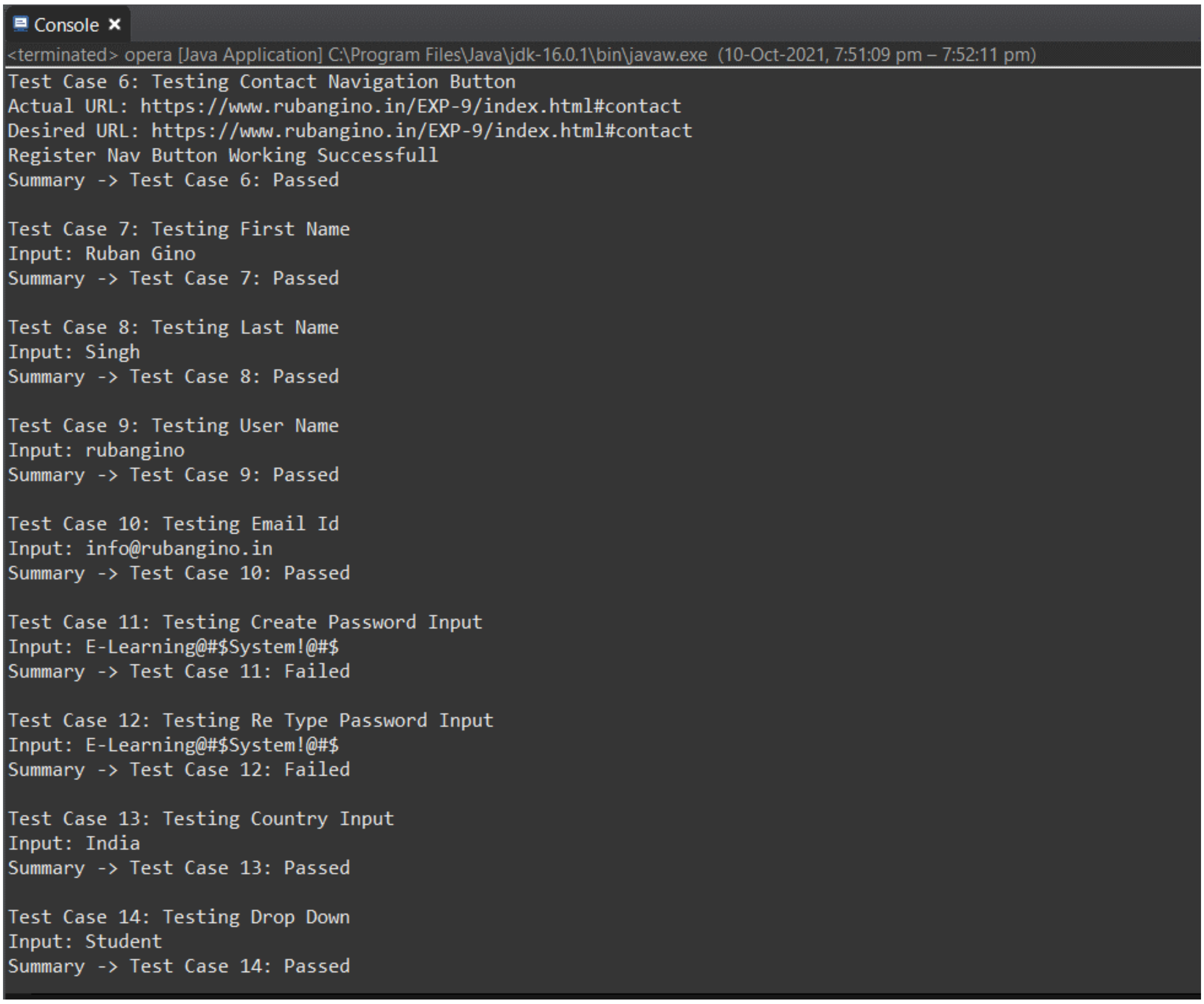
First name	Last name	Username
Ruban Gino	Singh	@ rubangino
Email address	Create password	Re-type password
info@rubangino.in	••••••••••••••••••••••••	••••••••••••••••••••••••
Country	Profession	Phone Number
India	Student	+919344232829

☑ Agree to terms and conditions

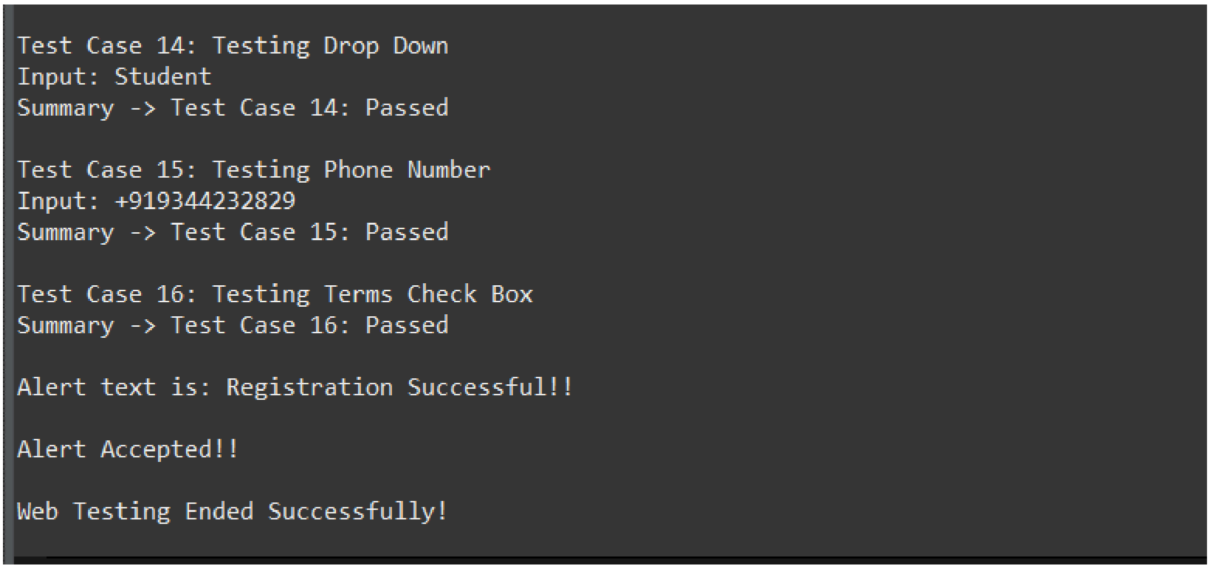
Browser: Opera Browser



```
<terminated> opera [Java Application] C:\Program Files\Java\jdk-16.0.1\bin\javaw.exe  (10-Oct-2021, 7:51:09 pm – 7:52:11 pm)
Starting OperaDriver 94.0.4606.61 (418b78f5838ed0b1c69bb4e51ea0252171854915-refs/branch-heads/4606@{#1204}) on port 48698
Only local connections are allowed.
Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping OperaDriver safe.
OperaDriver was started successfully.
Oct 10, 2021 7:51:11 PM org.openqa.selenium.remote.ProtocolHandshake createSession
INFO: Detected dialect: OSS

Web Testing Started Successfully!!

TEST CASES

Test Case 1: Verifying the URL
Desired URL: https://www.rubangino.in/EXP-9/index.html
Acutal URL: https://www.rubangino.in/EXP-9/index.html
Status: URL is Correct
Summary -> Test Case 1: Passed

Test Case 2: Verifying the Page TITLE
Actual TITLE: E-Learning System
Desired TITLE: E-Learning System
Status: Title is Correct
Summary -> Test Case 2: Passed

Test Case 3: Testing Register Button
Actual URL: https://www.rubangino.in/EXP-9/index.html#register
Desired URL: https://www.rubangino.in/EXP-9/index.html#register
Summary -> Test Case 3: Failed

Test Case 4: Testing About Navigation Button
Actual URL: https://www.rubangino.in/EXP-9/index.html#about
Desired URL: https://www.rubangino.in/EXP-9/index.html#about
About Button Working Successfull
Summary -> Test Case 4: Passed

Test Case 5: Testing Register Navigation Button
Actual URL: https://www.rubangino.in/EXP-9/index.html#register
Desired URL: https://www.rubangino.in/EXP-9/index.html#register
Register Nav Button Working Successfull
```



```
<terminated> opera [Java Application] C:\Program Files\Java\jdk-16.0.1\bin\javaw.exe  (10-Oct-2021, 7:51:09 pm – 7:52:11 pm)
Test Case 6: Testing Contact Navigation Button
Actual URL: https://www.rubangino.in/EXP-9/index.html#contact
Desired URL: https://www.rubangino.in/EXP-9/index.html#contact
Register Nav Button Working Successfull
Summary -> Test Case 6: Passed

Test Case 7: Testing First Name
Input: Ruban Gino
Summary -> Test Case 7: Passed

Test Case 8: Testing Last Name
Input: Singh
Summary -> Test Case 8: Passed

Test Case 9: Testing User Name
Input: rubangino
Summary -> Test Case 9: Passed

Test Case 10: Testing Email Id
Input: info@rubangino.in
Summary -> Test Case 10: Passed

Test Case 11: Testing Create Password Input
Input: E-Learning@#$System!@#$
Summary -> Test Case 11: Failed

Test Case 12: Testing Re Type Password Input
Input: E-Learning@#$System!@#$
Summary -> Test Case 12: Failed

Test Case 13: Testing Country Input
Input: India
Summary -> Test Case 13: Passed

Test Case 14: Testing Drop Down
Input: Student
Summary -> Test Case 14: Passed
```



```
Test Case 14: Testing Drop Down
Input: Student
Summary -> Test Case 14: Passed

Test Case 15: Testing Phone Number
Input: +919344232829
Summary -> Test Case 15: Passed

Test Case 16: Testing Terms Check Box
Summary -> Test Case 16: Passed

Alert text is: Registration Successful!!

Alert Accepted!!

Web Testing Ended Successfully!
```

RESULT:

A web page was tested in the Chrome & Opera Browsers by creating various testcases

For the project E-Learning System using java and selenium framework.

Ex. No: 10	**ENTITY RELATIONSHIP DIAGRAM**
Date: 11/10/2021	
Video Link:	https://drive.google.com/file/d/1C6-2YY7UM-SnnTytq4lHPcA7P3uPYgYn/view?usp=sharing

OBJECTIVE

The objective of the experiment is to model Entity Relationship diagram for the project E-Learning System.

DESCRIPTION:

An entity relationship diagram (ERD) shows the relationships of entity sets stored in a database. It is usually drawn in a graphical form as boxes (entities) that are connected by lines (relationships) which express the associations and dependencies between entities. An ER diagram illustrates the logical structure of databases. Also, ER diagrams are used to sketch out the design of a database.

Components of ER diagram

Entity: An entity is an object or concept about which you want to store information. An entity set is a collection of similar entities. These entities can have attributes that define its properties.

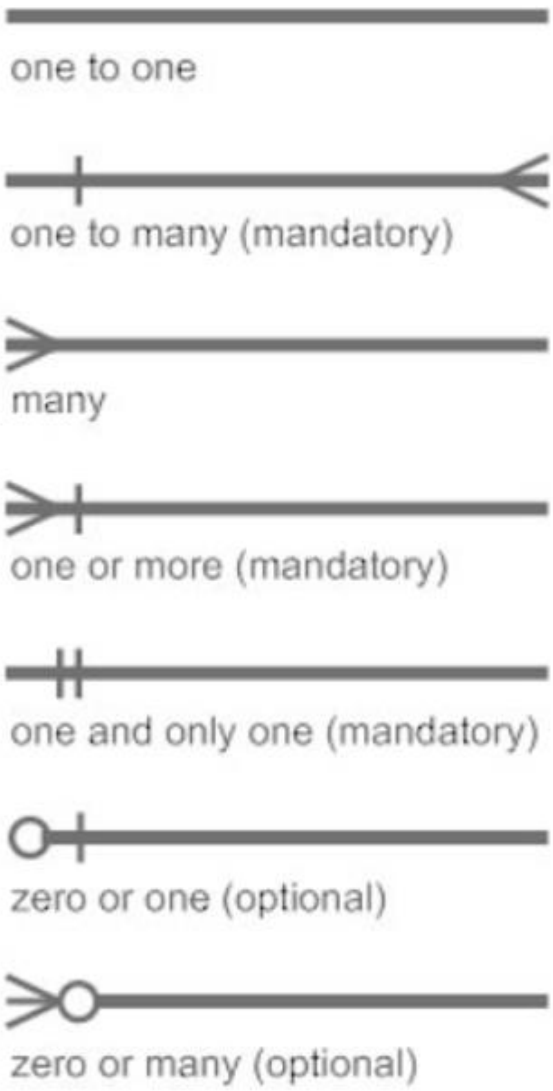
Attribute: Attributes could be properties or traits of an entity A key attribute is the unique, distinguishing characteristic of the entity. An entity’s attribute is made unique or related with another entity by declaring attribute key.

Entity Relationship

A primary key is used to ensure data in the specific column is unique. It is a column cannot have NULL values. It is either an existing table column or a column that is specifically generated by the database according to a defined sequence.

A foreign key is a column or group of columns in a relational database table that provides a link between data in two tables. It is a column (or columns) that references a column (most often the primary key) of another table.

Cardinality: Cardinality specifies how many instances of an entity relate to one instance of another entity.



However, STARUML supports One-to-One Relationship, One-to-Many Relationship or Many-to-Many Relationship.

ALGORITHM

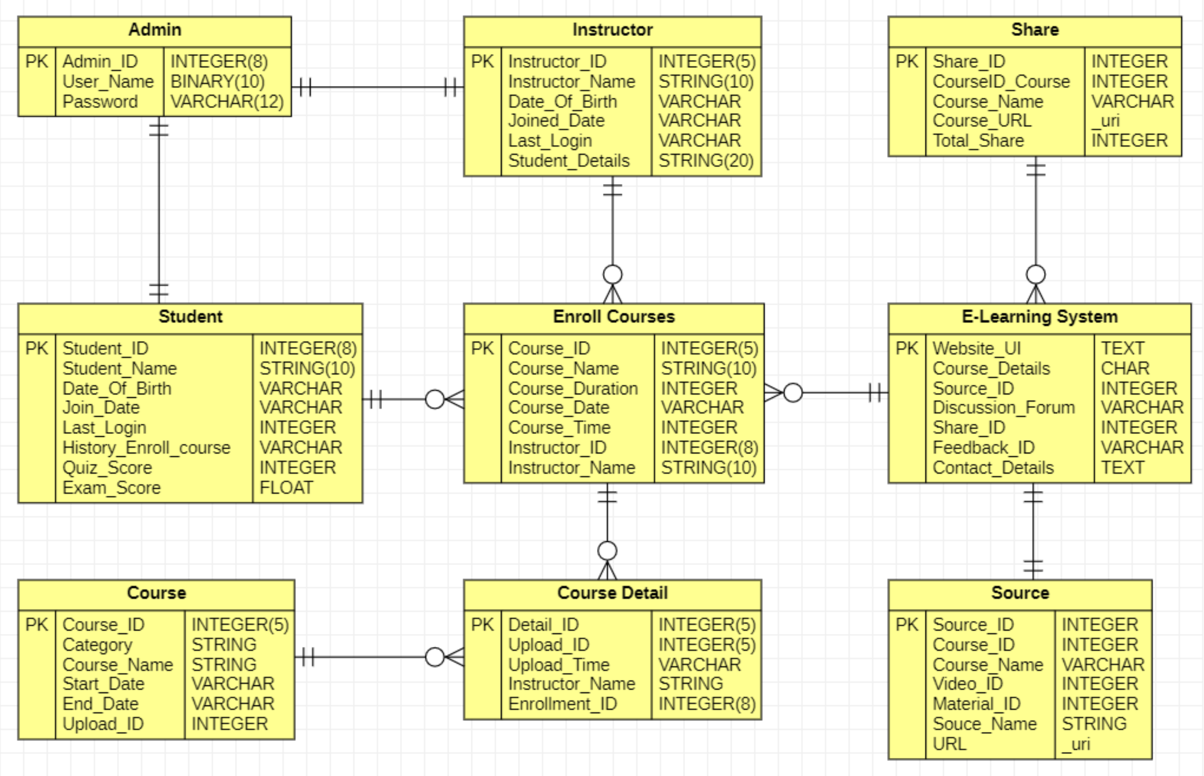
Step 1: Identify the cardinalities and its attributes to the system to be drawn.

Step 2: Identify the primary keys, foreign keys for the attributes.

Step 3: Determine how the cardinalities between the entities are formed

Step 4: Create the ER diagram based on the information from step 1 to 3 with the StarUML provided.

OUTPUT:



	Admin	

PK	Admin_ID
	User_Name
	Password
		INTEGER(8)
		BINARY(10)
		VARCHAR(12)

	Instructor	

PK	Instructor_ID
	Instructor_Name
	Date_Of_Birth
	Joined_Date
	Last_Login
	Student_Details
		INTEGER(5)
		STRING(10)
		VARCHAR
		VARCHAR
		VARCHAR
		STRING(20)

	Share	
PK	Share_ID CourseID_Course Course_Name Course_URL Total_Share	INTEGER INTEGER VARCHAR _uri INTEGER

	Student	
PK	Student_ID Student_Name Date_Of_Birth Join_Date Last_Login History_Enroll_course Quiz_Score Exam_Score	INTEGER(8) STRING(10) VARCHAR VARCHAR INTEGER VARCHAR INTEGER FLOAT

	Enroll Courses	

PK	Course_ID
	Course_Name
	Course_Duration
	Course_Date
	Course_Time
	Instructor_ID
	Instructor_Name
		INTEGER(5)
		STRING(10)
		INTEGER
		VARCHAR
		INTEGER
		INTEGER(8)
		STRING(10)

	E-Learning System	
PK	Website_UI Course_Details Source_ID Discussion_Forum Share_ID Feedback_ID Contact_Details	TEXT CHAR INTEGER VARCHAR INTEGER VARCHAR TEXT

	Course	
PK	Course_ID Category Course_Name Start_Date End_Date Upload_ID	INTEGER(5) STRING STRING VARCHAR VARCHAR INTEGER

	Course Detail	

PK	Detail_ID
	Upload_ID
	Upload_Time
	Instructor_Name
	Enrollment_ID
		INTEGER(5)
		INTEGER(5)
		VARCHAR
		STRING
		INTEGER(8)

	Source	
PK	Source_ID Course_ID Course_Name Video_ID Material_ID Souce_Name URL	INTEGER INTEGER VARCHAR INTEGER INTEGER STRING _uri

RESULT:

The Entity Relationship diagram for the project E-Learning System was modelled successfully.

<table>
<tr><td>Ex No: 11</td><td rowspan="2"><b>SOFTWARE REQUIREMENTS SPECIFICATION</b>
A Prototype model - Develop the prototype SRS of the product</td></tr>
<tr><td>Date: 18/10/2021</td></tr>
<tr><td>Video Link:</td><td>https://drive.google.com/file/d/10Pk6mFPuH_s2eYAE9qwrnyAsv-PkG-r9/view?usp=sharing</td></tr>
</table>

OBJECTIVE

The objective of the experiment is to generate a detailed SRS for the project **AUDITOR MANAGEMENT SYSTEM.**

1. ARRANGEMENT OF CONTENTS

The sequence in which the project report material should be arranged and bound is as follows:

1. Cover Page
2. First Page-same as cover page
3. Table of Contents
4. Chapter 1 – Introduction
5. Chapter 2 – Overall Description
6. Chapter 3 – Specific Requirements
7. Chapter 4 – Technologies used
8. Chapter 5 – Limitation of the project

2. SRS SPECIFICATIONS

- The dimension of the pages should be A4 size.
- Total number of pages for the SRS should be **minimum 40 pages** and should not exceed 50 pages including first page and table of contents page

3. PREPARATION FORMAT

- **Cover Page and First Page-** A specimen copy of the “Cover page” and “Title page” is included at page no.4

- **Table of Contents -**A specimen copy of the "Table of Contents" is included at page no.5 and 6
- **Sections –**The SRS may be broadly divided into 5 sections.

1. Introduction

1.1 Purpose – should mention the objective of the project

1.2 Scope – should describe the scope of the project

1.3 Definitions, Acronyms, and Abbreviations – should provide a very brief explanation of the abbreviations and acronyms used

1.4 References – should specify the references taken (includes book, web references)

1.5 Technologies to be used – should specify the technologies to be used (computer languages, IDEs, or any other specific tool)

1.6 Overview – should describe the sections of the SRS

2. Overall Description

2.1 Product Perspective – should mention the interactions

2.2 Software Interface – should specify the front end and back end of the system

2.3 Hardware Interface – should specify the client and server requirements

2.4 Product Functionalities – should specify the functionalities of the system

2.5 User Characteristics – the characteristics of the user should be mentioned

2.6 Constraints – the constraints of the system should be mentioned

2.7 Architecture Diagram – the interaction between the components/modules should be shown

2.8 Use Case Model Description – use case diagram should be drawn

2.9 Class Diagram – class diagram should be drawn (starUML)

2.9.1 Sequence Diagrams – sequence diagrams should be drawn

2.10 Database Design – ER and Database schema diagrams should be drawn (starUML)

2.10.1 ER Diagram

2.10.2 Schema

2.11 Assumptions and Dependencies – should specify the assumptions made

3. Specific Requirements

3.1 Use Case Reports – should specify the functionalities of the users

3.2 Supplementary Requirements – specify the non-functional requirements

4. Technologies used

4.1 Explanation of computer languages used

4.2 Explanation of IDEs with its features, plug-ins, etc.,

4.3 Explanation of APIs/Frameworks

4.4 Explanation of developer's documentation on how to use the tools

5. Limitation of the project

SRS PREPARATION GUIDELINES

- Tables and figures in a chapter should be placed in the immediate vicinity of the reference where they are cited.
- Figures should have captions and should be numbered. Eg. Fig 5.8 Variation of speed with load (font size-10,title case).
- Figures, Equations, Tables to be continuously numbered within a chapter eg. Fig 2.1, Fig 2.2…../Table 2.1, Table 2.2…..
- For Figures, graphs, charts, flow diagrams etc., captions should be at the bottom.
- For tables, captions should be above the tabular column.
- Tables and figures should confirm to the margin specifications. Large size figures should be reduced to the appropriate size before insertion.
- MS Word Times New Roman font, size 12 for body of text; size 13 for sub section titles and bold & CAPS, for sub sections of sub sections, font size is 13, Title Case; size 10 for table heading and figures and captions (title case).
- For chapter title, font size as 14 and bold & CAPS and font Times New Roman.
- Enough matter (text, figures etc.) should be there in a page; avoid white spaces.
- All pages should be numbered continuously; printed at bottom and centre of each page starting from chapter 1
- Exactly 1.5 line spacing.
- Margin: Left=1¼" (1.25), Right= ¾" (0.75), top=1", bottom=1".
- Subscripts and superscripts should be properly printed.

- Reference in body to be given with square bracket. **E.g.:** Visual cryptography is a new technique which provides information security which uses simple algorithm unlike the complex, computationally intensive algorithms. [25]

RESULT:

A software requirement specification for modelling a prototype of **AUDITOR MANAGEMENT SYSTEM** was carried out successfully.

AUDITOR MANAGEMENT SYSTEM

SOFTWARE REQUIREMENT SPECIFICATION

[BATCH NO: 03]

Submitted by:

RUBAN GINO SINGH.A - Reg. No. UR20CS2001

In partial fulfillment of the requirements for the

Software Engineering

Under the guidance of

Mr. TITUS

KARUNYA INSTITUTE OF TECHNOLOGY AND SCIENCES

Declared as Deemed to be University under sec.3 of the UGC Act, 1956

Karunya Nagar, Coimbatore -641 114

OCTOBER 2021

TABLE OF CONTENTS

Page

No

1.Introduction:

Auditor Management System - provides a user-friendly way of managing Auditors in a company. It allows tracking the reports, predicts risks, provides opinions, risk management, and gets the history of the projects from the central repository.

1.1 Purpose:

An auditor management system is to manage the auditors in a particular company. It helps companies streamline their audit processes and comply with regulations or internal policies. Financial records of a company have perfectly verified after every transaction, so to attain this, every company needs auditors to make the monthly records and want to clear tax norms. Audit management helps simplify and well-organize the workflow and collaboration process of compiling audits. Most audit teams heavily rely on email and shared drive for sharing information. This Auditor management system will help the Auditors to fulfil their daily audit needs. Typical tasks such as submitting client requests, sending reminders, and following up on findings had done utilizing broad tools. Investing in the right software could help save time, reduce errors and save on resources. The purpose of a management audit is not to appraise individual executive performance but to evaluate the management team in its effectiveness to work in the interests of shareholders, maintain good relations with employees, and uphold reputational standards. It is important to stress that the management audit assesses the overall management of the company, not the performance of individual managers. This system will assist the auditors and the companies to finish and turned the work at the correct time.

1.2 Scope:

- Create different users like Auditors, managers, and the Board of Directors to assign and receive the privileges and duties.
- All the users need to authenticate to avail of the services from this management system.
- Details of auditors can view easily using the unique id.
- Allows Administrator to add, view, edit and delete auditors records efficiently. And also, to generate reports.
- Allows tracking of audit reports can generated in pdf or other usable

- Maintain an FAQ section as guidance for all the users.
- Secure Socket Layer (SSL) connectivity to ensure security for the Auditor management system as a website
- Usage of jQuery for web JS
- Usage of SOA for more effortless connectivity and enhancement

1.3 DEFINITIONS, ACRONYMS, AND ABBREVIATIONS

- **HTML:** Hypertext Markup Language is used to create static websites
- **CSS:** Cascading Style Sheet used to style the static webpage
- **JS:** JavaScript is a scripting language used for client-side web development
- **Node JS:** A JavaScript library used for backend web development
- **JQUERY:** jQuery is a JavaScript library that emphasizes interaction between JavaScript and HTML.
- **HTTP:** Hypertext Transfer Protocol is a transaction-oriented client/server protocol between the web browser and a web server
- **HTTPS:** Secure Hypertext Transfer Protocol is HTTP over SSL (Secure Socket Layer)
- **SOA:** Service-Oriented Architecture
- **EJB:** Enterprise Java Beans is an architecture for the development and deployment of distributed object application-based server-side software components
- **DB2:** IBM Database 2 is a database management system that provides a flexible and efficient database platform to raise strong "on demand" business applications
- **AJAX:** Asynchronous JavaScript and XML is a technique used in JavaScript to create dynamic web pages
- **WASCE:** Web Sphere Application Server Community Edition is an application server that runs and supports the web applications

1.4 REFERENCES

- IBM Database management: https://www.ibm.com/in-en/products/db2-database

- Wikipedia: www.wikipedia.com

1.5 TOOLS AND TECHNOLOGIES

- **HTML:** Hypertext Markup Language
- **CSS:** Cascading Style Sheets
- **JS:** JavaScript
- **JQ:** jQuery
- **XML:** Extensible Markup Language
- **UML:** Unified Modelling Language
- **EJS:** Express JavaScript
- **SOA:** Service-Oriented Architecture
- **WASCE:** (Web Sphere Application Server Community Edition) Web server.

1.6 OVERVIEW

The SRS includes two sections,

Overall Description: This section will describe significant components of the system, interconnections, and external interfaces.

Specific Requirements: This section will describe the functions of actors, their roles in the system, and the constraints faced by the system.

2. OVERALL DESCRIPTION

Overall description of auditor management system describes the major components of the system, inter-connections, and external interfaces.

2.1 PRODUCT PERSPECTIVE

- The auditor management system follows a service-based architecture for easy pluggability and interaction with other services.
- The front-end tier has a web application, which had performed with JavaScript.
- The business components then interact with the IBM DB2 database. Pure XML will utilize as it offers a hierarchical storage mechanism that is easier to use.

2.2 SOFTWARE INTERFACE

- **Front End Client:** Web browser, Operating System (any operating system applicable)
- **Web Server:** cPanel or WASCE, Operating System (any operating system applicable)
- **Database Server:** DB2, Operating System (any operating system is applicable)
- **Back End:** DB2
- **Development Tool:** Visual Studio Code, Windows, Web Sphere server

2.3 HARDWARE INTERFACE

- **Client-side:** Internet Explorer 7.0 and above, Google Chrome, Mozilla Firefox, and Opera
- **Processor:** i3 core and above
- **RAM:** 1GB or above for better experience
- **Disk Space:** 2GB

- **Server Side:** Web server application
- **Processor:** i3 or above
- **RAM:** 2GB or above for better experience
- **Disk Space:** 2GB

2.4 PRODUCT FUNCTIONALITIES

- **Administrator Facilities:** The administrator facilities include adding new users, editing user-profiles, and generating reports
- **Auditor facilities:** Auditor’s facilities include Risk management, providing opinions, and giving finished reports.
- **Reports:** The Auditor and management reports can generate every week. Also, automatic mail client works on the release to send directly to the Board of Directors.
- **User (Auditor Facilities):** The user can view or edit their profiles, change password, log in and log out.

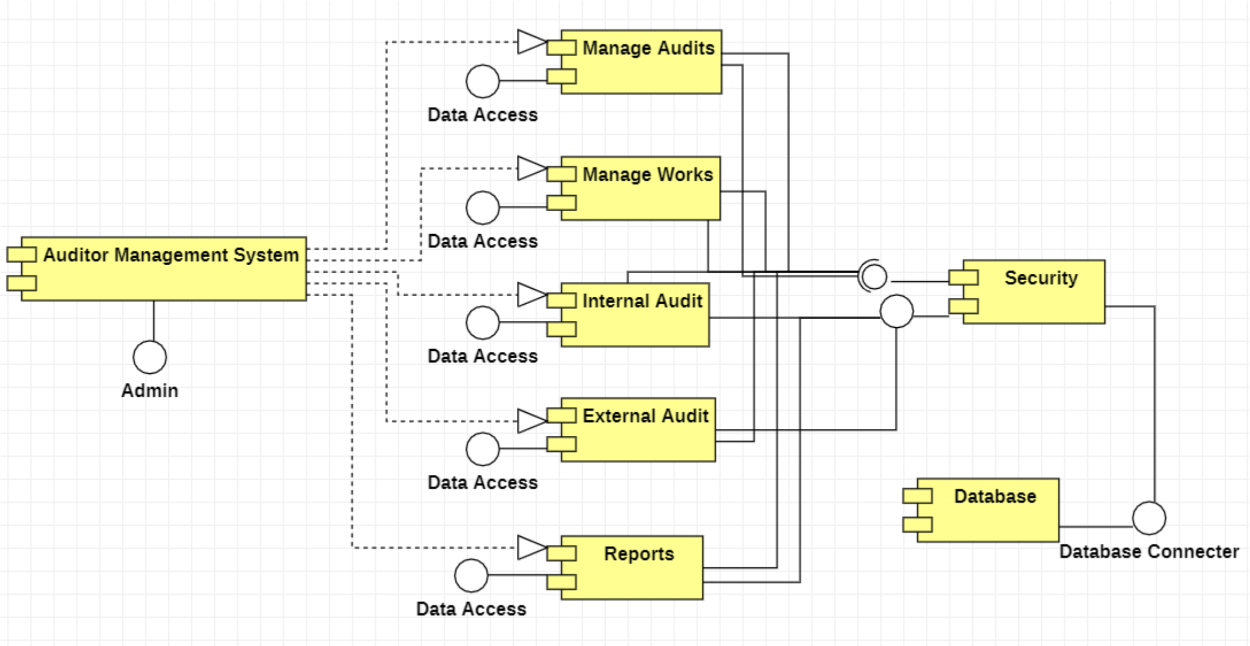
2.5 USER CHARACTERISTICS:

The user must know about operating a computer, browsing the internet and English to login and access the facilities assigned.

2.6 CONSTRAINTS

- GUI is only in English.
- Username and password only used for only the web user authentication
- Only Registered Auditors, managers, company's board of directors are allowed to use the services

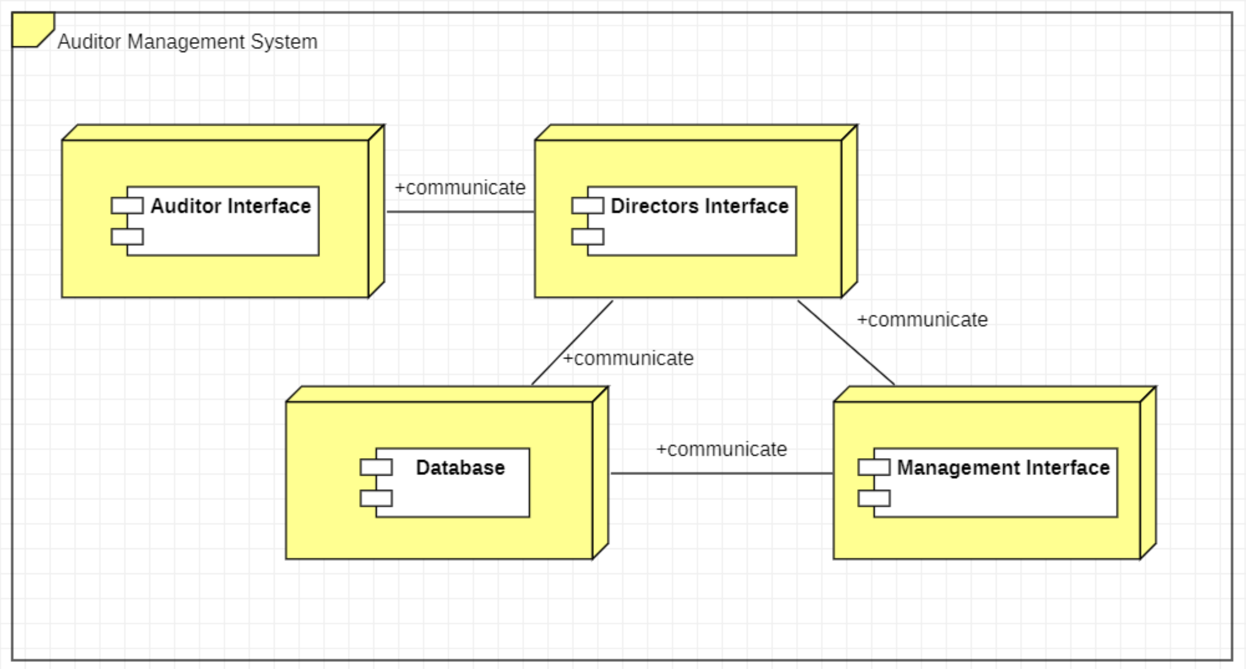
2.7 ARCHITECTURE DIAGRAM



Architecture Diagram is having it's another name of Component diagram which states that the Components which are available in a particular system. Here, in this Auditor management system the component diagram states that the Auditor management system which has connected to the Interface called Admin. And the system has seven components are Manage Audits, Manage Works, Internal Audits, External Audits, Reports, Security, and the Database.

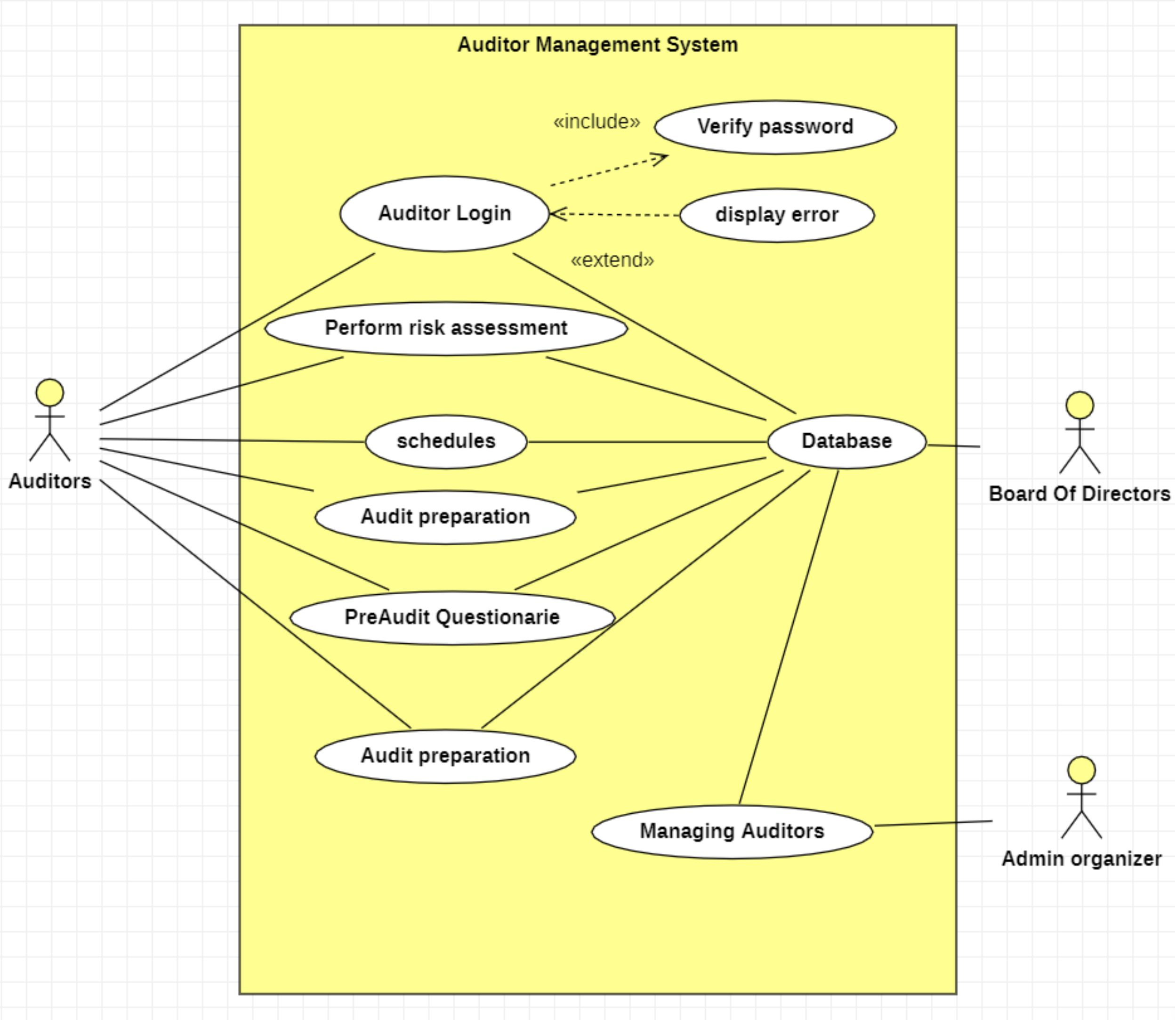
There are two nodes under every component which are available in the main Interface. The main system is connected to the first node of each, and every component and the secondary node is connected to the second node which is the Data Access, and the same nodes are connected to the security of the system with two interfaces. And the last database is connected by using database connecter.

2.7.1 DEPLOYMENT DIAGRAM



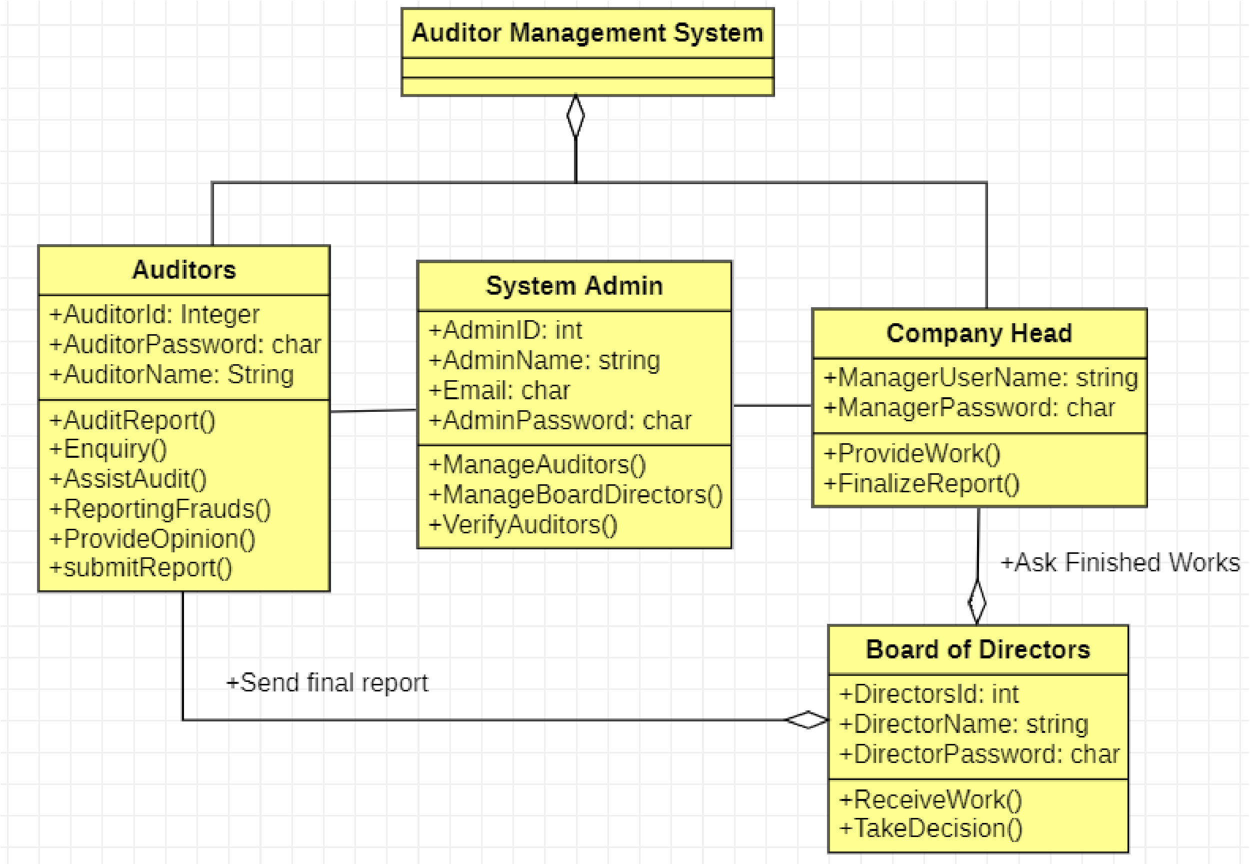
Deployment diagram states that the Interface deployment in the auditor management system. There are four main interfaces in the auditor management system namely, Auditor Interface, Directors Interface, Database, and Management Interface. These Interface are connected with a communication path where the four interfaces of this management system will communicate to the main UI.

2.8 USE CASE MODEL DESCRIPTION



Use case diagram tells us the Main categorial uses of this Auditor Management System. It has three actors where the three actors of Auditors, Board of Directors, and Admin Organizer. Here, Auditors having the use case of Auditor Login, perform risk management, schedules, audit preparation, preaudit questionnaire and audit. All these operations are connected to the Database where the board of directors can easily get the sufficient data's from the database. The Admin panel organizer can easily access these statements at right time to generate the stable Audit reports.

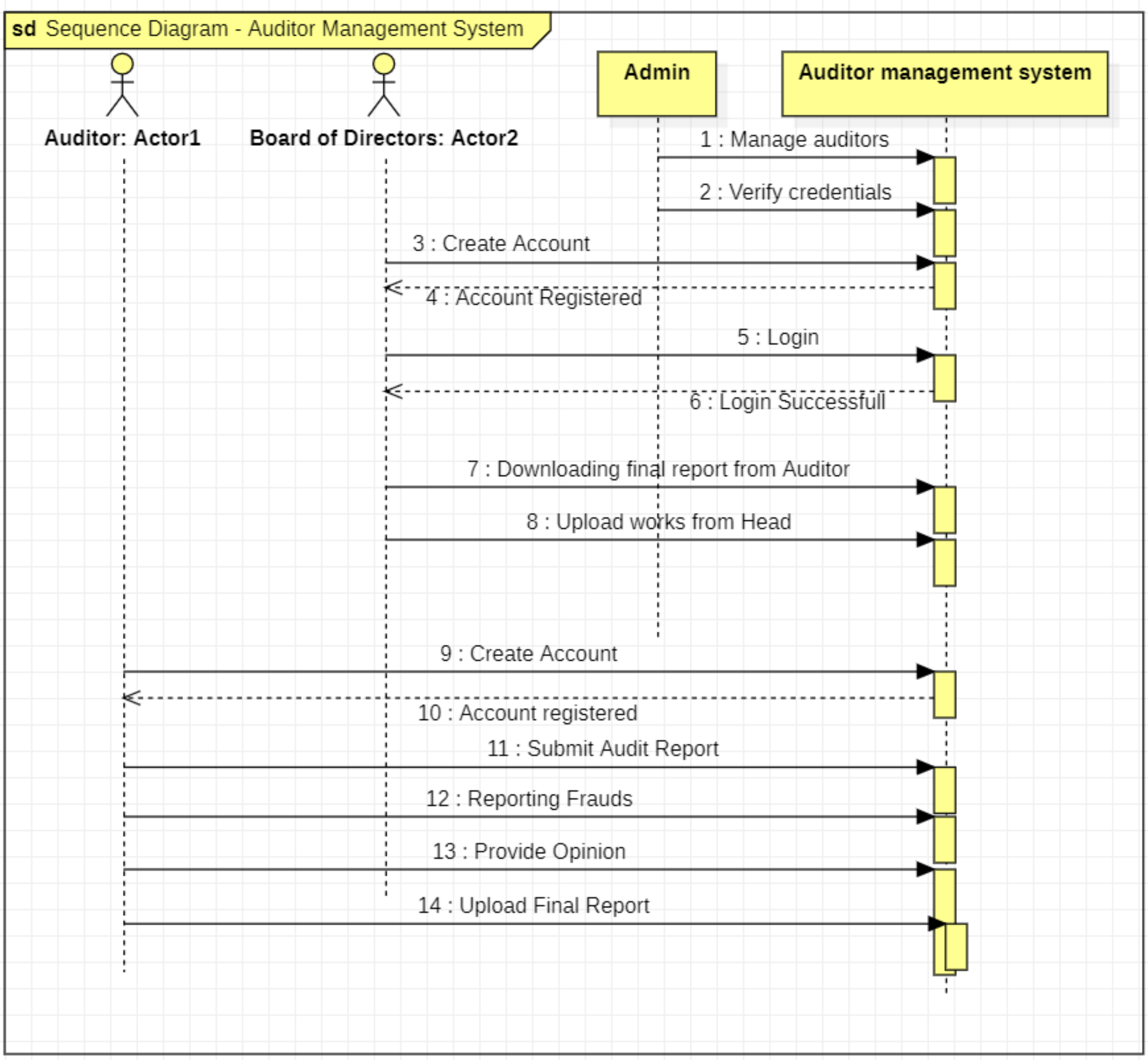
2.9 CLASS DIAGRAM



Class diagrams separate the single system to different classes of Auditors, System Admin, company head and the Board of directors from the auditor management system. Here the auditors have the operations of preparing the Audit report, enquiry, assist audit, reporting frauds, provide opinion, and submit the final report.

System admin Organized the auditors and managing the board of director accounts and verifying the auditors. Here all the classes are connected using aggregation.

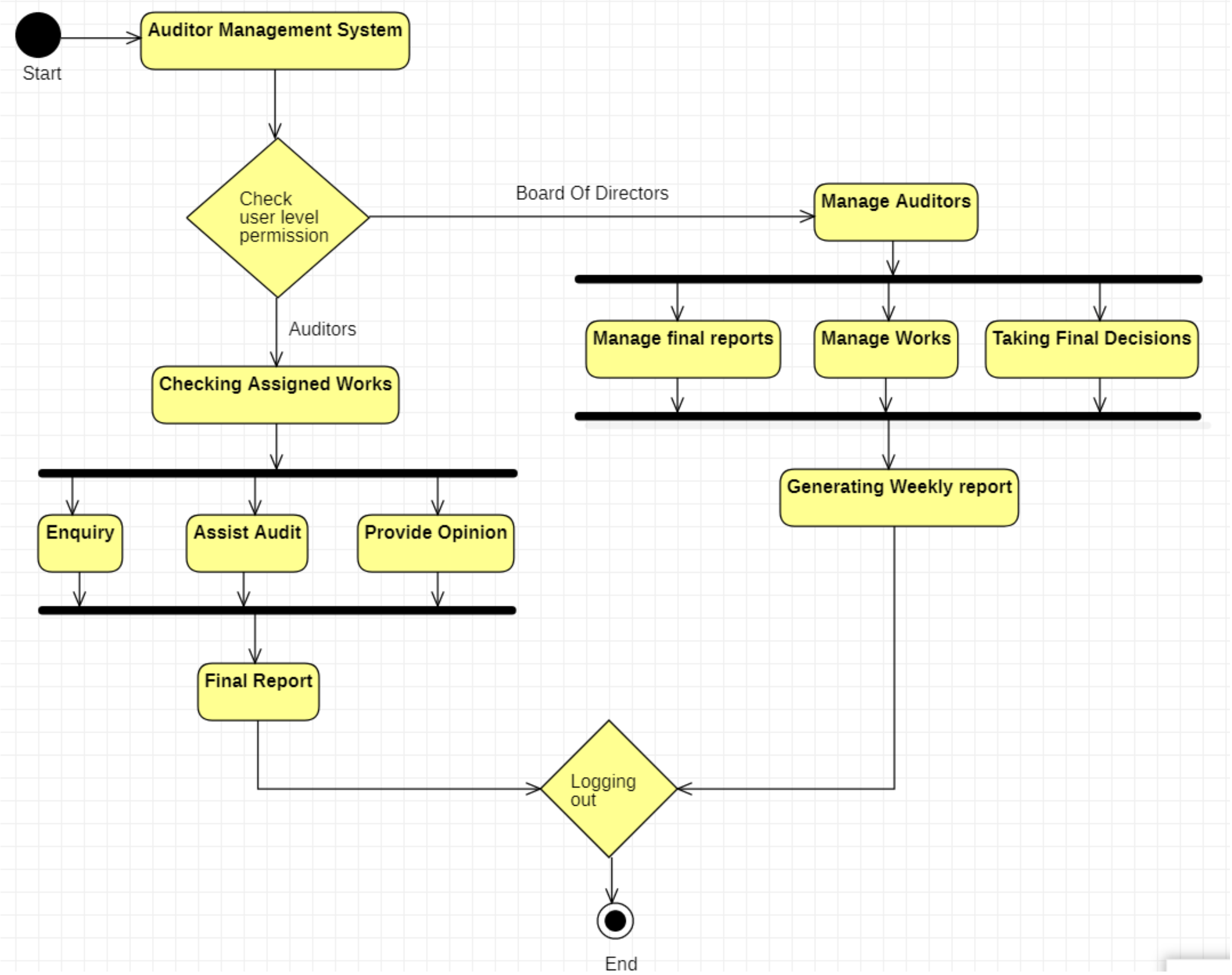
2.9.1 SEQUENCE DIAGRAMS



Sequence Diagram separates the auditor management system into four number of sequences. Here, it has two actors, where first actor is Auditors, and the second actor is board of directors. These two actors are connected into the lifeline of Admin and auditor management system.

First the admin will be verifying the Auditors and the board of directors' accounts. Then the board of directors will take an initiative to create an account and the system will retrieves the message as the account created successful. After the board of directors will download the reports from the auditors. Same things can be done by the auditors. Where they are just upload the final report.

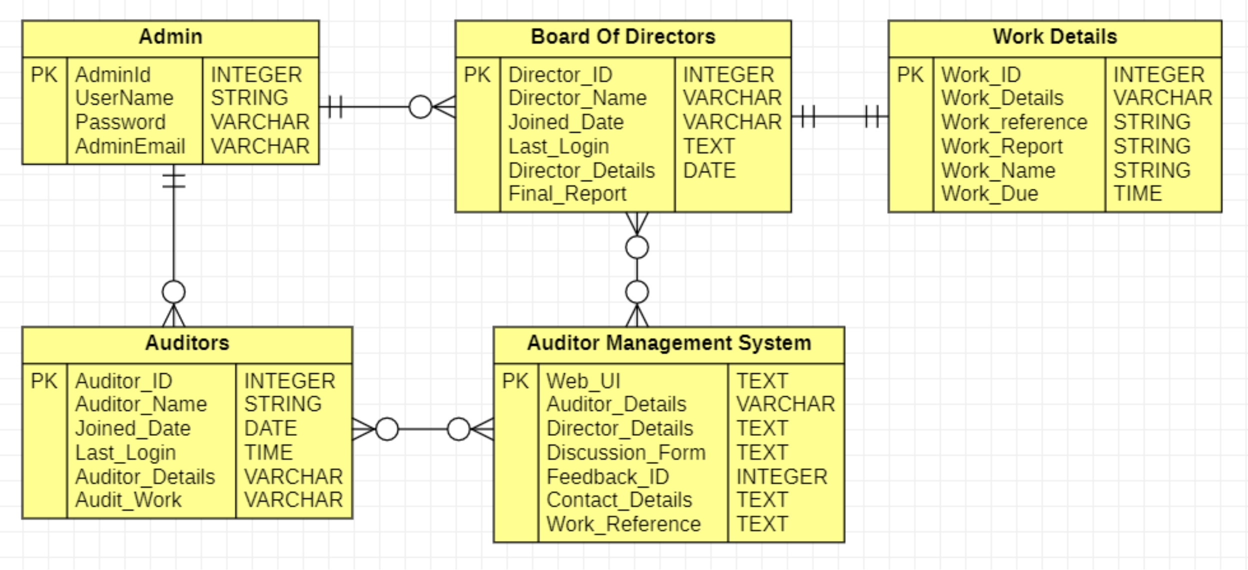
2.10 ACTIVITY DIAGRAM



Activity diagram separates the activities of the system into parts it has start and it check the user level permission. According to the user level permission it will get into the interfaces. In this auditor management system first, it will check the user level permission, and if the user is auditor then it will redirect the user to the auditor interface else it will get the user into the board of director interface. And after both the interface or the works finished in the interface it will proceed to the log out section to end the system.

2.11 DATABASE DESIGN

2.11.1 ER DIAGRAM



ER Diagram is nothing but the Entity Relationship Diagram. Where the Entities are the databases which are in the Auditor Management System, here this system has five databases where the Admin, Auditors, Board of Directors, the Work details are occupies in the Single individual databases.

Admin has the entities of admin id, username, password, and admin email. The Board of directors has the entities of director id, director name, joined date, last login and the director details and the final report. The work details have an entities of work id, work details, work reference, work report, work name, and the work due. The auditors have the entitles of auditor id, auditor name, joined date, last login, auditor details and the audit work.

2.12 ASSUMPTIONS AND DEPENDENCIES

- The controller is created in the system already to add, edit, delete and view the details of the users.
- The users and their roles are predefined.
- The auditors or users of the system should have basic knowledge of English and how to operate a computer and a web browser.

3. SPECIFIC REQUIREMENTS

This auditor management system does not have any specific requirements since the data from the system is fully handled by the cloud

3.1 USE CASE REPORTS

- **Login:** The Auditors, Board of directors and managers can login with username and password.
- **View/Edit profile:** Every individual can be able to edit his/her profile.
- **Assign work:**

3.2 SUPPLEMENTARY REQUIREMENTS

- Secure access of confidential data
- 24/7 availability - The system can be automated so that it works around the clock and during power failures
- Better component design to get better performance at peak time.
- Flexible service-based architecture for future extension

4. TECHNOLOGIES USED

- **HTML:** Hypertext Markup Language
- **CSS:** Cascading Style Sheets
- **JS:** JavaScript
- **JQ:** jQuery
- **XML:** Extensible Markup Language
- **UML:** Unified Modelling Language
- **EJS:** Express JavaScript
- **SOA:** Service-Oriented Architecture

- **WASCE:** (Web Sphere Application Server Community Edition) Web server.

4.1 LANGUAGES USED

HTML: Hypertext Markup Language

CSS: Cascading Style Sheets

JavaScript: Scripting Language

4.2 IDE/TOOLS USED

VS Code: Visual Studio Code

4.3 API/FRAMEWORK DOCUMENTATION

Bootstrap: https://getbootstrap.com/

IBM DB2: https://cloud.ibm.com/apidocs/cloudant

Apache: http://httpd.apache.org/docs/

4.4 DEVELOPER’S DOCUMENTATION

HTML: https://developer.mozilla.org/en-US/docs/Web/HTML

CSS: https://developer.mozilla.org/en-US/docs/Web/CSS

JS: https://developer.mozilla.org/en-US/docs/Web/JavaScript

REST API: https://developers.google.com/docs/api/reference/rest

5. LIMITATION OF THE PROJECT

An auditor management system is a heavily used project for big companies so, it wants to be stable and efficient in calculating the risks.

The main limitation would be the time needed to develop this website is around five months. since we have to implement such highly addressable algorithms in Artificial intelligence and some algorithms based on Blockchain technologies. Therefore, the time taken is the only limitation for this project.

RESULT:

A software requirement specification for modelling a prototype of **AUDITOR MANAGEMENT SYSTEM** was carried out successfully.